

ALGORITHMS FOR IMAGE CLASSIFICATION USING WAVELETS AND FUSION METHODS

by

Yuki C. Yunes Saito

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2005

Approved by:

Domingo Rodríguez Rodríguez, PhD
Member, Graduate Committee

Date

Hamed Parsiani, PhD
Member, Graduate Committee

Date

Ramón Vásquez Espinosa, PhD
President, Graduate Committee

Date

Edgar Acuña, PhD
Representative of Graduate Studies

Date

Isidoro Couvertier, PhD
Chairperson of the Department

Date

ABSTRACT

This work presents a set of algorithms for feature extraction and image classification using wavelet transform and fusion methods. First we use wavelet transform as a feature extraction method. We present some basic definitions and theoretical background of the problem and proposed methodologies for solving it.

We decompose the image to a second level decomposition giving us the coefficients of approximation, diagonal, horizontal and vertical. We used nine different wavelet filters in order to test how the classification performed depending on the filters, images and coefficients used. The algorithms used for image classification are Euclidean Distance, Maximum Likelihood and K-Nearest Neighbor. Then all of the results from the algorithms individually go to a fusion center where we use Majority Voting.

Images from the Brodatz album and an image of Mayagüez Bay are used. Different experiments were done trying to aim at the best results possible. We got really interesting results for some of the textures. Some of them gave us really good results for the classifiers and filters, but the others did not. We normalized all of the samples in order to achieve a better discrimination.

RESUMEN

Este trabajo presenta un conjunto de algoritmos para “feature extraction” y “clasificación de imágenes” utilizando “transformadas de wavelet” y “métodos de fusión”. Primeramente utilizamos las transformadas de wavelet como método de “feature extraction”. Presentamos algunas definiciones básicas y un marco teórico del problema y los métodos propuestos para la solución del mismo.

Descomponemos la imagen hasta el segundo nivel de descomposición, obteniendo los coeficientes de aproximación, diagonales, verticales y horizontales. Utilizamos nueve filtros diferentes para probar el desempeño del proceso de clasificación, dependiendo de los filtros, imágenes y coeficientes utilizados. Los algoritmos utilizados para la clasificación de las imágenes son el “Euclidean Distance”, “Maximun Likelihood” y “K-Nearest Neighbor”. Luego los resultados de estos algoritmos son llevados a un centro de fusión, donde utilizamos “Majority Voting”.

Utilizamos imágenes del “Brodatz album” y una imagen de la bahía de Mayagüez. Realizamos varios experimentos tratando de alcanzar los mejores resultados posibles. Obtuvimos resultados realmente interesantes para algunas texturas. Algunos de ellos nos dieron resultados muy buenos para los clasificadores y los filtros, aunque con otros no los obtuvimos. Normalizamos todas las muestras con el fin de obtener una mejor discriminación.

ACKNOWLEDGEMENTS

First I would like to thank Dr. Ramón Vázquez Espinosa for being my chairman and an excellent advisor in my graduate studies. Thanks for your unconditional support in the development of this thesis.

I want to thank in a very special way Dr. Domingo Rodríguez Rodríguez for lending me his valuable time and resources whenever I needed them. I also like to thank Dr. Hamed Parsiani for his good advice and experimental ideas. To Dr. Edgar Acuña, thank you for guiding me through the statistics areas needed for the research.

Thanks to NASA, NOAA and UPRM for granting me the opportunity and support to obtain my master's degree.

In a unique way, I would like to thank my family and friends for being a source of motivation, inspiration and support.

Finally, thank you to God for the gift of life and the strength to complete this challenge.

Table of Contents

ABSTRACT	II
RESUMEN	III
ACKNOWLEDGEMENTS	IV
TABLE OF CONTENTS.....	V
TABLE LIST	VIII
FIGURE LIST	X
1 INTRODUCTION	1
1.1	MOTIVATION
1.2	PROBLEM FORMULATION AND PROPOSED SOLUTION
1.3	CONTRIBUTION
1.4	SUMMARY OF FOLLOWING CHAPTERS
2 THEORETICAL BACKGROUND.....	6
2.1	PIXEL DEFINITION
2.2	TEXTURE DEFINITION
2.3	TEXTON DEFINITION
2.4	CLASS DEFINITION
2.5	CLASSIFICATION
2.6	DISCRIMINANT FUNCTIONS
2.7	SUPERVISED CLASSIFICATION
2.8	CLASSIFIER LAWS
2.8.1	<i>Euclidean Distance (ED)</i>
2.8.2	<i>Fisher Linear Discriminant (FLD)</i>
2.8.3	<i>Minimum Mahalanobis Distance Classifier (MD)</i>
2.8.4	<i>Maximum Likelihood Classifier (ML)</i>
2.8.5	<i>Maximum A Posteriori (MP)</i>
2.8.6	<i>K-Nearest Neighbor (KNN)</i>
2.8.7	<i>Neural Networks (NN)</i>
2.9	MAX, MIN, AVERAGE AND MAJORITY VOTING FUSION RULES

2.10.....	WAVELET ANALYSIS	19
2.10.1 <i>Wavelet Definition</i>		19
2.10.2 <i>Wavelet Analysis</i>		20
2.10.3 <i>Multiple-Level Decomposition</i>		22
2.11.....	PREVIOUS WORK	24
3	METHODOLOGY	29
3.1.....	SAMPLE SELECTION	31
3.1.1 <i>Images used</i>		31
3.1.2 <i>Image Sampling</i>		32
3.1.3 <i>Data arrange</i>		34
3.1.4 <i>Training and Testing Samples Selection and Data Filtering</i>		35
3.2.....	ALGORITHMS	37
4	EXPERIMENTS AND RESULTS	40
4.1.....	RESULTS FROM THE CLASSIFIERS	40
4.1.1 <i>Results using samples size of 32 by 32</i>		41
4.2.....	RESULTS FROM THE FUSION METHODS	59
4.2.1 <i>Results using Fusion Method for samples size of 32 by 32</i>		59
4.3.....	OTHER EXPERIMENTS MADE	64
4.3.1 <i>Variance as a measure of activity</i>		64
4.3.2 <i>Classification of samples taken from Mayagüez Bay</i>		66
4.3.3 <i>Results of Fusion Methods using samples size of 32 by 32 with approximation coefficients</i>		68
4.3.4 <i>Results using samples size of 64 by 64 with approximation coefficients</i>		73
4.3.5 <i>Results using Fusion Method for samples size of 64 by 64</i>		80
4.3.6 <i>Histograms of samples from the classes used</i>		82
5	CONCLUSIONS AND FUTURE WORK	88
6	USER GUIDE FOR THE ALGORITHMS	91
	REFERENCES	96
	APPENDIX A. SAMPLES SELECTION	107
	APPENDIX A1 CODES FOR 32 BY 32 SAMPLE SIZE WITHOUT OVERLAPPING.....	107
	APPENDIX A2 CODES FOR 64 BY 64 SAMPLE SIZE WITHOUT OVERLAPPING.....	107
	APPENDIX A3 CODES FOR 64 BY 64 SAMPLE SIZE WITH OVERLAPPING.....	108
	APPENDIX B PROGRAMS FOR FILTERING, TRAINING AND TESTING SAMPLES SELECTION	109
	APPENDIX B1 PROGRAM FOR TRAINING AND TESTING SAMPLES SELECTION.....	109
	APPENDIX B2 WAVELET FUNCTION.....	115

APPENDIX B3 WAVELET FILTERS.....	118
APPENDIX C CLASSIFIERS ALGORITHMS AND FUSION METHODS	130
APPENDIX C1 EUCLIDEAN DISTANCE CLASSIFIER.....	130
APPENDIX C2 EUCLIDEAN DISTANCE FUNCTION.....	197
APPENDIX C2 K-NEAREST NEIGHBOR K=3 CLASSIFIER.....	198
APPENDIX C3 K-NEAREST NEIGHBOR K=5 CLASSIFIER.....	410
APPENDIX C4 MAXIMUM LIKELIHOOD CLASSIFIER	549
APPENDIX C5 MAXIMUM LIKELIHOOD FUNCTION.....	594
APPENDIX C6 MAJORITY VOTING FUSION RULE.....	595

Table List

Tables	Page
TABLE 4.1 Euclidean Distance results for ‘db1’ using samples of 32 by 32	41
TABLE 4.2 Maximum Likelihood results for ‘db1’ using samples of 32 by 32.....	41
TABLE 4.3 K-Nearest Neighbor K=3 results for ‘db1’ using samples of 32 by 32	42
TABLE 4.4 K-Nearest Neighbor K=5 results for ‘db1’ using samples of 32 by 32	42
TABLE 4.5 Euclidean Distance results for ‘haar’ using samples of 32 by 32	43
TABLE 4.6 Maximum Likelihood results for ‘haar’ using samples of 32 by 32.....	43
TABLE 4.7 K-Nearest Neighbor K= 3 results for ‘haar’ using samples of 32 by 32	44
TABLE 4.8 K-Nearest Neighbor K=5 results for ‘haar’ using samples of 32 by 32	44
TABLE 4.9 Euclidean Distance results for ‘coif1’ using samples of 32 by 32.....	45
TABLE 4.10 Maximum Likelihood results for ‘coif1’ using samples of 32 by 32.....	45
TABLE 4.11 K-Nearest Neighbor K=3 results for ‘coif1’ using samples of 32 by 32	46
TABLE 4.12 K-Nearest Neighbor K=5 results for ‘coif1’ using samples of 32 by 32	46
TABLE 4.13 Euclidean Distance results for ‘sym2’ using samples of 32 by 32	47
TABLE 4.14 Maximum Likelihood results for ‘sym2’ using samples of 32 by 32	47
TABLE 4.15 K-Nearest Neighbor K=3 results for ‘sym2’ using samples of 32 by 32....	48
TABLE 4.16 K-Nearest Neighbor K=5 results for ‘sym2’ using samples of 32 by 32....	48
TABLE 4.17 Euclidean Distance results for ‘bior1.1’ using samples of 32 by 32.....	49
TABLE 4.18 Maximum Likelihood results for ‘bior1.1’ using samples of 32 by 32	49
TABLE 4.19 K-Nearest Neighbor K=3 results for ‘bior1.1’ using samples of 32 by 32.	50
TABLE 4.20 K-Nearest Neighbor K=5 results for ‘bior1.1’ using samples of 32 by 32.	50
TABLE 4.21 Euclidean Distance results for ‘rbio1.1’ using samples of 32 by 32.....	51
TABLE 4.22 Maximum Likelihood results for ‘rbio1.1’ using samples of 32 by 32	51
TABLE 4.23 K-Nearest Neighbor K=3 results for ‘rbio1.1’ using samples of 32 by 32.	52
TABLE 4.24 K-Nearest Neighbor K=5 results for ‘rbio1.1’ using samples of 32 by 32.	52
TABLE 4.25 Euclidean Distance results for ‘db2’ using samples of 32 by 32.....	53
TABLE 4.26 Maximum Likelihood results for ‘db2’ using samples of 32 by 32.....	53
TABLE 4.27 K-Nearest Neighbor K=3 results for ‘db2’ using samples of 32 by 32	54
TABLE 4.28 K-Nearest Neighbor K=5 results for ‘db2’ using samples of 32 by 32	54
TABLE 4.29 Euclidean Distance results for ‘bior3.1’ using samples of 32 by 32.....	55
TABLE 4.30 Maximum Likelihood results for ‘bior3.1’ using samples of 32 by 32	55
TABLE 4.31 K-Nearest Neighbor K=3 results for ‘bior3.1’ using samples of 32 by 32.	56
TABLE 4.32 K-Nearest Neighbor K=5 results for ‘bior3.1’ using samples of 32 by 32.	56
TABLE 4.33 Euclidean Distance results for ‘rbio3.1’ using samples of 32 by 32.....	57
TABLE 4.34 Maximum Likelihood results for ‘rbio3.1’ using samples of 32 by 32	57
TABLE 4.35 K-Nearest Neighbor K=3 results for ‘rbio3.1’ using samples of 32 by 32.	58
TABLE 4.36 K-Nearest Neighbor K=5 results for ‘rbio3.1’ using samples of 32 by 32.	58
TABLE 4.37 Majority Voting results for ‘db1’ using samples of 32 by 32.....	59

TABLE 4.38 Majority Voting results for ‘haar’ using samples of 32 by 32	60
TABLE 4.39 Majority Voting results for ‘coif1’ using samples of 32 by 32.....	60
TABLE 4.40 Majority Voting results for ‘sym2’ using samples of 32 by 32	61
TABLE 4.41 Majority Voting results for ‘bior1.1’ using samples of 32 by 32	61
TABLE 4.42 Majority Voting results for ‘rbio1.1’ using samples of 32 by 32	62
TABLE 4.43 Majority Voting results for ‘db2’ using samples of 32 by 32.....	62
TABLE 4.44 Majority Voting results for ‘bior3.1’ using samples of 32 by 32	63
TABLE 4.45 Majority Voting results for ‘rbio3.1’ using samples of 32 by 32	63
TABLE 4.46 Variance as a measure of activity results for ‘db1’	64
TABLE 4.47 Majority Variance as a measure of activity results for ‘sym2’	65
TABLE 4.48 Majority Variance as a measure of activity results for ‘rbio31’	65
TABLE 4.49 Euclidean Distance results for ‘db1’ samples from Mayagüez Bay	67
TABLE 4.50 Maximum Likelihood results for ‘db1’ samples from Mayagüez Bay.....	67
TABLE 4.51 K-Nearest Neighbor K=3 results for ‘db1’ samples from Mayagüez Bay .	67
TABLE 4.52 K-Nearest Neighbor K=5 results for ‘db1’ samples from Mayagüez Bay .	67
TABLE 4.53 Majority Voting results for ‘db1’ samples from Mayagüez Bay	68
TABLE 4.54 Euclidean Distance results for ‘db1’ using samples of 64 by 64.....	74
TABLE 4.55 Maximum Likelihood results for ‘db1’ using samples of 64 by 64.....	74
TABLE 4.56 K-Nearest Neighbor K=3 results for ‘db1’ using samples of 64 by 64	75
TABLE 4.57 K-Nearest Neighbor K=5 results for ‘db1’ using samples of 64 by 64	75
TABLE 4.58 Euclidean Distance results for ‘coif1’ using samples of 64 by 64.....	76
TABLE 4.59 Maximum Likelihood results for ‘coif1’ using samples of 64 by 64.....	76
TABLE 4.60 K-Nearest Neighbor K=3 results for ‘coif1’ using samples of 64 by 64	77
TABLE 4.61 K-Nearest Neighbor K=5 results for ‘coif1’ using samples of 64 by 64	77
TABLE 4.62 Euclidean Distance results for ‘sym2’ using samples of 64 by 64	78
TABLE 4.63 Maximum Likelihood results for ‘sym2’ using samples of 64 by 64	78
TABLE 4.64 K-Nearest Neighbor K=3 results for ‘sym2’ using samples of 64 by 64....	79
TABLE 4.65 K-Nearest Neighbor K=5 results for ‘sym2’ using samples of 64 by 64....	79
TABLE 4.66 Majority Voting results for ‘db1’ using samples of 64 by 64.....	80
TABLE 4.67 Majority Voting results for ‘coif1’ using samples of 64 by 64.....	81
TABLE 4.68 Majority Voting results for ‘sym2’ using samples of 64 by 64	81

Figure List

Figures	Page
Figure 2.1 Simple example classifier schemes, from reference [50].	8
Figure 2.2 Example of an image classified into nine different classes.	9
Figure 2.3 Example of two classes separated or classified with mismatch assignment.	9
Figure 2.4 KNN density estimation.	17
Figure 2.5 Scaling function and wavelets from iteration of the lowpass filter [51].	20
Figure 2.6 Representations of signals by different techniques, from reference [55].	21
Figure 2.7 Representation of a sine wave and a wavelet (db 10), from reference [55].	22
Figure 2.8 Wavelet decomposition tree, from reference [55].	22
Figure 2.9 Wavelet decomposition tree.	23
Figure 2.10 Two-Dimensional Discrete Wavelet Transform	23
Figure 2.11 Discrete Wavelet Transform Example	24
Figure 3.1 Images used from the Brodatz album.	32
Figure 3.2 Sampling selection example of cork image from Brodatz album.	33
Figure 3.3 Example of the order of the 64 samples in the 256 by 256 cork image	33
Figure 3.4 Order of arrangement of samples and image.	35
Figure 3.5 Array arrangements of the new data.	36
Figure 3.6 Complete feature extraction, classification and fusion algorithms	38
Figure 3.7 Classification algorithm	39
Figure 4.1 Mayagüez Bay image	66
Figure 4.2 Majority Voting Cork image	68
Figure 4.3 Majority Voting French image	69
Figure 4.4 Majority Voting Liz image	69
Figure 4.5 Majority Voting Net image	70
Figure 4.6 Majority Voting Repskin image	70
Figure 4.7 Majority Voting Scloth image	71
Figure 4.8 Majority Voting Straw image	71
Figure 4.9 Majority Voting Strcloth image	72
Figure 4.10 Majority Voting Water image	72
Figure 4.11 Majority Voting Weave image	73
Figure 4.12 Histograms of Cork sample approximation and diagonal coefficients	82
Figure 4.13 Histograms of French sample approximation and diagonal coefficients	83
Figure 4.14 Histograms of Liz sample approximation and diagonal coefficients	83
Figure 4.15 Histograms of Net sample approximation and diagonal coefficients.	84
Figure 4.16 Histograms of Repskin sample approximation and diagonal coefficients	84
Figure 4.17 Histograms of Scloth sample approximation and diagonal coefficients	85
Figure 4.18 Histograms of Straw sample approximation and diagonal coefficients	85
Figure 4.19 Histograms of Strcloth sample approximation and diagonal coefficients	86

Figure 4.20 Histograms of Water sample approximation and diagonal coefficients.....	86
Figure 4.21 Histograms of Weave sample approximation and diagonal coefficients	87
Figure 0.1 The decomposition low-pass filter for ‘db1’	119
Figure 0.2 The decomposition high-pass filter for ‘db1’	119
Figure 0.3 The decomposition low-pass filter for ‘haar’	120
Figure 0.4 The decomposition high-pass filter for ‘haar’	121
Figure 0.5 The decomposition low-pass filter for ‘coif1’	122
Figure 0.6 The decomposition high-pass filter for ‘coif1’	123
Figure 0.7 The decomposition low-pass filter for ‘sym2’	124
Figure 0.8 The decomposition high-pass filter for ‘sym2’	125
Figure 0.9 The decomposition low-pass filter for ‘bior1.1’	126
Figure 0.10 The decomposition high-pass filter for ‘bior1.1’	127
Figure 0.11 The decomposition low-pass filter for ‘rbio1.1’	128
Figure 0.12 The decomposition high-pass filter for ‘rbio1.1’	129

1 INTRODUCTION

The two most important steps in pattern recognition systems are feature extraction and classifier design. Pattern recognition is the process of identifying objects or classes in an image. Feature extraction can be defined as the conversion of a set of measurements in order to form a subset that can be useful in a classification module. The process of making quantitative measurements from an image in order to produce a description of it is known as image analysis.

Texture is fundamental to perception, is a very useful characteristic when we want to analyze satellite images, biomedical, object surfaces, among others. A precise quantitative definition is required in order to be able to handle texture in a computer. Texture is intrinsically related to the gray level properties of pixels as well as their spatial relationships; wavelet transform has the property of increased spatial and frequency resolutions [2].

1.1 Motivation

We can date the use of the remote sensing techniques to the beginning of the space age, which began with the launch of Sputnik by the Soviet Union in October 1957. This has stimulated the development of remote sensing technology for space and aircraft systems, to collect information needed to manage land resources. The problem of going to higher altitudes to collect information was the quantities of data to be dealt with, thus

different ways of analyzing the data to extract the desired information are still been develop.

The economy is the primary driver for the development of technology, if we need information about large areas; we have to find ways to make it less expensive, the processing of large amount of data collected by high resolution sensors is not the way to achieve that. This problem can be solved by using lower resolution, that means less data, sensors less expensive, lighter in weight and requiring less precise pointing, but able to provide the desired information. This led the researchers to use spectral characteristics and pattern recognition methods for deriving labels for each pixel.

The subject of a lot of research works have always been texture analysis and is fundamental in many applications. Wavelet transform is used for characterizing and to represent textural images and its properties in multi-resolution or multi-channel analysis. What makes the wavelet transform an attractive method for analyzing data is the ability to examine different scales and time simultaneously in a time scale joint representation. The 2D wavelet transform can be considered as a 2D filtering process of an image by a low-pass filter and three high-pass filters, followed by a down sampling by two applied on filtered images [5].

According to reference [3], by having the second or higher order statistics of the wavelet coefficients, numerical description is achieved for texture characterization. In reference [5] they claim that orthogonal wavelet decomposition is complete at any scale, unlike the Fourier decomposition of a signal. In texture characterization, as the Fourier coefficients decrease at high frequencies, no information is brought for the case of

wavelet coefficients if the set of low-pass sub-image coefficients tends toward their means or more clearly if their variances tend towards zero. The evolution of the variance and variance values depend on the size of texture patterns and on texture variability.

In the past decade classifier combination has received considerable attention as a pattern recognition offspring. A lot of research has been made in order to explain why combination methods and strategies work so well. The central idea is that by combining the individual outputs, we aim at a higher accuracy than that of the best classifier, but to gain such accuracy we need a diversity of classifiers and an appropriate fusion method that can improve further on the performance of the team; the most obvious choice when constructing a multiple classifier system is a simple fusion method. In our case we will define fusion as an integration of the results obtained by the discriminant functions from different classifiers.

1.2 Problem Formulation and Proposed Solution

The general problem is to identify the class of an object or sample by a proper analysis given a set of samples and measurements, belonging to a set of predefined classes. We want to design image analysis systems with the ability to extract useful information from an irrelevant detail background that can be trained, and capable of making deductions or getting to conclusions without complete information. We want this system to implement optimal decisions, but for real problems this systems are not able to achieve 100% of correct classification.

The feature extraction process is critical for the classification of textured images. Sometimes different images have important features in common; thus a new problem is presented, the need of other features to make a decision, or classification. In our case, we are using textured images from the Brodatz album. These images are single-scale, which means that we have limitations regarding the features that we can extract from them.

Classification may not be as simple as the case of this diagrams, see Figure 2.1; because:

1. Usually the classes desired have distributions in the feature space that are not obviously separated.
2. It is necessary to use more than three features. Thus one cannot see what the class distribution look like.

The most common way to determine the location of the decision boundaries required is by using discriminant functions.

As a proposed solution for these problems we will use the advantage of the wavelet transform which introduces multiple scales characterization of texture properties. As said before, wavelet transforms has the property of increased spatial and frequency resolutions. Therefore we use wavelet transform to extract features from each scale or resolution cell. By doing this we have the advantage of adding independent local textural features which will increase the class separation while reducing computational loads.

For the case of the classifiers, we will use three different classifiers of our choice, and a fusion rule that will combine the results obtained by each individual classifier. The use of algorithms based on wavelet transforms and fusion methods is the solution that we propose in order to solve the problems of classification accuracy. We are confident that

with the use of the fusion methods we will aim a better overall result than that of any single classifier.

1.3 Contribution

The contribution of this project is that it presents an application of the wavelet Transform as a feature extraction method and fusion methods as a way to improve the classification accuracy that have not been made. Other research only had use single classifiers to classify the features extracted using wavelets and we will be using fusion methods for that case or vice versa. We will compare our results using single classifiers with the ones obtained using fusion methods. We will test the algorithms with different images from the Brodatz texture album. Feature selection and other recommendations will be proposed based on the results obtained in the cases of studied in this thesis.

1.4 Summary of Following Chapters

In Chapter 1 we give a brief introduction of the theory and motivation as the necessary background. Chapter 2 deals with the theoretical background more extensively related to classification, fusion methods and wavelets, including commentaries about previous works. Chapter 3 presents the methodology, data used, statement of the problem, results and analysis. In Chapter 4 we present the conclusions and future works.

2 THEORETICAL BACKGROUND

2.1 Pixel Definition

Discrete picture elements or Pixel can be defined as the elements that compose spatially the image data when it is available in digital format.

2.2 Texture Definition

Dr. Vidya Manian defines texture as the structural pattern of surfaces such as wood, grain, grass, which is homogeneous in spite of fluctuations in brightness and color. It is the most important visual cue in identifying regions and contributes in a number of ways to human vision-segmentation, shape detection and object recognition.

2.3 Texton Definition

Julesz introduced the term texton, as the putative units of preattentive human texture perception more than 20 years ago. He described them qualitatively for simple binary line segment stimuli-oriented segments, crossings and terminators. Texton theory fell into disfavor as a model of human texture discrimination as accounts based on spatial filtering with orientation and scale-selective mechanisms which could be applied to arbitrary gray-level images became popular [57].

2.4 Class Definition

The American Heritage dictionary defines class as a set, collection, group, or configuration containing members regarded as having certain attributes or traits in common; a kind or category. A division based on quality, rank, or grade. In our case think of it as different groups available in an image that we want to differentiate them from each other.

2.5 Classification

Quantitative analysis is the computer interpretation of remote sensing image data. It is also the ability to identify pixels based upon their numerical properties. It's generally called classification, the method where pixels will be attached to labels depending on their spectral character. A computer can implement this labeling by having trained it beforehand to recognize pixels with spectral similarities [52].

We want to one to have a feature space divided into a set of non-overlapping regions for each class of interest, in order that each point in that space will be associated uniquely to one of the desired classes, this implies establishing "decision boundaries". See Figure 2.1 for an example of decision boundaries [50].

There are two different procedures of doing classification and both have areas of application in the analysis of the data obtained for image classification. These procedures are:

1. *Unsupervised classification*: in this procedure the pixels in an image are labeled to a spectral class without the user having foreknowledge of the existence or names of those classes. Clustering methods is the most common way to perform this procedure. It is used to determine the spectral classes of each pixel and the number and location of the spectral classes into which the data falls.
2. *Supervised classification*: for extracting the quantitative information from remotely sensed image data this is the essential analytical tool. Each spectral class can be described by a probability distribution in multispectral case. The most valuable distribution is the normal or Gaussian that gives tractable mathematical descriptions of the supervised classification process. We will present in detail in the following sections.

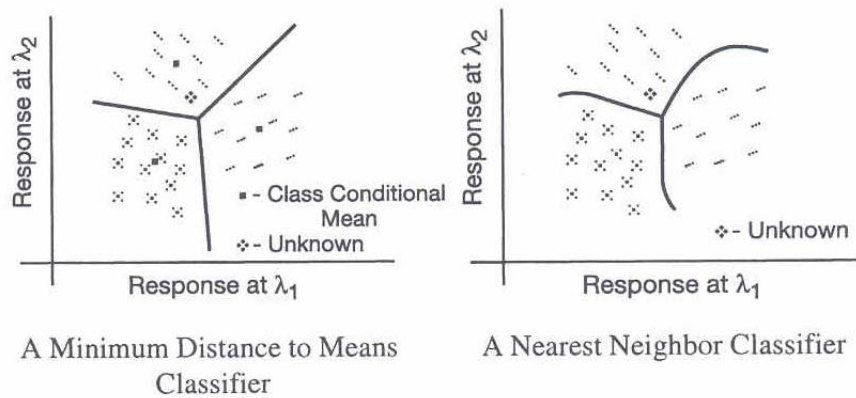


Figure 2.1 Simple example classifier schemes, from reference [50].

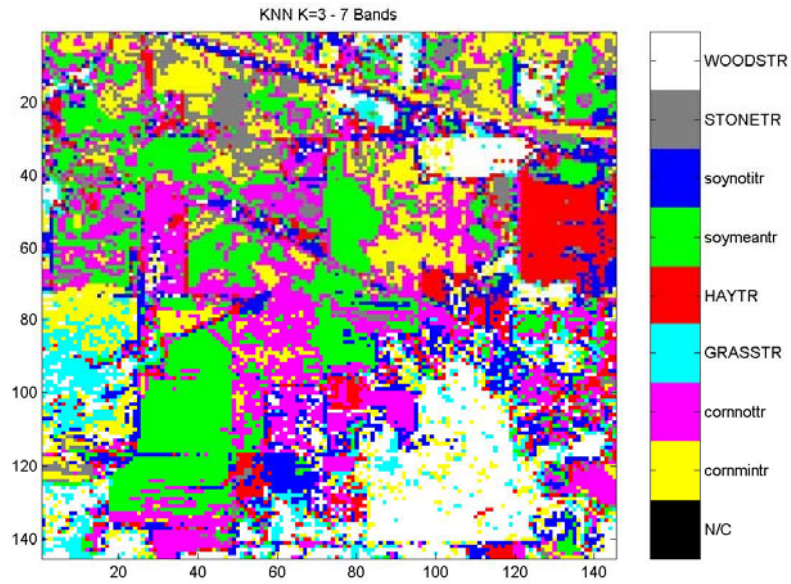


Figure 2.2 Example of an image classified into nine different classes.

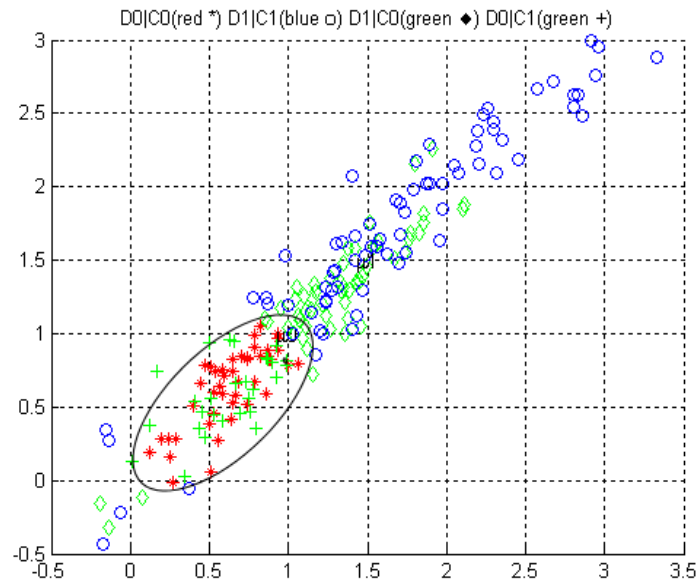


Figure 2.3 Example of two classes separated or classified with mismatch assignment.

Imagery-based and spectrally-based systems are the two bases for the Sensor and Data Systems where images gathered from spacecrafts are considered as an important vehicle for collecting information. If the image-based systems were obtained at a high spatial resolution it can be considered of the most immediate use because the users are able to see the details of interest in the scene. Due to the cost to identify for example a vegetative species using spatial resolution, spectrally-based methods would function best using less resolution, thus reducing the volume of data and processing cost.

Information is potentially available at altitude from energy fields arising from the Earth's surface and in particular from the Spectral, Spatial and Temporal variations of those fields, this is provided by the Information theoretic principles. To capture the information, one must measure the variations of the electromagnetic and gravitational energy fields, and relate them to the information desired.

The part of the system that collects the main body of the data to be used is the sensor. There are two types of sensors passive and active. The passive sensors only gathers information using the sun as an illumination source, the active sensors produces its own source that will reflect on the scene. The most common used are passive optical systems, active microwave (radar) and active optical (lidar) systems. The processing system is where the data analyst is able to exercise the most control and have the most choices.

The energy arising from the Earth is finite in magnitude; the data collection process must divide it spatially into pixels. Each pixel has a power level that can be

divided up into a number of spectral bands. There is a tradeoff given the finite nature of the field, because one moves to finer spatial resolution and spectral band intervals, less power is left to overcome the internal noise in evidently present in the sensor system [50].

Spatial resolution, spectral resolution and signal to noise ratio are the fundamental parameters of this information-gathering process, and they are all interrelated. The motivation is to avoid the need for very high spatial resolution because the power level decreases and the volume of data that must be dealt with increases both as the square of the spatial resolution, but only linearly with spectral resolution.

We can represent the data in three different spaces where each of them has its advantages and disadvantages:

1. Image space: useful to locate geographically where a pixel is from, but it's only effective for two or at most three dimensions or spectral bands.
2. Spectral space: useful relating a pixel's response to the physical basis for any number of spectral bands, but is difficult to display the variations in a spectral space.
3. Feature space: possible to display all of the information mathematically and computation, but it cannot be shown graphically for human view.

2.6 Discriminant Functions

Because of the difficulty to define the decision boundaries in the feature space and the computational problem of determine on which side of a boundary a given unknown pixel falls, an easier way is formulated, and it is called Discriminant Function. The

Discriminant Function is a collection of functions that allows us to determine by evaluating them at certain point, a value that would make easier the classification process. Assume that we are able to find a set of m functions $\{g_1(\mathbf{X}), g_2(\mathbf{X}), \dots, g_m(\mathbf{X})\}$, such that every time that $g_i(\mathbf{X})$ is larger than the other functions, is because \mathbf{X} belongs to class i . This helps to find an appropriate classification rule which is:

Let ω_i denote the i^{th} class.

Choose $\mathbf{X} \rightarrow \omega_i$ (\mathbf{X} is in class ω_i) if and only if:

$$g_i(\mathbf{X}) \geq g_j(\mathbf{X}) \quad \forall j = 1, 2, \dots, m$$

$$j \neq i$$

2.7 Supervised Classification

Is a procedure often used for quantitative analysis of remote sensing image data, in which the use of suitable algorithms based upon probability distribution models of the classes of interest are essential for labeling the pixels in an image. The steps to follow often in this particular method are:

1. Select the desire classes in the image.
2. Select representative pixels or training data from the classes selected.
3. Estimate the parameters of a particular classifier using the training data.
4. Use the trained classifier to label the entire image.
5. Tabulate the results of the classification.

2.8 Classifier Laws

To focus on our work, we will present some of the different types of classifiers by defining their discriminant functions.

2.8.1 Euclidean Distance (ED)

Let \mathbf{X} be the pixel vector, \mathbf{M}_i and Σ_i the mean vector and the covariance matrix of the class ω_i respectively, in this classifier we assumed that that the covariance matrices are equal to the identity matrix \mathbf{I} .

$$g_i(\mathbf{X}) = -(\mathbf{X} - \mathbf{M}_i)^T (\mathbf{X} - \mathbf{M}_i)$$
$$g_i(\mathbf{X}) > g_j(\mathbf{X}) \quad \forall j \neq i \longrightarrow x \in \omega_i \quad [54]$$

2.8.2 Fisher Linear Discriminant (FLD)

In this classifier we use the average of the covariance matrices or we assumed that they are equal, let m be the total number of classes.

$$\Sigma = \frac{1}{m} \sum_i \Sigma_i$$
$$g_i(\mathbf{X}) = -(\mathbf{X} - \mathbf{M}_i)^T \Sigma^{-1} (\mathbf{X} - \mathbf{M}_i)$$
$$g_i(\mathbf{X}) > g_j(\mathbf{X}) \quad \forall j \neq i \longrightarrow x \in \omega_i \quad [54]$$

2.8.3 Minimum Mahalanobis Distance Classifier (MD)

In this classifier we use the covariance matrix of each class.

$$g_i(\mathbf{X}) = -(\mathbf{X} - \mathbf{M}_i)^T \Sigma_i^{-1} (\mathbf{X} - \mathbf{M}_i)$$

$$g_i(\mathbf{X}) > g_j(\mathbf{X}) \quad \forall j \neq i \longrightarrow x \in \omega_i \quad [54]$$

2.8.4 Maximum Likelihood Classifier (ML)

It is the most common supervised classification method used with remote sensing image data. This is developed in the following in a statistical acceptable manner [52].

Bayes Classification

Let the spectral classes for an image be represented by $\omega_i = 1, \dots, m$ where m is the total number of classes. In trying to determine the class or category to which a pixel at a location x belongs it is strictly the conditional probabilities.

$$p(\omega_i | x), i = 1, \dots, m \quad [52]$$

The probability $p(\omega_i | x)$ gives the likelihood that the correct class is ω_i for a pixel at a position x . $p(\omega_i | x)$ is called a posteriori or posterior probabilities.

Classification is performed according to

$$x \in \omega_i \text{ if } p(\omega_i | x) > p(\omega_j | x) \text{ for all } j \neq i \quad [52]$$

Maximum Likelihood Decision Rule

We can estimate from the training data available $p(x|\omega_i)$, then $p(\omega_i|x) = p(x|\omega_i)p(\omega_i)/p(x)$, $p(\omega_i)$ is called *a priori* or prior probabilities [52].

Then the classification rule can be seen as

$$x \in \omega_i \text{ if } p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j) \text{ for all } j \neq i$$

For mathematical convenience

$g_i(x) = \ln p(x|\omega_i) + \ln p(\omega_i)$, then is restated as $x \in \omega_i$ if $g_i(x) > g_j(x)$ for all $j \neq i$ [52]

For the Multivariate Models

$$g_i(X) = -\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) [52]$$

Maximum Likelihood with Threshold

If $-\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) > T$ [54], then accept, otherwise reject.

T = threshold

2.8.5 Maximum A Posteriori (MP)

It is the same as the ML classifier but with an A Priori probability.

$$g_i(X) = \ln [p(\omega_i)] - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) [54]$$

2.8.6 *K-Nearest Neighbor (KNN)*

K is constant (nearest neighbor), nearest neighbor assign the same label as that of the nearest training pixel. We calculate the distance, that can be Euclidean, Mahalanobis, etc. This classifier assign label according to the majority label of K -Nearest Neighbor training pixels.

$$f_n(x) = \frac{k/n}{V(x)} \quad k = \text{is constant (nearest neighbor)}$$

V = is variable

$$g_i(x) = f(x) = \frac{k_i/n_i}{V_i(x)}$$

$$V_i(x) = K \cdot \text{dist}_i^d(x) \quad d = \text{dimensionality}$$

$$g_i(X) > g_j(X) \quad \forall j = 1, 2, \dots, m \quad j \neq i$$

$$V_i(X) < V_j(X) \quad \forall j = 1, 2, \dots, m \quad j \neq i$$

$$\text{dist}_i^d(X) < \text{dist}_j^d(X) \quad \forall j = 1, 2, \dots, m \quad j \neq i$$

$$\text{dist}_i(X) < \text{dist}_j(X) \quad \forall j = 1, 2, \dots, m \quad j \neq i \quad [54]$$

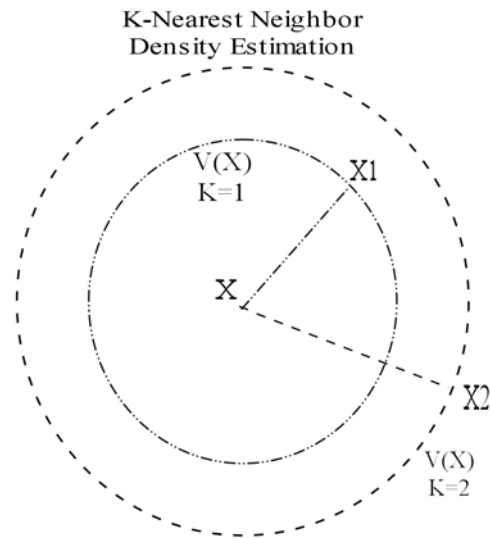


Figure 2.4 KNN density estimation.

2.8.7 Neural Networks (NN)

Perceptron convergence procedure:

1. Initialize weights and threshold. Set ω_i ($0 \leq i \leq d$) to small random values where ω_i is the weight for input i [54].
2. Present new input and desire output [54].
3. Calculate the actual output for the h^{th} training sample [54].
4. Adapt weights [54].

2.9 Max, Min, Average and Majority Voting Fusion Rules

To achieve the best possible classification performance for the task is the goal for every pattern recognition system. The two main reasons for combining classifiers are efficiency and accuracy. An important issue in combining classifiers is that this is particularly useful if they are different [53].

Let $i = \text{ith}$ group of bands (from 1 to N) and $k = \text{class number}$ (from 1 to M) [54]. Assume the probability density function of x_i given the class k is $p(x_i | \omega_k)$ [54]. Let x_i define the pixel's sensor measurements in the i th's group of adjacent bands and X be the pixel with all sensor's measurements [54]. The Max, Min, Average and Majority Voting fusion rules are defined as follows [54].

a) Max rule:

Assign X to ω_j if

$$\max_{i=1}^N p(x_i | \omega_j) = \max_{k=1}^M \max_{i=1}^N p(x_i | \omega_k)$$

b) Min rule:

Assign X to ω_j if

$$\min_{i=1}^N p(x_i | \omega_j) = \max_{k=1}^M \min_{i=1}^N p(x_i | \omega_k)$$

c) Average rule:

Assign X to ω_j if

$$\frac{1}{N} \sum_{i=1}^N p(x_i | \omega_j) = \max_{k=1}^M \left[\frac{1}{N} \sum_{i=1}^N p(x_i | \omega_k) \right]$$

d) Majority Voting rule:

$$\text{Let } \Delta_{k_i} = \begin{cases} 1, & \text{if } p(x_i | \omega_k) = \max_{k=1}^M [p(x_i | \omega_{kk})] \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Assign } X \text{ to } \omega_j \text{ if } \sum_{i=1}^N \Delta_{ji} = \max_{k=1}^M \sum_{i=1}^N \Delta_{ki}$$

The classification fusion center can be based on supervised schemes using the output of the local classifications as the new features [54]. For a nonlinear separable problem that includes multiclass, a multilayer neural network must be used [54].

2.10 Wavelet Analysis

2.10.1 Wavelet Definition

Wavelets are localized waves, that instead of oscillating for ever, they drop to zero [51]. They come from the iteration of filters (with rescaling) [51]. The link between discrete-time filters and continuous-time wavelets is in the limit of a logarithmic filter tree:

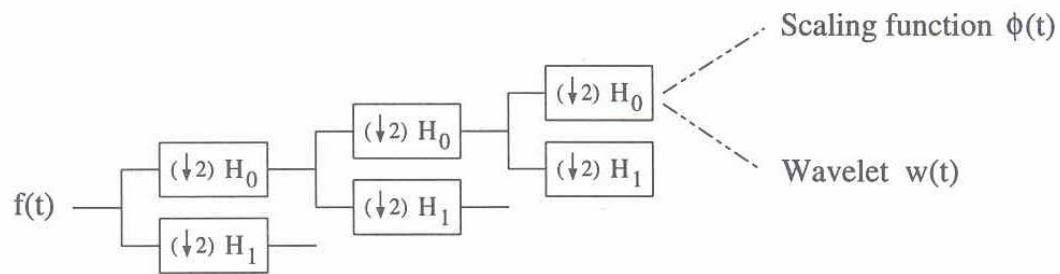


Figure 2.5 Scaling function and wavelets from iteration of the lowpass filter [51].

2.10.2 Wavelet Analysis

For analyzing signals, perhaps the most well known tool is Fourier analysis, which is the breaking down of a signal into constituent sinusoids of different frequencies for transforming our view of the signal from time-based to frequency-based. If we are analyzing non-stationary signals time information is lost when transforming to frequency domain. That's why D. Gabor (1946) develop a technique called *windowing* the signal, that maps a signal into a two-dimensional function of time and frequency, called the *Short-Time Fourier Transform (STFT)*.

The only problem was that the size of the window was the same for all frequencies once you choose a particular size, and some times we need a more flexible approach. Wavelet analysis is a windowing technique with variable-sized regions [55]. As you can notice in Fig. 2.2, it uses a time-scale region instead of time-frequency like the STFT, which gives more flexibility when looking at low or high frequency.

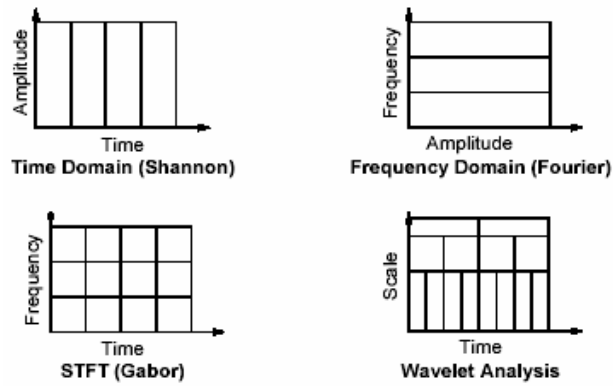


Figure 2.6 Representations of signals by different techniques, from reference [55].

The common goals concern the signal or image clearance and simplification, which are parts of de-noising or compression [55]. A wavelet is a waveform of effectively limited duration that has an average value of zero [55]. Wavelets tend to be irregular and asymmetric, unlike sinusoids that are smooth, predictable and do not have limited duration; see Fig. 2.7 for an example. Wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or *mother*) wavelet. In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components [55].

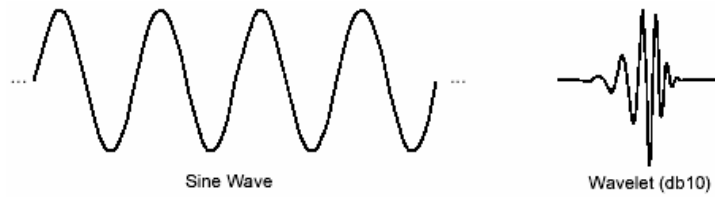


Figure 2.7 Representation of a sine wave and a wavelet (db 10), from reference [55].

2.10.3 Multiple-Level Decomposition

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called the *wavelet decomposition tree*, see Fig. 2.8. Looking at a signal's wavelet decomposition tree can yield valuable information.

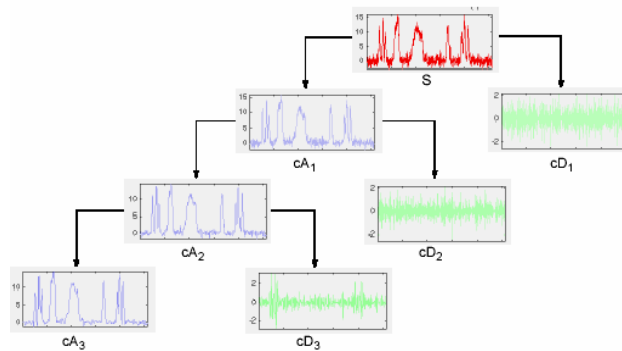


Figure 2.8 Wavelet decomposition tree, from reference [55].

It is possible to reconstruct our original signal from the coefficients of the approximations and details, see Fig. 2.9.

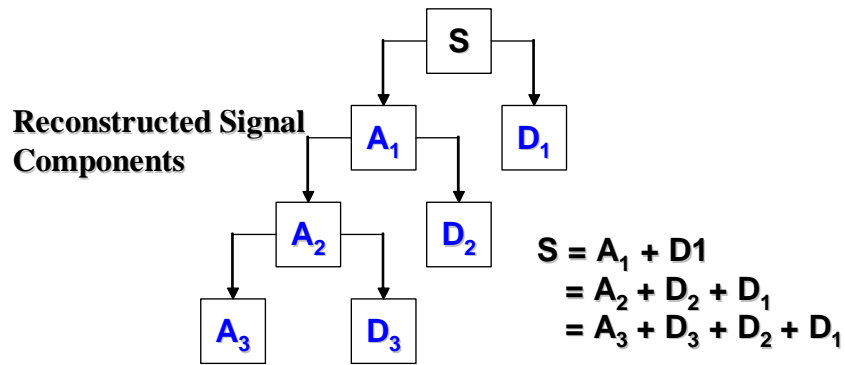


Figure 2.9 Wavelet decomposition tree.

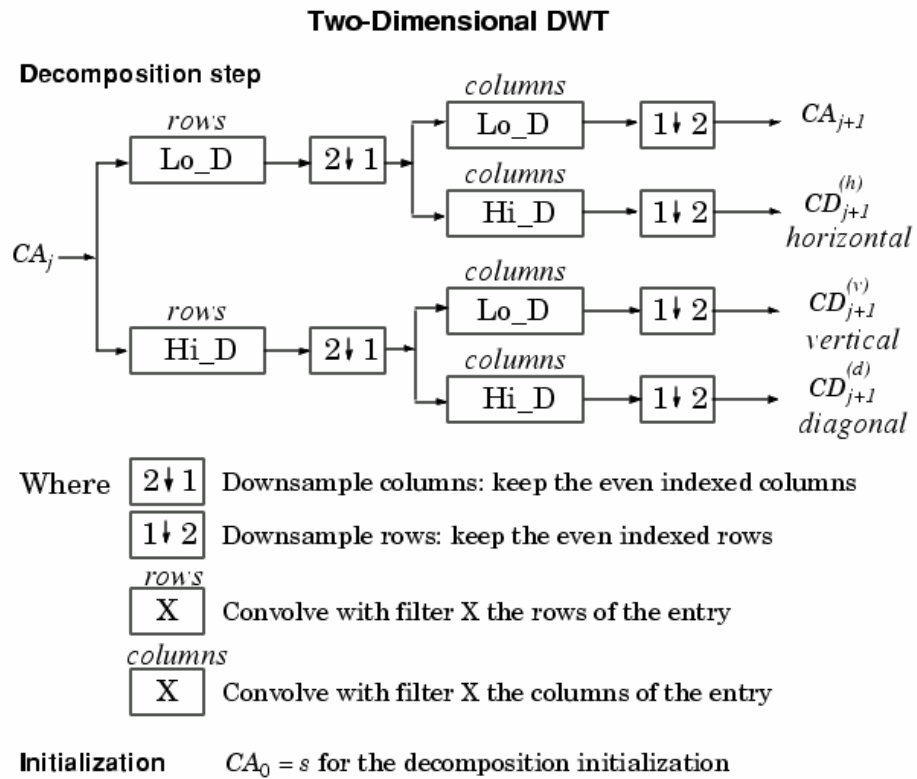


Figure 2.10 Two-Dimensional Discrete Wavelet Transform

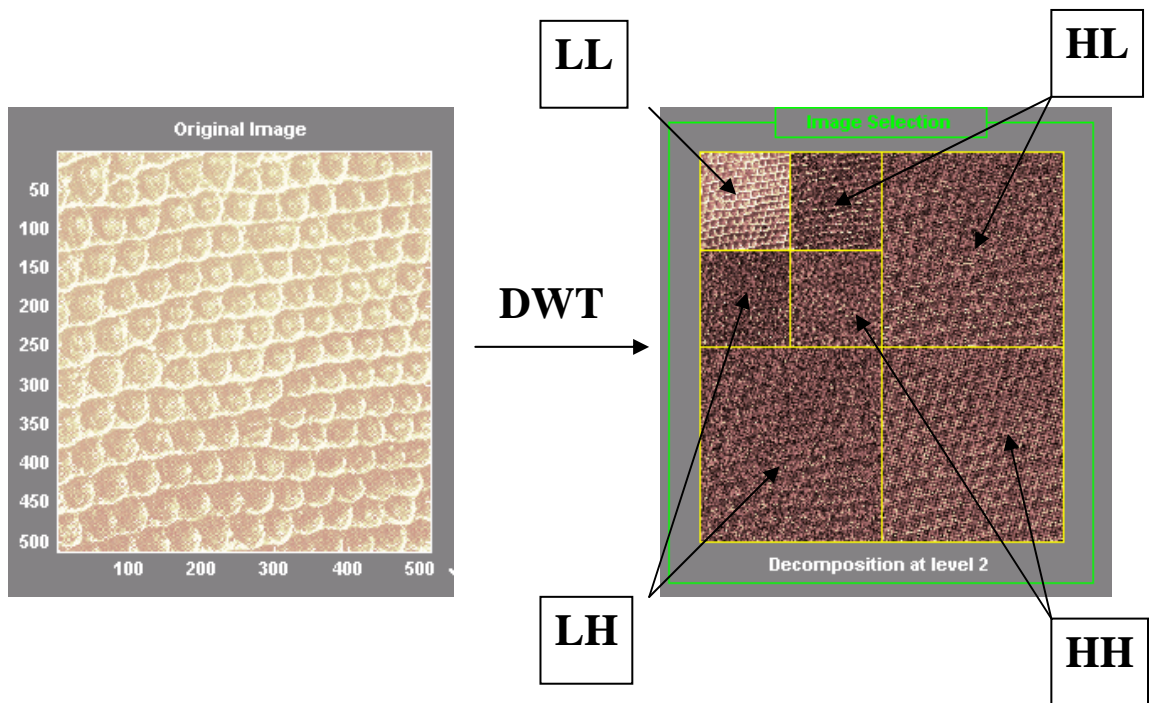


Figure 2.11 Discrete Wavelet Transform Example

2.11 Previous Work

After all the brief definitions we are going to present the most important previous work related to this project.

Texture Classification using Wavelet Transform This is from reference [2]. Here they use wavelet transform to represent the textural images in multiresolution. They compute Co-occurrence matrices to obtain textural features, from the different scales of the wavelet transform. Julesz was the first to use co-occurrence statistics in the study of human texture discrimination experiments, according to him, texture is a second order

process [2]. In this project they denote $P(i, j, d, \theta^\circ)$ as the unnormalized co-occurrence matrix, and it becomes a second-order probability distribution, when P is normalized by the total number of possible pairs of pixels with specified separation and orientation.

They claim that Maximum Likelihood approach to texture classification is optimal from a statistical decision theoretic point of view, in the sense that it results in the least classification errors. The size of the co-occurrence matrix corresponds to the number of gray levels in the texture [2]. Because wavelet transform has the property of increased spatial and frequency resolutions, they use it to represent the textural images in multi-resolutions and extract features from each scale. After this feature extraction they use they use a Maximum Likelihood estimator, assuming has known the probability density function of the feature vector.

They use one level and one dimensional wavelet transform in a row-column fashion to obtain the lowpass and detail images of all textural images. Next, they obtained four images low-low, high-low, low-high and high-high. The low-low image contains all the essential features of the original image in a lower resolution, and the high frequency components contain the edge details useful in the textural description.

In reference [3] they use wavelet transform for description of image texture. The objective is to design a wavelet based feature extraction algorithms that will measure texture quality along the most important perceptual dimensions [3]. They quantify texture the way human see them as coarse, regular, directional, symmetric, etc. An essential part

in computer graphics is texture mapping, for improving the rendering quality of a 3D environment. They propose an algorithm that can also be used to test texture replicability in computer graphics applications. After the wavelet packet decomposition, feature extraction is performed in the selected subbands. The objective of the feature extraction is to compute a vector which contains the information about directionality, regularity and symmetry features.

In reference [4] the use of wavelet decomposition to detect and analyze time-varying signals, is their basic tool. In [5] they present an algorithm based on conventional pyramid-structured wavelet transform, which provides several important characteristics that can be used for the texture classification problem. The 2D wavelet transform can be considered as a 2D filtering process of an image by a lowpass filter $H(k)$ and three highpass filters $H_1(k)$, $H_2(k)$, $H_3(k)$ followed by a downsampling by two applied on filtered images, which lead to the decomposition of the input image into four subimages at lower resolution [5]. They calculate the form factor which is a texture feature that has a multiscale structure. Orthogonal wavelet is complete at any scale; unlike Fourier decomposition were the coefficients decrease appreciably at high frequencies.

For the case of reference [6], wavelets and neural networks are combined to adaptively compute a super wavelet filter that is optimized for feature extraction and classification. Extracting adaptive wavelet features from the signals estimated by modeling them using the nonlinear approach is the first step. They chose a Morlet

wavelet as a mother wavelet for modeling the sonar signals.

In [8], the approach that they use is a over-complete wavelet decomposition called wavelet frames which yields the descriptions of both translation invariance and stability. In their experimental studies they have found that the most significant information often appears in the middle channels.

In reference [59], they create a directional, nonredundant, complex wavelet transform with potential benefits for image coding systems, thanks to a balance between degree of shift sensitivity and transform redundancy that the mapping stage offers. This is a multidimensional, mapping-based, complex wavelet transforms. It consists of a mapping onto a complex function space followed by a DWT of the complex mapping. Their approach has the flexibility to use any DWT in the transform implementation.

For the case of reference [60], they propose a wavelet-based multiscale linear minimum mean square-error estimation (LMMSE) scheme for image denoising. They also discussed the determination of the optimal wavelet basis with respect to their proposed scheme. They combined the pixels at the same spatial location across scales as a vector and applied LMMSE to the vector in order to explore the strong interscale dependencies of overcomplete wavelet expansion.

In reference [61], they propose a new approach for the estimation of wavelets that is matched to a given signal in the statistical sense. The key idea of their work lies in the estimation of analysis wavelet filters from a given signal. They applied minimum mean square error (MMSE) criterion to minimize the error.

Reference [62] authors developed an efficient fuzzy wavelet packet (WP) based feature extraction method for the classification of high-dimensional biomedical data. In order to find optimal WP decomposition for signal classification, they mapped the original signals to many WP feature spaces. Then they did feature extraction based on the optimal WP decomposition and used a linear classifier to do the signal classification. Their extraction algorithm constructs fuzzy sets of features to evaluate their discriminatory effectiveness.

The authors of reference [63] use wavelet-based statistical models and hyperspectral soil signatures in an automatic soil texture classification system. They say that soil texture classification systems are closely related to texture classification methods, where images are used for training and testing. Maximum likelihood and hidden Markov models are used for the classification and to get the properties and information about the soil textures they developed a novel system using hyperspectral soil textures. They say that their methods are robust and reliable.

3 METHODOLOGY

We will perform an analysis of the methods proposed. By using Matlab we will simulate feature extraction and the classification of the textures. A more detailed procedure is presented next:

1. Literature review and selection of the most relevant research.
2. Selection of the images to be used for our purpose from the Brodatz texture album.
3. Training and testing samples selection of the images.
4. Feature extraction using wavelet transform.
5. Choose different classifiers to obtain different approaches.
6. Training of the classifiers to obtain the best set of training samples for testing the classifiers.
7. Comparison of results obtained in several classifiers.
8. Fusion of the classifiers results.
9. Analysis of the results to determine if the classification was improved.

For our work because of the amount of filters used we will focus only at second level decomposition coefficients acquired from the filtering process. We've taken special interest in the coefficients given by the approximation or the Low-Low results, because in

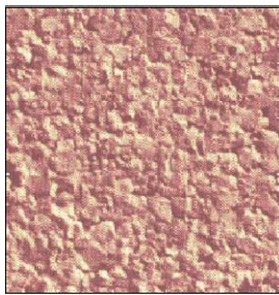
the experiments that we've made they were the ones with the best results and best performance in the classification process. Also the classifiers that we used were Maximum Likelihood, Euclidean Distance and K-Nearest Neighbor for $K=3$ and $K=5$. The fusion rule that we chose was Majority Voting that was applied to the results obtained by the classifiers.

3.1 Sample selection

3.1.1 Images used

The images that we used are from the Brodatz textures album, we will present the images used below.

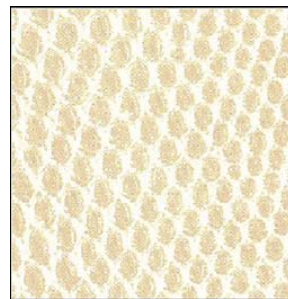
d4 cork



d21 french



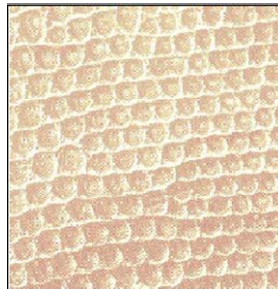
d36 liz



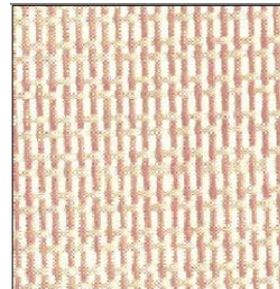
d34 net



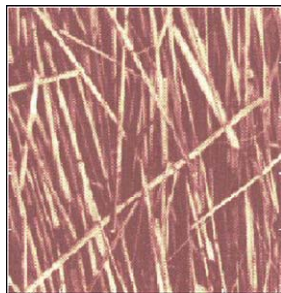
d22 repskin



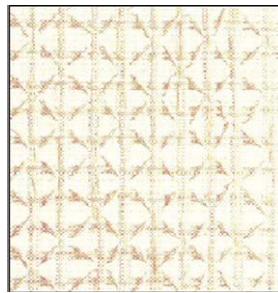
d53 scloth2



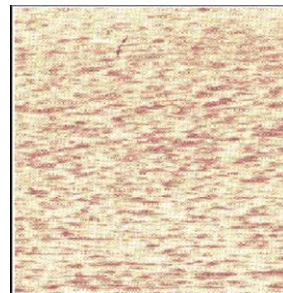
d15 straw



d52 strcloth



d38 water



d16 weave

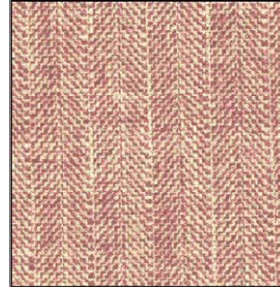


Figure 3.1 Images used from the Brodatz album

3.1.2 Image Sampling

We will start by explaining how we took the samples from the image. First we took 10 images from the Brodatz album. Some of the images were 256 by 256 and others were 512 by 512. With the images that were 512 by 512 we created a window of 256 by 256 and we used that part of the image for sampling. We needed to have same size of images to make it easier and to have the same amount of samples. The model of image sampling is presented below.

First we take samples of the image using a window of 32 by 32. This window moves around the images starting from the top left corner and going down in the first 256 rows by 32 columns without overlapping the ones that already were sampled, so we take eight samples out of each sweep; when it reaches the end it moves to the top and left to the next 256 rows by 32 columns and start again taking 32 by 32 samples all over, see Figure 3.1 and Figure 3.2 for a graphical review. This will be our testing and training samples.

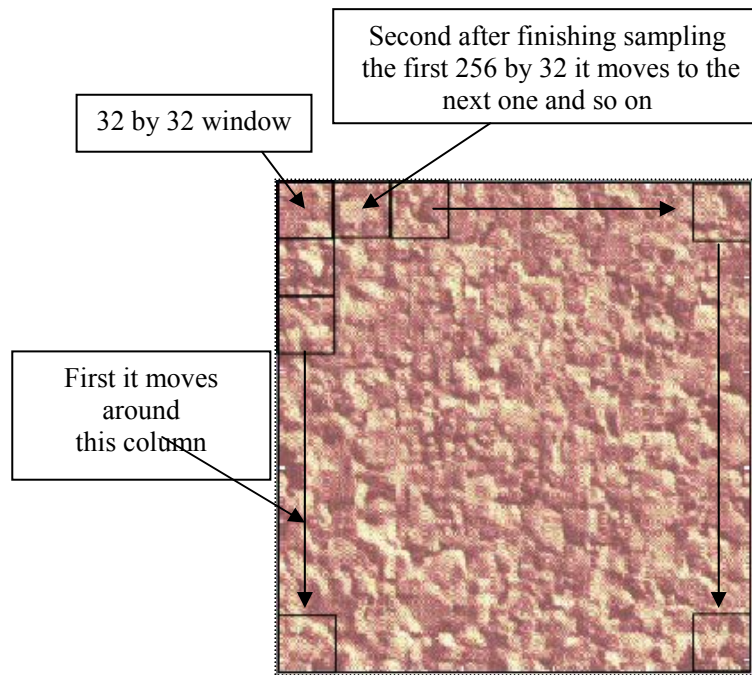


Figure 3.2 Sampling selection example of cork image from Brodatz album.

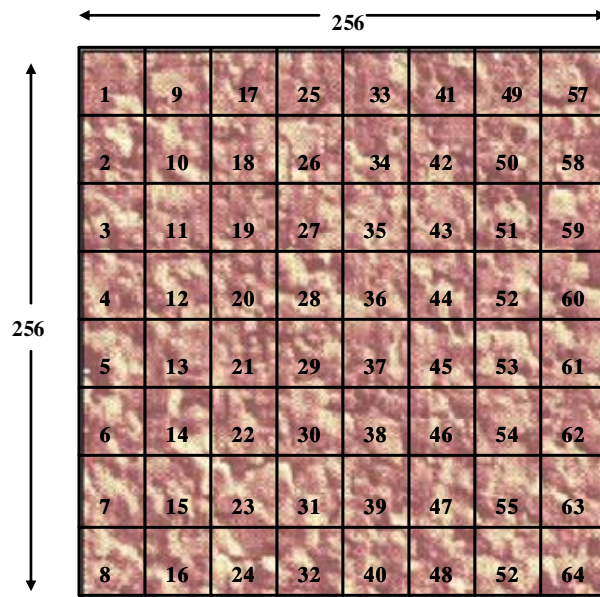


Figure 3.3 Example of the order of the 64 samples in the 256 by 256 cork image

For the case of the samples of 64 by 64 when the size of the image was 512 by 512 we used the same approach of the image sampling explained before for the 32 by 32 sample size. This means that we selected the samples without overlapping the windows of 64 by 64 and we got 64 samples from it. But for the case when the image was of 256 by 256 we used another method. In this case we first moved the window from top to bottom each 16 rows that gave us 13 samples per column and from left to right without overlapping gave us 4 samples per row, so 4 by 13 it's a total of 52 samples. We needed 12 more samples and we got them by moving the window from top to bottom without overlapping moving each 64 rows that gave us 4 samples per column and starting 32 columns from left to right without overlapping again moving each 64 columns that gave us 3 samples per row, so 4 by 3 it's a total of 12 more samples giving us the total of 64 samples.

3.1.3 Data arrange

After having 64 samples of each image we arrange them. The arrangement starts by converting every 32 by 32 sample into column vectors. We take each column from left to right and place it under the first one in order. Afterwards we transpose the column vector and turn it into a row vector. We do the same thing to all of the 32 by 32 samples and we place them underneath each other in the order that we take them, see Figure 3.3 for reference.

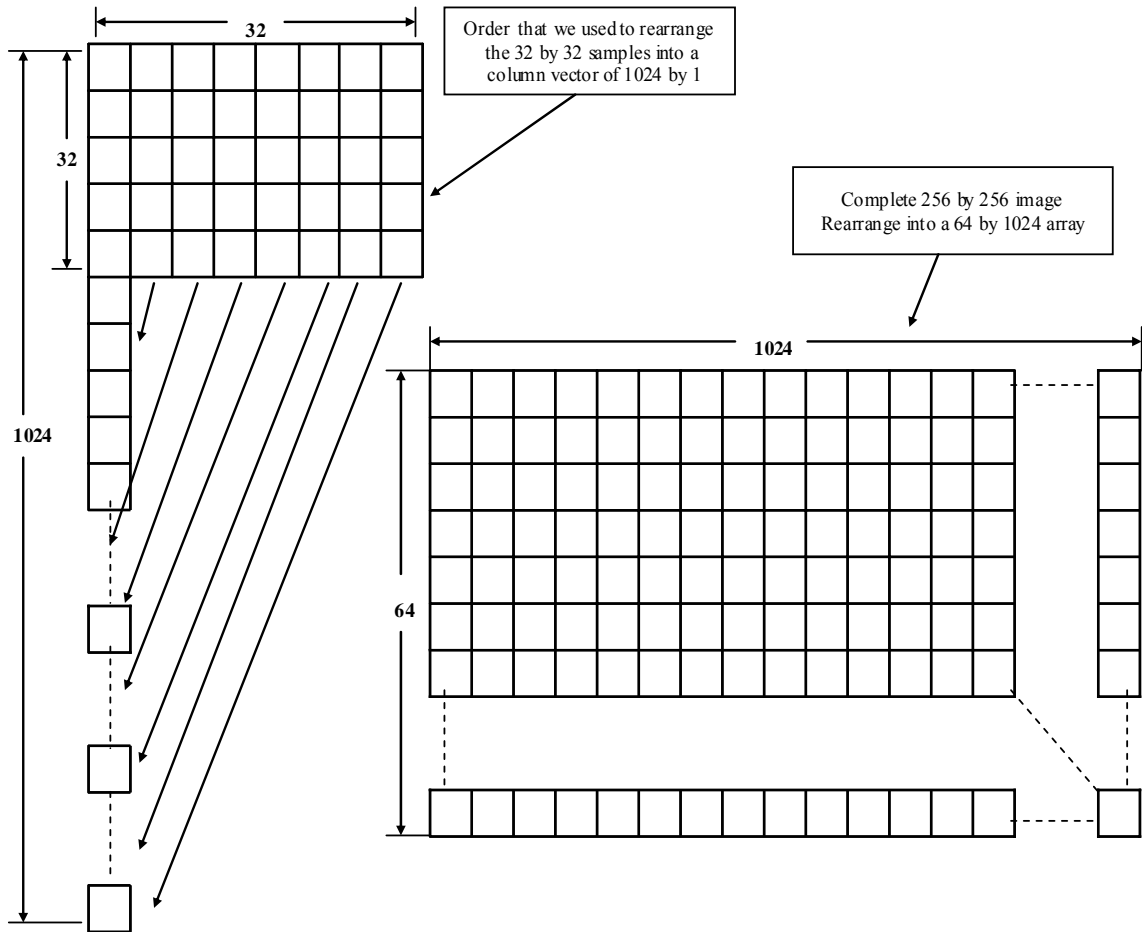


Figure 3.4 Order of arrangement of samples and image

3.1.4 Training and Testing Samples Selection and Data Filtering

The image sampling and data arrangement was only a way to facilitate the training and testing selection. We are going to do this process randomly. In the process of selecting which are going to be training or testing samples we are going pass the data through the wavelets. When we filter the samples we take from them the features that we

need for the classification process. These features are the second level decomposition coefficients plus the mean and the covariance of them. Now instead of the original data of the 32 by 32 samples, we now have a new array of 4 by 69 that have a column vector of 4 by 1 of the means of approximation, horizontal, vertical and diagonal coefficients, ns followed by a matrix of 4 by 4 of the covariance and one more matrix of 4 by 64 of the coefficients of the second level decomposition were each row is a coefficient in the next order first row approximation, second horizontal, third vertical and the last one is diagonal coefficients. The order that we arrange them is approximation coefficients, horizontal detail, vertical detail and diagonal detail, see Figure 3.4.

mean	cov	coefficients
4 by 1	4 by 4	4 by 64

Figure 3.5 Array arrangements of the new data

In the filtering process we used different filters for obtaining different data and features from the same images. In order to maintain certain similarity between the data obtained by each filter, we only took randomly the first set of training and testing samples. This means that for each image we randomly selected which samples were training and testing, generating a vector containing this order. Then we used the same vectors that we generated for each image in the processes with other filters. The filters that we used were the following:

1. Daubechies: 'db1', 'db2'.
2. Haar: 'haar'.
3. Coiflets: 'coif1'.

4. Symlets: 'sym2'.
5. Biorthogonal: 'bior1.1', 'bior3.1'.
6. Reverse Biorthogonal: 'rbio1.1', 'rbio3.1'.

For a more detail information on the filters used, see the Appendix B3 wavelet filters.

3.2 Algorithms

A complete feature extraction, classification and fusion algorithms will be presented in Figure 3.18. Then we will continue with a detailed feature extraction algorithm explaining the first steps of image sampling, data arrangement, filtering and sample selection. Then we will present the classifiers algorithms and the fusion algorithm used for our research, see Figure 3.19.

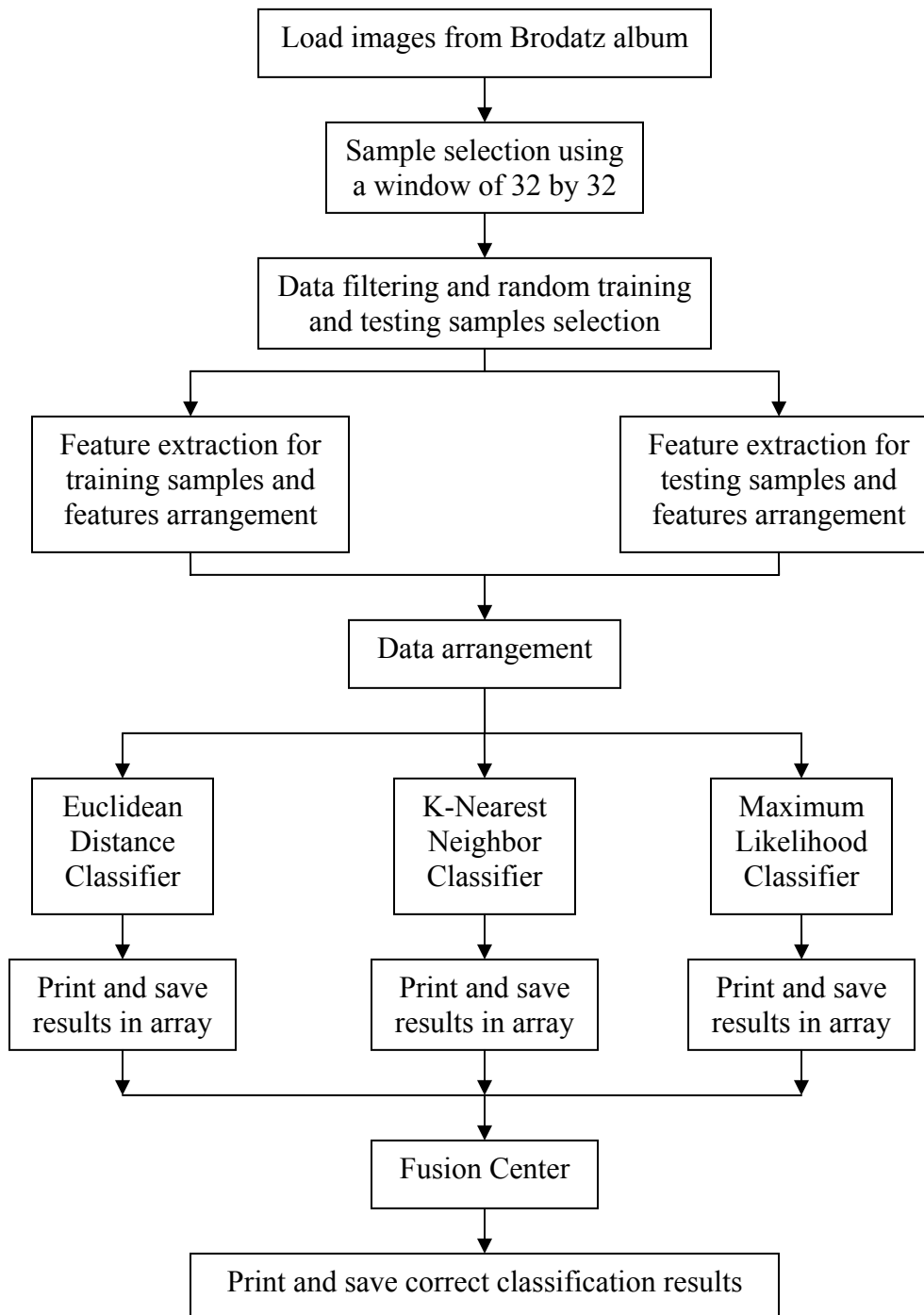


Figure 3.6 Complete feature extraction, classification and fusion algorithms

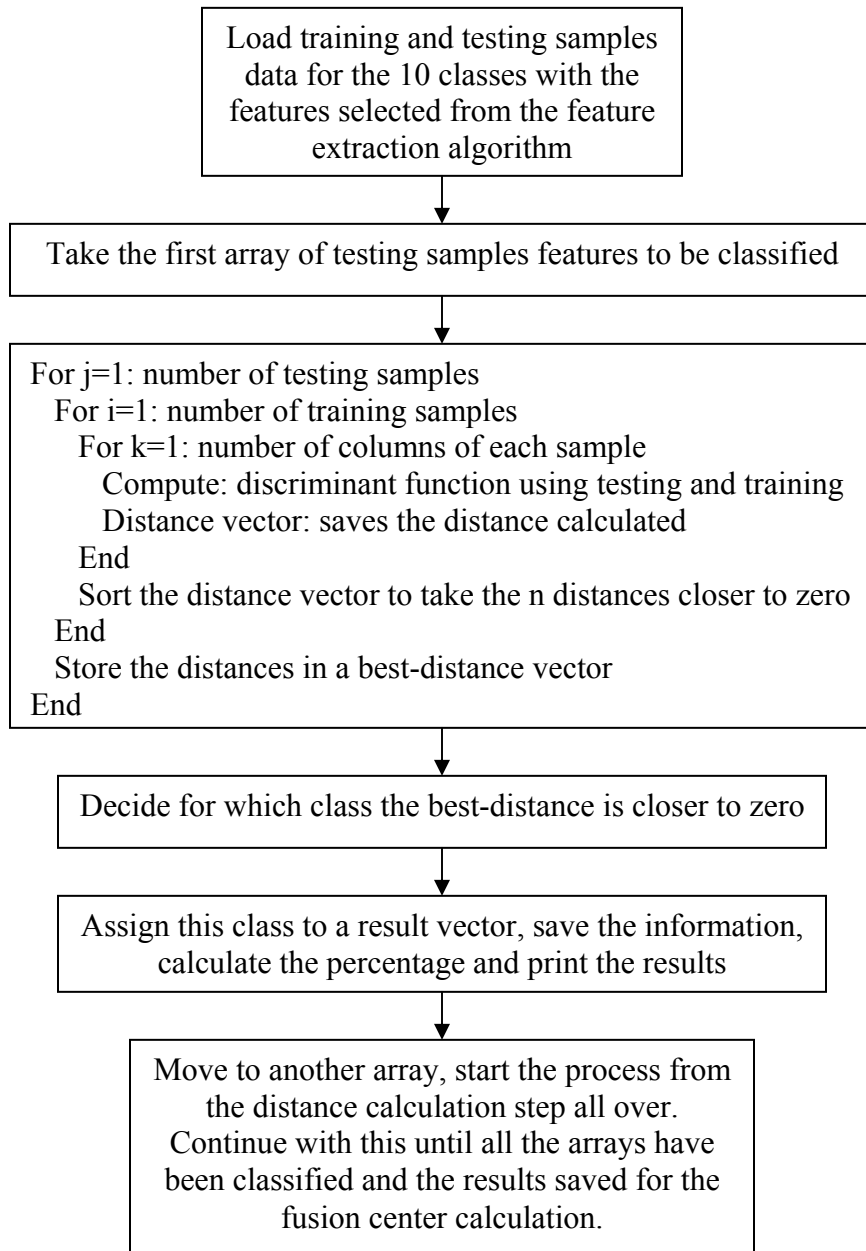


Figure 3.7 Classification algorithm

4 EXPERIMENTS AND RESULTS

4.1 Results from the Classifiers

We used 3 different classifiers, Maximum Likelihood, Euclidean Distance and K-Nearest Neighbor for $K=3$ and $K=5$. Also for the data filtering we used the command ‘dwt2’ (discrete wavelets transform 2D), that uses linear convolution. Because of the linear convolution, for some filters the quantity of samples differs. The results obtained from the classifiers used are presented as follows, using the filters described before. These results are obtained decomposing the first level approximation coefficients to second level decomposition using samples of 32 by 32. We made different combination and decomposing the sample to a second level using the other coefficients and combining them and the best results were obtained by decomposing to second level from the approximation of the first level and using all of the coefficients obtained of the second level.

4.1.1 Results using samples size of 32 by 32

Results for Daubechies filter ‘db1’:

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	282	179	49	51	63	62	1061	43	48	722	11.02
French	173	220	84	638	29	42	379	255	59	681	8.59
Liz	10	199	43	1477	201	109	7	55	42	417	1.68
Net	0	2	0	2550	2	0	0	2	1	3	99.61
Repskin	927	362	10	488	161	93	34	42	27	416	6.29
Scloth	190	122	55	1159	49	55	337	161	34	398	2.15
Straw	48	43	24	78	17	14	2218	17	8	93	86.64
Strcloth	12	34	76	1913	10	15	50	284	32	134	11.09
Water	78	186	193	351	78	62	25	227	114	1246	4.45
Weave	100	250	86	81	100	72	82	64	45	1680	65.63

TABLE 4.1 Euclidean Distance results for ‘db1’ using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	4	17	1	96	121	208	1531	182	114	286	0.16
French	6	44	0	494	75	215	481	804	109	332	1.72
Liz	0	2	0	1107	256	497	24	256	122	296	0.00
Net	0	0	0	2470	1	3	0	83	0	3	96.48
Repskin	1	19	0	422	164	679	752	180	81	262	6.41
Scloth	0	8	0	907	93	166	643	543	71	129	6.48
Straw	0	5	0	107	30	43	2141	128	41	65	83.63
Strcloth	0	0	0	1528	45	40	37	803	67	40	31.37
Water	0	14	0	355	269	165	150	766	402	439	15.70
Weave	6	72	2	141	178	284	412	337	286	842	32.89

TABLE 4.2 Maximum Likelihood results for ‘db1’ using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	1	0	2555	0	0	4	0.00
French	2	0	2	1411	3	0	0	0	0	1142	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2550	6	0	4	0	0	0	0	0	0	0.00
Scloth	8	0	0	2415	3	1	2	1	0	130	0.04
Straw	0	1	0	0	0	0	2556	0	0	3	99.84
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	1	0	0	27	0	0	0	60	0	2472	0.00
Weave	0	2	1	2	0	0	0	0	0	2555	99.80

TABLE 4.3 K-Nearest Neighbor K=3 results for 'db1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	2556	0	0	4	0.00
French	1	0	0	1245	2	0	0	0	0	1312	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2548	9	0	1	0	0	0	2	0	0	0.00
Scloth	6	0	0	2395	0	0	10	0	0	149	0.00
Straw	0	0	0	0	0	0	2557	0	0	3	99.88
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	0	0	0	27	0	0	1	59	0	2473	0.00
Weave	1	0	3	0	0	0	0	0	0	2556	99.84

TABLE 4.4 K-Nearest Neighbor K=5 results for 'db1' using samples of 32 by 32

Results for Haar filter ‘haar’:

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	282	179	49	51	63	62	1061	43	48	722	11.02
French	173	220	84	638	29	42	379	255	59	681	8.59
Liz	10	199	43	1477	201	109	7	55	42	417	1.68
Net	0	2	0	2550	2	0	0	2	1	3	99.61
Repskin	927	362	10	488	161	93	34	42	27	416	6.29
Scloth	190	122	55	1159	49	55	337	161	34	398	2.15
Straw	48	43	24	78	17	14	2218	17	8	93	86.64
Strcloth	12	34	76	1913	10	15	50	284	32	134	11.09
Water	78	186	193	351	78	62	25	227	114	1246	4.45
Weave	100	250	86	81	100	72	82	64	45	1680	65.63

TABLE 4.5 Euclidean Distance results for ‘haar’ using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	4	17	1	96	121	208	1531	182	114	286	0.16
French	6	44	0	494	75	215	481	804	109	332	1.72
Liz	0	2	0	1107	256	497	24	256	122	296	0.00
Net	0	0	0	2470	1	3	0	83	0	3	96.48
Repskin	1	19	0	422	164	679	752	180	81	262	6.41
Scloth	0	8	0	907	93	166	643	543	71	129	6.48
Straw	0	5	0	107	30	43	2141	128	41	65	83.63
Strcloth	0	0	0	1528	45	40	37	803	67	40	31.37
Water	0	14	0	355	269	165	150	766	402	439	15.70
Weave	6	72	2	141	178	284	412	337	286	842	32.89

TABLE 4.6 Maximum Likelihood results for ‘haar’ using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	1	0	2555	0	0	4	0.00
French	2	0	2	1411	3	0	0	0	0	1142	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2550	6	0	4	0	0	0	0	0	0	0.00
Scloth	8	0	0	2415	3	1	2	1	0	130	0.04
Straw	0	1	0	0	0	0	2556	0	0	3	99.84
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	1	0	0	27	0	0	0	60	0	2472	0.00
Weave	0	2	1	2	0	0	0	0	0	2555	99.80

TABLE 4.7 K-Nearest Neighbor K= 3 results for 'haar' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	2556	0	0	4	0.00
French	1	0	0	1245	2	0	0	0	0	1312	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2548	9	0	1	0	0	0	2	0	0	0.00
Scloth	6	0	0	2395	0	0	10	0	0	149	0.00
Straw	0	0	0	0	0	0	2557	0	0	3	99.88
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	0	0	0	27	0	0	1	59	0	2473	0.00
Weave	1	0	3	0	0	0	0	0	0	2556	99.84

TABLE 4.8 K-Nearest Neighbor K=5 results for 'haar' using samples of 32 by 32

Results for Coiflets filter 'coifl':

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	481	396	118	101	62	38	1967	128	134	1415	9.94
French	104	184	310	847	67	23	608	817	281	1599	3.80
Liz	10	455	72	2562	268	111	2	213	174	973	1.49
Net	0	1	1	4826	3	1	0	2	2	4	99.71
Repskin	1134	699	36	750	105	66	36	96	83	1835	2.17
Scloth	339	286	110	1932	121	26	642	341	125	918	0.54
Straw	122	107	30	78	22	14	4011	80	45	331	82.87
Strcloth	17	54	129	3829	16	3	112	443	86	151	9.15
Water	137	393	529	630	109	60	53	705	447	1777	9.24
Weave	301	481	192	182	114	58	556	212	194	2550	52.69

TABLE 4.9 Euclidean Distance results for 'coifl' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	8	36	1	107	244	241	2455	293	438	1017	0.17
French	30	110	20	634	215	161	706	1664	605	695	2.27
Liz	0	9	1	1850	1001	552	3	563	576	285	0.02
Net	0	0	0	4740	4	1	0	87	5	3	97.93
Repskin	2	36	0	518	522	473	450	295	333	2211	10.79
Scloth	4	15	0	1470	284	204	1135	852	375	501	4.21
Straw	1	16	2	117	100	98	3704	244	221	337	76.53
Strcloth	1	4	0	3112	72	60	56	1243	214	78	25.68
Water	2	48	9	413	380	377	213	1440	1267	691	26.18
Weave	22	123	14	161	416	331	1147	421	794	1411	29.15

TABLE 4.10 Maximum Likelihood results for 'coifl' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	4840	0	0	0	0.00
French	0	1	0	1512	0	1	1	259	0	3066	0.02
Liz	0	0	0	4840	0	0	0	0	0	0	0.00
Net	0	0	0	4840	0	0	0	0	0	0	100.00
Repskin	451	265	1	5	0	0	0	0	0	4118	0.00
Scloth	1	7	1	3191	21	0	540	0	4	1075	0.00
Straw	2	0	1	0	0	0	4837	0	0	0	99.94
Strcloth	0	0	0	4840	0	0	0	0	0	0	0.00
Water	0	0	0	19	0	0	0	857	0	3964	0.00
Weave	0	0	0	1	1	0	12	3	0	4823	99.65

TABLE 4.11 K-Nearest Neighbor K=3 results for 'coif1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	4840	0	0	0	0.00
French	0	0	0	1493	0	0	2	223	0	3122	0.00
Liz	0	0	0	4840	0	0	0	0	0	0	0.00
Net	0	0	0	4840	0	0	0	0	0	0	100.00
Repskin	691	290	0	5	0	0	0	0	0	3854	0.00
Scloth	0	5	0	3750	9	0	442	12	0	622	0.00
Straw	3	0	0	0	0	0	4837	0	0	0	99.94
Strcloth	0	0	0	4840	0	0	0	0	0	0	0.00
Water	1	0	0	19	0	0	0	1050	0	3770	0.00
Weave	3	1	1	0	0	0	8	3	0	4824	99.67

TABLE 4.12 K-Nearest Neighbor K=5 results for 'coif1' using samples of 32 by 32

Results for Symlets filter 'sym2':

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	319	455	62	56	77	49	1679	84	99	1120	7.98
French	120	406	211	620	76	42	311	726	260	1228	10.15
Liz	13	500	101	1773	231	144	2	275	201	760	2.53
Net	0	5	3	3973	3	1	0	7	5	3	99.33
Repskin	981	783	43	493	99	104	28	102	92	1275	2.48
Scloth	308	462	103	1248	87	67	425	470	110	720	1.68
Straw	100	162	29	52	41	10	3291	43	32	240	82.28
Strcloth	14	68	86	3075	12	19	98	421	53	154	10.53
Water	91	428	298	492	133	106	49	631	371	1401	9.28
Weave	200	761	135	153	138	77	281	232	200	1823	45.58

TABLE 4.13 Euclidean Distance results for 'sym2' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	11	12	0	87	253	240	2226	156	275	740	0.28
French	12	49	0	580	265	183	530	1198	533	650	1.23
Liz	0	1	0	1152	758	391	7	777	330	584	0.00
Net	0	0	0	1271	3	2	0	2715	4	5	31.78
Repskin	0	8	1	424	327	619	503	200	258	1660	8.18
Scloth	2	5	0	1025	238	208	848	838	306	530	5.20
Straw	4	6	1	128	147	118	2943	191	182	280	73.58
Strcloth	0	1	0	2119	70	61	51	1479	135	84	36.98
Water	1	14	2	483	459	350	192	1020	848	631	21.20
Weave	18	75	4	186	393	339	782	475	529	1199	29.98

TABLE 4.14 Maximum Likelihood results for 'sym2' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	1	0	0	3998	0	0	1	0.00
French	0	4	1	1370	1	0	0	8	0	2616	0.10
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	1519	1	0	5	2	3	0	0	0	2470	0.05
Scloth	2	297	2	2266	0	0	69	4	0	1360	0.00
Straw	4	0	0	0	0	0	3996	0	0	0	99.90
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	1	0	0	65	0	0	0	831	0	3103	0.00
Weave	0	1	1	3	1	0	58	0	0	3936	98.40

TABLE 4.15 K-Nearest Neighbor K=3 results for 'sym2' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	4000	0	0	0	0.00
French	1	3	0	1375	1	0	0	20	0	2600	0.08
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	3987	0	0	5	3	4	0	0	0	1	0.08
Scloth	3	479	1	2489	0	1	80	5	2	940	0.03
Straw	4	0	0	0	0	0	3996	0	0	0	99.90
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	1	1	0	71	0	0	1	846	0	3080	0.00
Weave	0	260	1	2	0	1	55	1	0	3680	92.00

TABLE 4.16 K-Nearest Neighbor K=5 results for 'sym2' using samples of 32 by 32

Results for Biorthogonal filter 'bior1.1':

Euclidean Distance Results db1				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	282	179	49	51	63	62	1061	43	48	722	11.02
French	173	220	84	638	29	42	379	255	59	681	8.59
Liz	10	199	43	1477	201	109	7	55	42	417	1.68
Net	0	2	0	2550	2	0	0	2	1	3	99.61
Repskin	927	362	10	488	161	93	34	42	27	416	6.29
Scloth	190	122	55	1159	49	55	337	161	34	398	2.15
Straw	48	43	24	78	17	14	2218	17	8	93	86.64
Strcloth	12	34	76	1913	10	15	50	284	32	134	11.09
Water	78	186	193	351	78	62	25	227	114	1246	4.45
Weave	100	250	86	81	100	72	82	64	45	1680	65.63

TABLE 4.17 Euclidean Distance results for 'bior1.1' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	4	17	1	96	121	208	1531	182	114	286	0.16
French	6	44	0	494	75	215	481	804	109	332	1.72
Liz	0	2	0	1107	256	497	24	256	122	296	0.00
Net	0	0	0	2470	1	3	0	83	0	3	96.48
Repskin	1	19	0	422	164	679	752	180	81	262	6.41
Scloth	0	8	0	907	93	166	643	543	71	129	6.48
Straw	0	5	0	107	30	43	2141	128	41	65	83.63
Strcloth	0	0	0	1528	45	40	37	803	67	40	31.37
Water	0	14	0	355	269	165	150	766	402	439	15.70
Weave	6	72	2	141	178	284	412	337	286	842	32.89

TABLE 4.18 Maximum Likelihood results for 'bior1.1' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	1	0	2555	0	0	4	0.00
French	2	0	2	1411	3	0	0	0	0	1142	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2550	6	0	4	0	0	0	0	0	0	0.00
Scloth	8	0	0	2415	3	1	2	1	0	130	0.04
Straw	0	1	0	0	0	0	2556	0	0	3	99.84
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	1	0	0	27	0	0	0	60	0	2472	0.00
Weave	0	2	1	2	0	0	0	0	0	2555	99.80

TABLE 4.19 K-Nearest Neighbor K=3 results for 'bior1.1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	2556	0	0	4	0.00
French	1	0	0	1245	2	0	0	0	0	1312	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2548	9	0	1	0	0	0	2	0	0	0.00
Scloth	6	0	0	2395	0	0	10	0	0	149	0.00
Straw	0	0	0	0	0	0	2557	0	0	3	99.88
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	0	0	0	27	0	0	1	59	0	2473	0.00
Weave	1	0	3	0	0	0	0	0	0	2556	99.84

TABLE 4.20 K-Nearest Neighbor K=5 results for 'bior1.1' using samples of 32 by 32

Results for Reverse Biorthogonal filter 'rbio1.1':

Euclidean Distance Results db1				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	282	179	49	51	63	62	1061	43	48	722	11.02
French	173	220	84	638	29	42	379	255	59	681	8.59
Liz	10	199	43	1477	201	109	7	55	42	417	1.68
Net	0	2	0	2550	2	0	0	2	1	3	99.61
Repskin	927	362	10	488	161	93	34	42	27	416	6.29
Scloth	190	122	55	1159	49	55	337	161	34	398	2.15
Straw	48	43	24	78	17	14	2218	17	8	93	86.64
Strcloth	12	34	76	1913	10	15	50	284	32	134	11.09
Water	78	186	193	351	78	62	25	227	114	1246	4.45
Weave	100	250	86	81	100	72	82	64	45	1680	65.63

TABLE 4.21 Euclidean Distance results for 'rbio1.1' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	4	17	1	96	121	208	1531	182	114	286	0.16
French	6	44	0	494	75	215	481	804	109	332	1.72
Liz	0	2	0	1107	256	497	24	256	122	296	0.00
Net	0	0	0	2470	1	3	0	83	0	3	96.48
Repskin	1	19	0	422	164	679	752	180	81	262	6.41
Scloth	0	8	0	907	93	166	643	543	71	129	6.48
Straw	0	5	0	107	30	43	2141	128	41	65	83.63
Strcloth	0	0	0	1528	45	40	37	803	67	40	31.37
Water	0	14	0	355	269	165	150	766	402	439	15.70
Weave	6	72	2	141	178	284	412	337	286	842	32.89

TABLE 4.22 Maximum Likelihood results for 'rbio1.1' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	1	0	2555	0	0	4	0.00
French	2	0	2	1411	3	0	0	0	0	1142	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2550	6	0	4	0	0	0	0	0	0	0.00
Scloth	8	0	0	2415	3	1	2	1	0	130	0.04
Straw	0	1	0	0	0	0	2556	0	0	3	99.84
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	1	0	0	27	0	0	0	60	0	2472	0.00
Weave	0	2	1	2	0	0	0	0	0	2555	99.80

TABLE 4.23 K-Nearest Neighbor K=3 results for 'rbio1.1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	2556	0	0	4	0.00
French	1	0	0	1245	2	0	0	0	0	1312	0.00
Liz	0	0	0	2560	0	0	0	0	0	0	0.00
Net	0	0	0	2560	0	0	0	0	0	0	100.00
Repskin	2548	9	0	1	0	0	0	2	0	0	0.00
Scloth	6	0	0	2395	0	0	10	0	0	149	0.00
Straw	0	0	0	0	0	0	2557	0	0	3	99.88
Strcloth	0	0	0	2560	0	0	0	0	0	0	0.00
Water	0	0	0	27	0	0	1	59	0	2473	0.00
Weave	1	0	3	0	0	0	0	0	0	2556	99.84

TABLE 4.24 K-Nearest Neighbor K=5 results for 'rbio1.1' using samples of 32 by 32

Results for Daubechies filter 'db2':

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	319	455	62	56	77	49	1679	84	99	1120	7.98
French	120	406	211	620	76	42	311	726	260	1228	10.15
Liz	13	500	101	1773	231	144	2	275	201	760	2.53
Net	0	5	3	3973	3	1	0	7	5	3	99.33
Repskin	981	783	43	493	99	104	28	102	92	1275	2.48
Scloth	308	462	103	1248	87	67	425	470	110	720	1.68
Straw	100	162	29	52	41	10	3291	43	32	240	82.28
Strcloth	14	68	86	3075	12	19	98	421	53	154	10.53
Water	91	428	298	492	133	106	49	631	371	1401	9.28
Weave	200	761	135	153	138	77	281	232	200	1823	45.58

TABLE 4.25 Euclidean Distance results for 'db2' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	11	12	0	87	253	240	2226	156	275	740	0.28
French	12	49	0	580	265	183	530	1198	533	650	1.23
Liz	0	1	0	1152	758	391	7	777	330	584	0.00
Net	0	0	0	1271	3	2	0	2715	4	5	31.78
Repskin	0	8	1	424	327	619	503	200	258	1660	8.18
Scloth	2	5	0	1025	238	208	848	838	306	530	5.20
Straw	4	6	1	128	147	118	2943	191	182	280	73.58
Strcloth	0	1	0	2119	70	61	51	1479	135	84	36.98
Water	1	14	2	483	459	350	192	1020	848	631	21.20
Weave	18	75	4	186	393	339	782	475	529	1199	29.98

TABLE 4.26 Maximum Likelihood results for 'db2' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	1	0	0	0	3998	0	0	1	0.00
French	0	4	1	1370	1	0	0	8	0	2616	0.10
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	1519	0	0	5	2	3	0	0	0	2471	0.05
Scloth	3	299	1	2266	1	0	69	4	0	1357	0.00
Straw	3	0	0	0	0	0	3997	0	0	0	99.93
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	1	0	0	66	0	0	0	830	0	3103	0.00
Weave	2	1	0	3	0	0	58	0	0	3936	98.40

TABLE 4.27 K-Nearest Neighbor K=3 results for 'db2' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	0	0	0	4000	0	0	0	0.00
French	0	3	1	1375	1	0	0	21	0	2599	0.08
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	3987	0	0	5	2	5	0	0	0	1	0.05
Scloth	4	479	0	2489	0	1	79	6	2	940	0.03
Straw	4	0	0	0	0	0	3996	0	0	0	99.90
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	3	0	0	71	0	0	0	846	0	3080	0.00
Weave	0	261	1	2	0	1	57	1	0	3677	91.93

TABLE 4.28 K-Nearest Neighbor K=5 results for 'db2' using samples of 32 by 32

Results for Biorthogonal filter ‘bior3.1’:

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	240	543	121	200	88	99	1036	63	97	1513	6.00
French	140	585	270	376	230	141	573	148	247	1290	14.63
Liz	54	329	1014	888	411	150	80	411	175	488	25.35
Net	43	3	433	3015	14	2	81	403	1	5	75.38
Repskin	220	664	263	576	192	308	300	102	416	959	4.80
Scloth	273	555	542	455	204	202	567	148	198	856	5.05
Straw	424	307	66	146	41	72	1959	23	42	920	48.98
Strcloth	381	446	460	601	503	202	388	259	341	419	6.48
Water	169	618	334	440	234	130	366	124	226	1359	5.65
Weave	176	443	159	396	86	108	742	82	141	1667	41.68

TABLE 4.29 Euclidean Distance results for ‘bior3.1’ using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	26	106	39	258	245	518	1342	250	318	898	0.65
French	15	163	77	497	609	276	660	519	482	702	4.08
Liz	0	65	70	1264	996	106	141	769	323	266	1.75
Net	0	1	0	3460	391	24	92	20	3	9	86.50
Repskin	0	53	23	575	499	340	696	473	822	519	12.48
Scloth	3	91	43	662	540	383	817	487	457	517	9.58
Straw	11	50	14	177	107	896	1643	119	167	816	41.08
Strcloth	2	32	26	874	764	275	612	696	407	312	17.40
Water	4	166	100	572	526	255	652	471	498	756	12.45
Weave	59	240	122	510	340	263	976	268	293	929	23.23

TABLE 4.30 Maximum Likelihood results for ‘bior3.1’ using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	0	0	101	0	0	1722	1	0	2176	0.00
French	0	2	1	251	0	0	701	0	0	3045	0.05
Liz	1	0	2532	1467	0	0	0	0	0	0	63.30
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	2	895	1	524	0	131	8	0	1	2438	0.00
Scloth	1	19	0	954	0	0	1099	0	0	1927	0.00
Straw	2	0	0	5	0	0	3993	0	0	0	99.83
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	0	684	0	528	0	0	0	0	0	2788	0.00
Weave	0	0	0	278	0	0	940	0	0	2782	69.55

TABLE 4.31 K-Nearest Neighbor K=3 results for 'bior3.1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	0	1	0	93	0	0	3682	0	0	224	0.00
French	0	0	1	232	0	0	1037	0	0	2730	0.00
Liz	0	0	3371	629	0	0	0	0	0	0	84.28
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	1	939	0	564	0	106	24	0	0	2366	0.00
Scloth	0	2	0	809	0	0	2770	0	0	419	0.00
Straw	0	0	0	5	0	0	3995	0	0	0	99.88
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	0	666	0	546	0	0	0	0	0	2788	0.00
Weave	0	0	0	245	0	0	1304	0	0	2451	61.28

TABLE 4.32 K-Nearest Neighbor K=5 results for 'bior3.1' using samples of 32 by 32

Results for Reverse Biorthogonal filter 'rbio3.1':

Euclidean Distance Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	499	342	189	64	165	149	1971	144	61	416	12.48
French	293	361	438	543	177	69	560	772	108	679	9.03
Liz	130	603	173	1605	319	531	9	192	85	353	4.33
Net	2	6	1	3977	0	2	1	7	0	4	99.43
Repskin	1779	444	131	509	260	334	157	76	52	258	6.50
Scloth	294	167	178	1528	189	131	809	466	37	201	3.28
Straw	154	51	53	104	67	46	3367	66	41	51	84.18
Strcloth	16	14	150	3138	39	12	82	472	19	58	11.80
Water	120	355	934	536	253	93	34	840	205	630	5.13
Weave	270	703	686	103	487	245	142	225	179	960	24.00

TABLE 4.33 Euclidean Distance results for 'rbio3.1' using samples of 32 by 32

Maximum Likelihood Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	8	36	7	138	253	518	2285	249	248	258	0.20
French	33	90	0	577	218	359	787	1148	439	349	2.25
Liz	0	4	0	1439	397	1053	7	258	274	568	0.00
Net	0	0	0	3953	1	7	0	33	2	4	98.83
Repskin	0	8	0	468	290	1331	911	203	242	547	7.25
Scloth	0	8	2	1305	165	344	1076	744	190	166	8.60
Straw	6	5	0	154	102	192	3145	179	134	83	78.63
Strcloth	0	0	0	2802	35	51	61	859	125	67	21.48
Water	2	8	0	553	403	293	123	1286	890	442	22.25
Weave	17	105	3	167	748	831	323	432	722	652	16.30

TABLE 4.34 Maximum Likelihood results for 'rbio3.1' using samples of 32 by 32

KNN K=3 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	2	2	0	32	0	0	3954	9	0	1	0.05
French	7	7	0	1188	4	2	1	2788	0	3	0.18
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	3947	0	0	52	1	0	0	0	0	0	0.03
Scloth	0	2	0	3523	7	0	468	0	0	0	0.00
Straw	0	0	0	0	6	0	3994	0	0	0	99.85
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	0	0	3771	228	0	1	0	0	0	0	0.00
Weave	11	828	642	70	126	0	1	0	0	2322	58.05

TABLE 4.35 K-Nearest Neighbor K=3 results for 'rbio3.1' using samples of 32 by 32

KNN K=5 Results				All Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	3	1	1	28	0	2	3957	8	0	0	0.08
French	0	1	0	1390	1	0	0	1868	0	740	0.03
Liz	0	0	0	4000	0	0	0	0	0	0	0.00
Net	0	0	0	4000	0	0	0	0	0	0	100.00
Repskin	3951	0	0	48	1	0	0	0	0	0	0.03
Scloth	0	1	0	3951	3	0	45	0	0	0	0.00
Straw	0	0	0	0	5	0	3995	0	0	0	99.88
Strcloth	0	0	0	4000	0	0	0	0	0	0	0.00
Water	0	0	3736	264	0	0	0	0	0	0	0.00
Weave	0	2788	678	73	92	0	0	0	0	369	9.23

TABLE 4.36 K-Nearest Neighbor K=5 results for 'rbio3.1' using samples of 32 by 32

4.2 Results from the Fusion Methods

The results obtained from the classifiers were saved for each sample. After all of the classifiers finished with the samples for each filter, we entered the data in a fusion center using majority voting. We will present the results as fusion results for samples of 32 by 32. In each of them we will present tables for each filter used.

4.2.1 Results using Fusion Method for samples size of 32 by 32

Fusion Results 'db1' for 32 by 32				All Coefficients										
Data	Classified as													
	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.			
Cork	149	145	3	0	0	0	2259	0	0	4	5.82			
French	378	355	11	968	2	1	7	19	2	817	13.87			
Liz	126	130	0	2304	0	0	0	0	0	0	0.00			
Net	0	3	0	2557	0	0	0	0	0	0	99.88			
Repskin	2271	3	2	168	12	21	16	6	3	58	0.47			
Scloth	261	221	2	1967	2	2	7	0	0	98	0.08			
Straw	45	56	0	0	0	0	2457	1	0	1	95.98			
Strcloth	112	106	0	2342	0	0	0	0	0	0	0.00			
Water	210	204	3	26	0	0	0	30	0	2087	0.00			
Weave	119	102	12	2	0	0	0	1	0	2324	90.78			

TABLE 4.37 Majority Voting results for 'db1' using samples of 32 by 32

Fusion Results 'haar' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	149	145	3	0	0	0	2259	0	0	4	5.82	
French	378	355	11	968	2	1	7	19	2	817	13.87	
Liz	126	130	0	2304	0	0	0	0	0	0	0.00	
Net	0	3	0	2557	0	0	0	0	0	0	99.88	
Repskin	2271	3	2	168	12	21	16	6	3	58	0.47	
Scloth	261	221	2	1967	2	2	7	0	0	98	0.08	
Straw	45	56	0	0	0	0	2457	1	0	1	95.98	
Strcloth	112	106	0	2342	0	0	0	0	0	0	0.00	
Water	210	204	3	26	0	0	0	30	0	2087	0.00	
Weave	119	102	12	2	0	0	0	1	0	2324	90.78	

TABLE 4.38 Majority Voting results for 'haar' using samples of 32 by 32

Fusion Results 'coif1' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	392	416	3	0	0	0	4029	0	0	0	8.10	
French	791	792	9	930	0	0	12	149	2	2155	16.36	
Liz	262	214	0	4364	0	0	0	0	0	0	0.00	
Net	1	2	0	4837	0	0	0	0	0	0	99.94	
Repskin	620	294	56	106	4	1	2	4	1	3752	0.08	
Scloth	589	612	22	2740	13	0	362	30	7	465	0.00	
Straw	148	120	1	0	0	0	4571	0	0	0	94.44	
Strcloth	178	212	1	4449	0	0	0	0	0	0	0.00	
Water	544	532	10	29	0	1	0	687	11	3026	0.23	
Weave	413	402	9	4	0	0	5	2	0	4005	82.75	

TABLE 4.39 Majority Voting results for 'coif1' using samples of 32 by 32

Fusion Results 'sym2' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	271	277	0	0	0	0	3452	0	0	0	6.78	
French	593	599	11	939	0	0	2	10	1	1845	14.98	
Liz	293	285	0	3422	0	0	0	0	0	0	0.00	
Net	5	4	0	3991	0	0	0	0	0	0	99.78	
Repskin	1983	138	12	207	45	37	9	36	32	1501	1.13	
Scloth	537	636	64	1915	2	4	90	31	6	715	0.10	
Straw	95	83	0	0	0	0	3822	0	0	0	95.55	
Strcloth	184	166	0	3650	0	0	0	0	0	0	0.00	
Water	430	432	3	63	1	0	0	611	1	2459	0.03	
Weave	311	298	41	12	5	1	41	16	7	3268	81.70	

TABLE 4.40 Majority Voting results for 'sym2' using samples of 32 by 32

Fusion Results 'bior1.1' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	149	145	3	0	0	0	2259	0	0	4	5.82	
French	378	355	11	968	2	1	7	19	2	817	13.87	
Liz	126	130	0	2304	0	0	0	0	0	0	0.00	
Net	0	3	0	2557	0	0	0	0	0	0	99.88	
Repskin	2271	3	2	168	12	21	16	6	3	58	0.47	
Scloth	261	221	2	1967	2	2	7	0	0	98	0.08	
Straw	45	56	0	0	0	0	2457	1	0	1	95.98	
Strcloth	112	106	0	2342	0	0	0	0	0	0	0.00	
Water	210	204	3	26	0	0	0	30	0	2087	0.00	
Weave	119	102	12	2	0	0	0	1	0	2324	90.78	

TABLE 4.41 Majority Voting results for 'bior1.1' using samples of 32 by 32

Fusion Results 'rbio1.1' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	149	145	3	0	0	0	2259	0	0	4	5.82	
French	378	355	11	968	2	1	7	19	2	817	13.87	
Liz	126	130	0	2304	0	0	0	0	0	0	0.00	
Net	0	3	0	2557	0	0	0	0	0	0	99.88	
Repskin	2271	3	2	168	12	21	16	6	3	58	0.47	
Scloth	261	221	2	1967	2	2	7	0	0	98	0.08	
Straw	45	56	0	0	0	0	2457	1	0	1	95.98	
Strcloth	112	106	0	2342	0	0	0	0	0	0	0.00	
Water	210	204	3	26	0	0	0	30	0	2087	0.00	
Weave	119	102	12	2	0	0	0	1	0	2324	90.78	

TABLE 4.42 Majority Voting results for 'rbio1.1' using samples of 32 by 32

Fusion Results 'db2' for 32 by 32					All Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	293	254	0	0	0	0	3453	0	0	0	7.33	
French	610	587	7	939	0	0	2	10	1	1844	14.68	
Liz	302	276	0	3422	0	0	0	0	0	0	0.00	
Net	6	3	0	3991	0	0	0	0	0	0	99.78	
Repskin	2001	140	9	202	49	35	7	39	34	1484	1.23	
Scloth	488	678	71	1917	2	4	90	31	6	713	0.10	
Straw	76	101	0	0	0	0	3823	0	0	0	95.58	
Strcloth	175	175	0	3650	0	0	0	0	0	0	0.00	
Water	464	397	5	63	1	0	0	611	1	2458	0.03	
Weave	318	288	43	13	5	1	41	14	7	3270	81.75	

TABLE 4.43 Majority Voting results for 'db2' using samples of 32 by 32

Fusion Results 'bior31' for 32 by 32					All Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	533	400	77	185	16	22	1938	6	12	811	13.33
French	454	438	37	201	16	1	591	4	8	2250	10.95
Liz	624	578	1874	750	81	8	1	29	9	46	46.85
Net	47	45	0	3908	0	0	0	0	0	0	97.70
Repskin	424	897	131	466	5	49	16	3	5	2004	0.13
Scloth	584	459	69	889	23	15	1276	19	9	657	0.38
Straw	225	215	8	3	0	0	3548	0	0	1	88.70
Strcloth	333	319	4	3344	0	0	0	0	0	0	0.00
Water	464	705	136	430	1	1	2	0	0	2261	0.00
Weave	509	451	37	223	0	3	762	5	3	2007	50.18

TABLE 4.44 Majority Voting results for 'bior3.1' using samples of 32 by 32

Fusion Results 'rbio31' for 32 by 32					All Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	160	184	5	19	0	0	3630	1	0	1	4.00
French	577	495	38	914	7	12	110	1712	12	123	12.38
Liz	268	277	2	3453	0	0	0	0	0	0	0.05
Net	2	1	0	3997	0	0	0	0	0	0	99.93
Repskin	3545	0	0	211	27	95	64	16	7	35	0.68
Scloth	516	513	9	2771	10	9	116	51	1	4	0.23
Straw	73	119	2	0	4	0	3802	0	0	0	95.05
Strcloth	179	158	0	3663	0	0	0	0	0	0	0.00
Water	537	550	2689	210	3	0	0	9	1	1	0.03
Weave	617	1136	734	217	168	57	37	52	39	943	23.58

TABLE 4.45 Majority Voting results for 'rbio3.1' using samples of 32 by 32

4.3 Other Experiments Made

We made a lot of different experiments trying to find a better result or another way to get a better classification result. The ones that we are going to present are the ones that we think are significant.

4.3.1 Variance as a measure of activity

We used the diagonal coefficients from the second level decomposition which are the High-High components and we calculated the variances of each sample and we then classified them. Then we made combinations using the other coefficients trying to find a good combination. Some of the results are presented below.

db1 variance results high decomp					High-High Diagonal Coefficients								
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	17	6	0	0	0	0	13	0	4	0	42.50		
French	23	17	0	0	0	0	0	0	0	0	42.50		
Liz	0	23	17	0	0	0	0	0	0	0	42.50		
Net	0	0	23	17	0	0	0	0	0	0	42.50		
Repskin	0	0	0	23	17	0	0	0	0	0	42.50		
Scloth	0	0	0	0	23	17	0	0	0	0	42.50		
Straw	0	0	0	0	0	23	17	0	0	0	42.50		
Strcloth	0	0	0	0	0	0	23	17	0	0	42.50		
Water	0	0	0	0	0	0	0	23	17	0	42.50		
Weave	0	0	0	0	0	0	0	0	23	17	42.50		

TABLE 4.46 Variance as a measure of activity results for ‘db1’

sym2 variance results high decomp					High-High Diagonal Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	17	0	23	0	0	0	0	0	0	0	42.50
French	23	17	0	0	0	0	0	0	0	0	42.50
Liz	0	23	17	0	0	0	0	0	0	0	42.50
Net	0	0	23	17	0	0	0	0	0	0	42.50
Repskin	0	0	0	23	17	0	0	0	0	0	42.50
Scloth	0	0	0	0	23	17	0	0	0	0	42.50
Straw	0	0	0	0	0	23	17	0	0	0	42.50
Strcloth	0	0	0	0	0	0	23	17	0	0	42.50
Water	0	0	0	0	0	0	0	23	17	0	42.50
Weave	0	0	0	0	0	0	0	0	23	17	42.50

TABLE 4.47 Majority Variance as a measure of activity results for 'sym2'

rbio31 variance results high decomp					High-High Diagonal Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	17	0	0	0	0	0	0	23	0	0	42.50
French	23	17	0	0	0	0	0	0	0	0	42.50
Liz	0	23	17	0	0	0	0	0	0	0	42.50
Net	0	0	23	17	0	0	0	0	0	0	42.50
Repskin	0	0	0	23	17	0	0	0	0	0	42.50
Scloth	0	0	0	0	23	17	0	0	0	0	42.50
Straw	0	0	0	0	0	23	17	0	0	0	42.50
Strcloth	0	0	0	0	0	0	23	17	0	0	42.50
Water	0	0	0	0	0	0	0	23	17	0	42.50
Weave	0	0	0	0	0	0	0	0	23	17	42.50

TABLE 4.48 Majority Variance as a measure of activity results for 'rbio31'

4.3.2 Classification of samples taken from Mayagüez Bay

We tested our algorithms with samples taken from an image of Mayagüez Bay. We selected three different classes from the image, vegetation, urban and water. We used all of the coefficients obtained by the wavelet decomposition at second level. We will present the results of the classification of those samples.

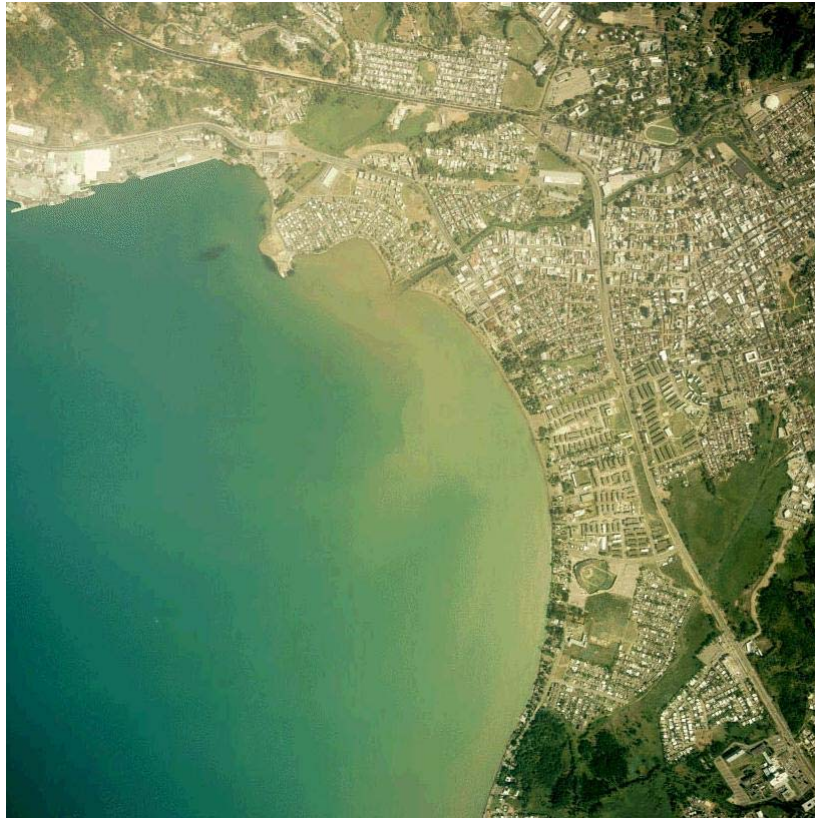


Figure 4.1 Mayagüez Bay image

EU Results db1				
	Classified as			
Data	veggie	urban	water	% Classif.
veggie	259	180	73	50.585938
urban	150	234	128	45.703125
water	8	28	476	92.96875

TABLE 4.49 Euclidean Distance results for 'db1' samples from Mayagüez Bay

ML Results db1				
	Classified as			
Data	veggie	urban	water	% Classif.
veggie	309	100	103	60.351563
urban	190	179	143	34.960938
water	4	5	503	98.242188

TABLE 4.50 Maximum Likelihood results for 'db1' samples from Mayagüez Bay

KNN K=3 Results db1				
	Classified as			
Data	veggie	urban	water	% Classif.
veggie	0	512	0	0
urban	26	486	0	94.921875
water	0	0	512	100

TABLE 4.51 K-Nearest Neighbor K=3 results for 'db1' samples from Mayagüez Bay

KNN K=5 Results db1				
	Classified as			
Data	veggie	urban	water	% Classif.
veggie	0	512	0	0
urban	4	507	1	99.023438
water	0	0	512	100

TABLE 4.52 K-Nearest Neighbor K=5 results for 'db1' samples from Mayagüez Bay

Fusion Results db1				
	Classified as			
Data	veggie	urban	water	% Classif.
veggie	111	382	19	21.679688
urban	72	397	43	77.539063
water	0	2	510	99.609375

TABLE 4.53 Majority Voting results for ‘db1’ samples from Mayagüez Bay

4.3.3 Results of Fusion Methods using samples size of 32 by 32 with approximation coefficients

We will present some of the results in a graphic that has the filters and the correct percentage by class. The results presented are of the Fusion Methods Results using only the approximation coefficients.

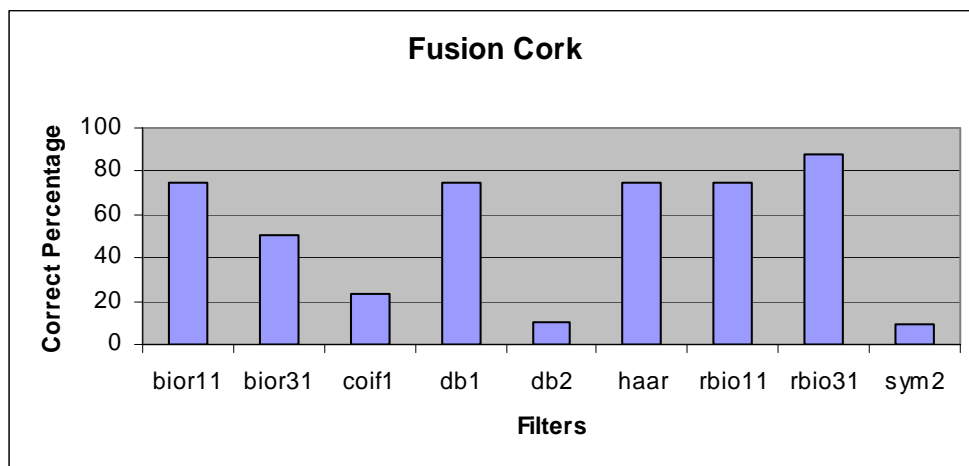


Figure 4.2 Majority Voting Cork image

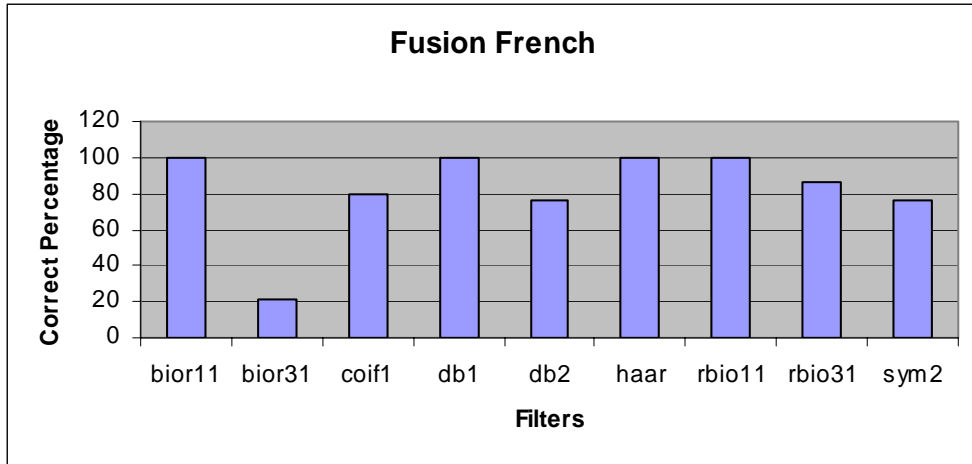


Figure 4.3 Majority Voting French image

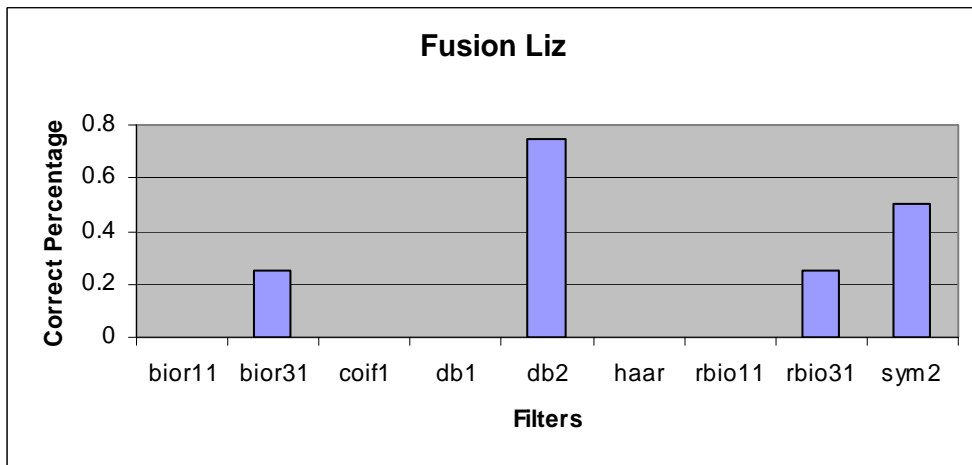


Figure 4.4 Majority Voting Liz image

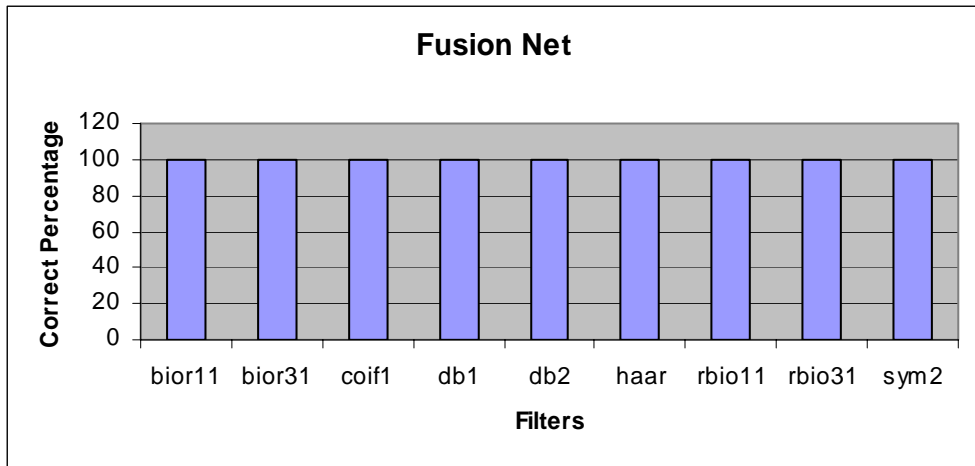


Figure 4.5 Majority Voting Net image

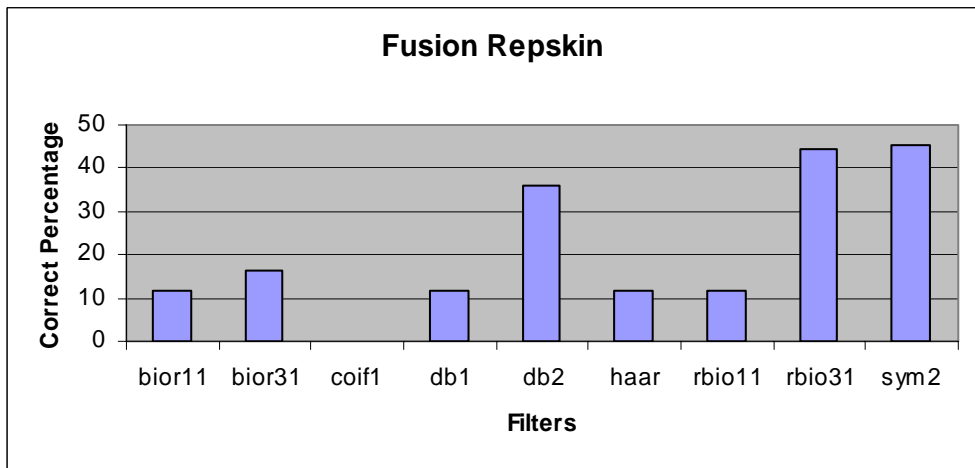


Figure 4.6 Majority Voting Repskin image

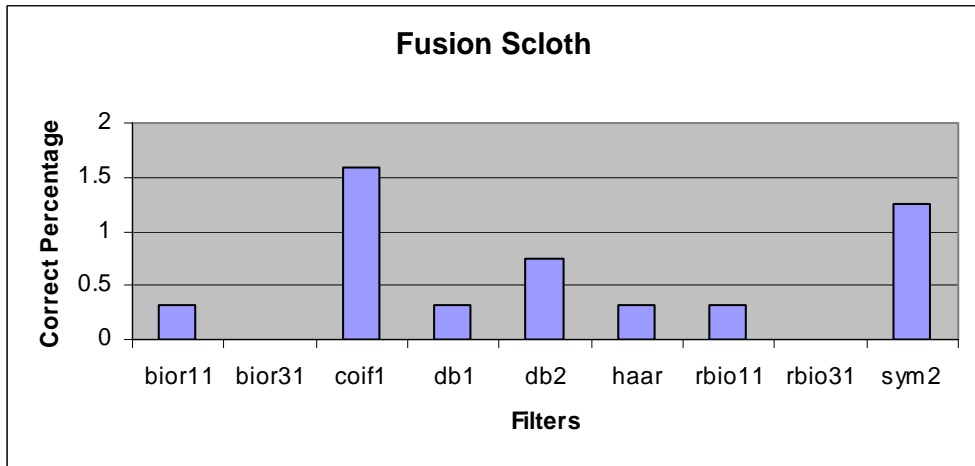


Figure 4.7 Majority Voting Scloth image

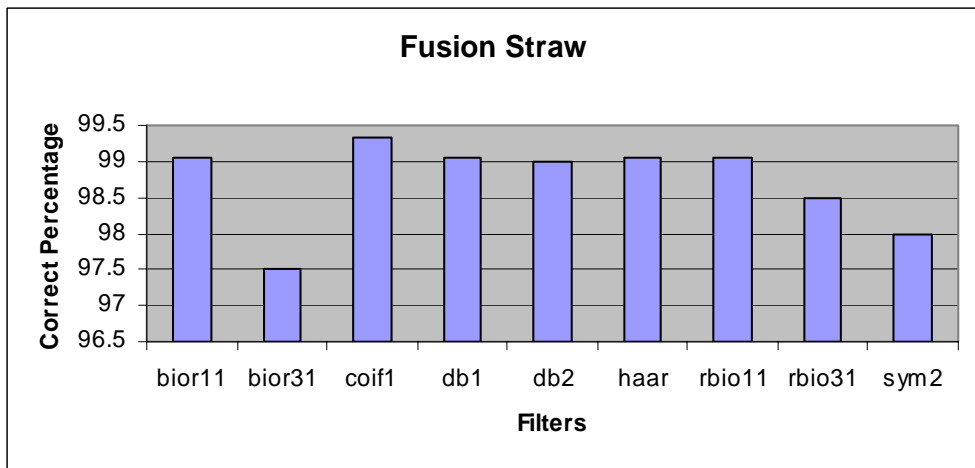


Figure 4.8 Majority Voting Straw image

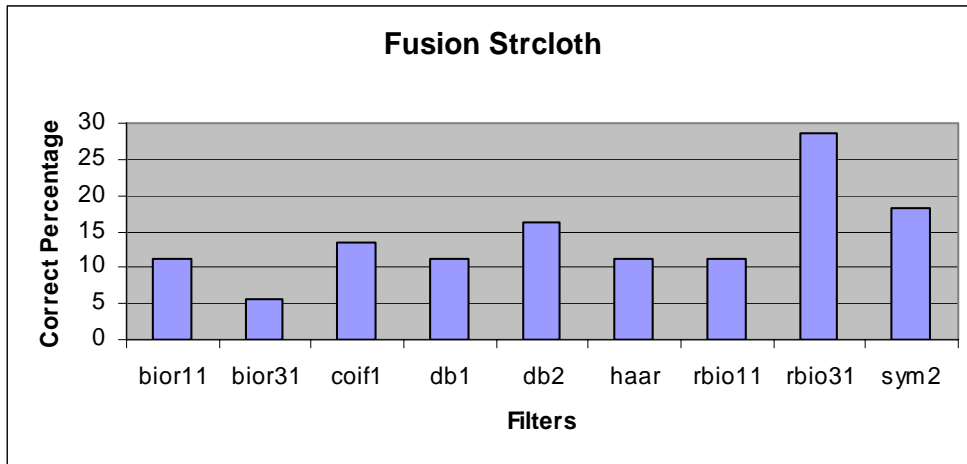


Figure 4.9 Majority Voting Strcloth image

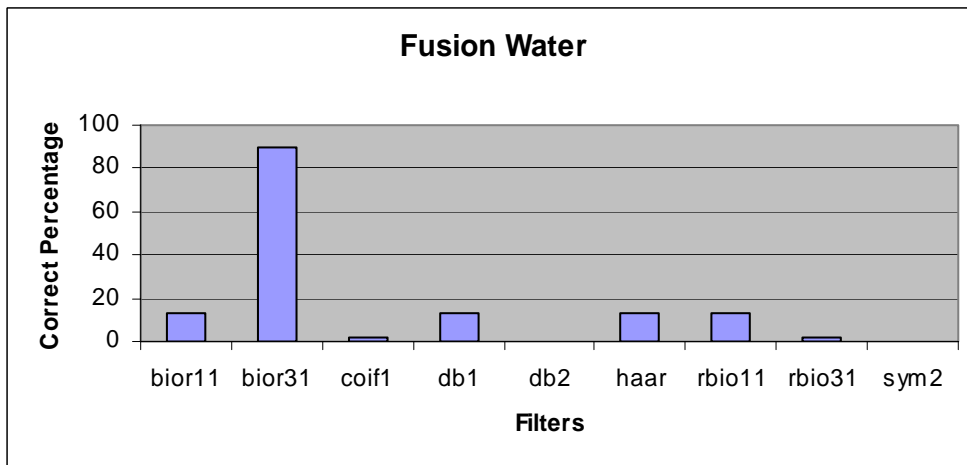


Figure 4.10 Majority Voting Water image

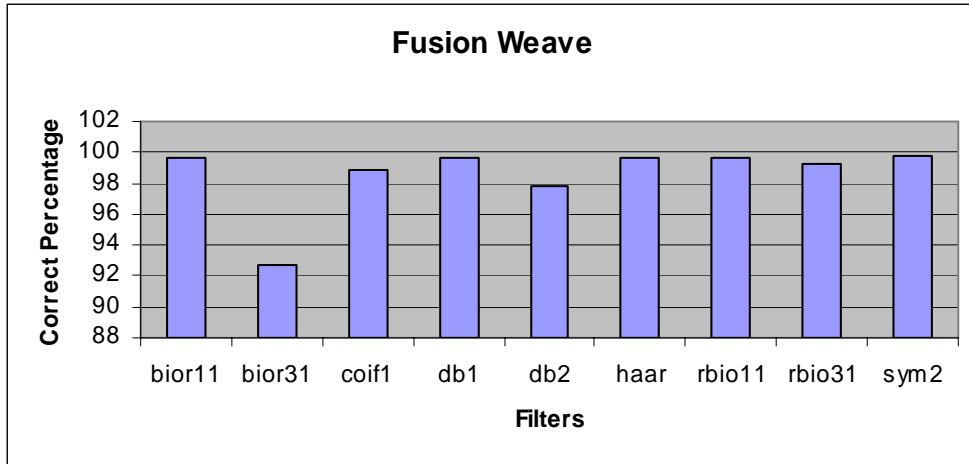


Figure 4.11 Majority Voting Weave image

4.3.4 Results using samples size of 64 by 64 with approximation coefficients

The results obtained from the classifiers are presented for each sample. We will present the results for the samples of 64 by 64. In each of them we will present tables for each filter used with the classifier.

Results for Daubechies filter ‘db1’:

Euclidean Distance Classifier Results				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	277	0	0	0	0	0	292	0	0	71	43.28		
French	0	625	0	0	2	10	0	0	3	0	97.66		
Liz	0	0	162	34	50	224	0	167	3	0	25.31		
Net	0	0	0	640	0	0	0	0	0	0	100.00		
Repskin	0	0	19	0	367	131	0	2	119	2	57.34		
Scloth	0	0	64	64	65	52	0	312	32	51	8.13		
Straw	40	0	0	0	8	0	547	0	11	34	85.47		
Strcloth	0	0	64	260	0	12	0	303	1	0	47.34		
Water	0	0	0	0	97	139	0	0	404	0	63.13		
Weave	72	0	0	0	0	0	4	0	0	564	88.13		

TABLE 4.54 Euclidean Distance results for ‘db1’ using samples of 64 by 64

Maximum Likelihood Classifier Results				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	16	115	138	88	72	29	24	65	61	32	2.50		
French	45	43	39	101	76	73	77	62	60	64	6.72		
Liz	74	126	14	43	38	47	138	44	49	67	2.19		
Net	89	91	96	6	4	22	103	14	85	130	0.94		
Repskin	63	160	60	61	44	32	68	65	41	46	6.88		
Scloth	57	132	67	41	44	52	90	43	57	57	8.13		
Straw	24	98	142	69	55	47	25	61	85	34	3.91		
Strcloth	103	121	21	27	30	47	112	20	64	95	3.13		
Water	56	161	49	54	39	30	94	51	54	52	8.44		
Weave	9	158	96	107	38	38	29	83	66	16	2.50		

TABLE 4.55 Maximum Likelihood results for ‘db1’ using samples of 64 by 64

KNN K=3 Classifier Results				Approximation Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	639	0	0	0	0	0	1	0	0	0	99.84
French	0	640	0	0	0	0	0	0	0	0	100.00
Liz	1	2	40	0	0	597	0	0	0	0	6.25
Net	0	0	0	640	0	0	0	0	0	0	100.00
Repskin	4	3	0	0	619	13	0	0	1	0	96.72
Scloth	0	0	0	0	0	0	0	639	0	1	0.00
Straw	0	0	0	0	0	0	640	0	0	0	100.00
Strcloth	1	0	0	1	0	0	0	638	0	0	99.69
Water	1	0	0	0	0	0	0	0	639	0	99.84
Weave	0	0	0	0	0	0	0	0	0	640	100.00

TABLE 4.56 K-Nearest Neighbor K=3 results for 'db1' using samples of 64 by 64

KNN K=5 Classifier Results				Approximation Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	637	0	0	0	0	0	3	0	0	0	99.53
French	0	640	0	0	0	0	0	0	0	0	100.00
Liz	1	2	77	0	0	560	0	0	0	0	12.03
Net	0	0	0	640	0	0	0	0	0	0	100.00
Repskin	2	0	0	0	437	200	0	0	1	0	68.28
Scloth	0	0	0	0	0	0	0	639	0	1	0.00
Straw	0	0	0	0	0	0	640	0	0	0	100.00
Strcloth	0	1	0	1	0	0	0	638	0	0	99.69
Water	0	0	0	0	0	0	0	0	640	0	100.00
Weave	0	0	0	0	0	0	0	0	0	640	100.00

TABLE 4.57 K-Nearest Neighbor K=5 results for 'db1' using samples of 64 by 64

Results for Coiflets filter 'coif1':

Euclidean Distance Classifier Results					Approximation Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	356	0	0	0	0	0	333	0	0	71	46.84
French	0	672	54	0	4	9	0	20	1	0	88.42
Liz	0	0	201	21	79	276	0	179	4	0	26.45
Net	0	0	0	760	0	0	0	0	0	0	100.00
Repskin	0	0	12	0	433	111	0	3	193	8	56.97
Scloth	3	0	109	72	77	50	0	358	23	68	6.58
Straw	50	0	4	0	5	7	665	0	7	22	87.50
Strcloth	0	0	70	295	2	13	0	378	2	0	49.74
Water	0	0	0	0	147	171	0	0	442	0	58.16
Weave	103	0	0	0	4	0	4	0	4	645	84.87

TABLE 4.58 Euclidean Distance results for 'coif1' using samples of 64 by 64

Maximum Likelihood Classifier Results					Approximation Coefficients						
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	22	53	222	110	41	84	28	83	81	36	2.89
French	49	66	71	73	45	66	76	132	81	101	8.68
Liz	39	238	32	43	44	36	164	53	40	71	4.21
Net	44	298	30	1	127	10	149	37	28	36	0.13
Repskin	50	135	56	79	65	54	114	93	59	55	8.55
Scloth	19	208	73	45	69	35	125	50	62	74	4.61
Straw	7	54	219	141	31	84	22	107	62	33	2.89
Strcloth	29	274	18	12	62	19	165	31	48	102	4.08
Water	52	148	64	70	56	65	92	77	74	62	9.74
Weave	17	74	170	189	25	87	33	92	61	12	1.58

TABLE 4.59 Maximum Likelihood results for 'coif1' using samples of 64 by 64

KNN K=3 Classifier Results				Approximation Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	760	0	0	0	0	0	0	0	0	0	100.00
French	1	665	94	0	0	0	0	0	0	0	87.50
Liz	1	1	208	0	0	550	0	0	0	0	27.37
Net	0	0	0	760	0	0	0	0	0	0	100.00
Repskin	1	1	0	0	754	0	0	0	4	0	99.21
Scloth	0	0	0	0	0	0	0	759	0	1	0.00
Straw	0	0	0	0	0	0	760	0	0	0	100.00
Strcloth	0	0	0	4	0	0	0	756	0	0	99.47
Water	1	0	0	0	0	8	0	0	751	0	98.82
Weave	0	0	0	0	0	0	0	0	0	760	100.00

TABLE 4.60 K-Nearest Neighbor K=3 results for 'coif1' using samples of 64 by 64

KNN K=5 Classifier Results				Approximation Coefficients							
	Classified as										
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.
Cork	760	0	0	0	0	0	0	0	0	0	100.00
French	0	665	94	1	0	0	0	0	0	0	87.50
Liz	3	2	556	0	0	199	0	0	0	0	73.16
Net	0	0	0	760	0	0	0	0	0	0	100.00
Repskin	3	2	0	0	748	0	0	0	7	0	98.42
Scloth	0	0	0	0	0	0	0	759	0	1	0.00
Straw	0	0	0	0	0	0	760	0	0	0	100.00
Strcloth	0	0	0	3	0	0	0	757	0	0	99.61
Water	1	0	0	0	0	5	0	0	754	0	99.21
Weave	0	0	0	0	0	0	0	0	0	760	100.00

TABLE 4.61 K-Nearest Neighbor K=5 results for 'coif1' using samples of 64 by 64

Results for Symlets filter 'sym2':

Euclidean Distance Classifier Results					Approximation Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	291	0	0	0	0	0	329	0	0	100	40.42	
French	0	634	51	0	3	16	0	12	4	0	88.06	
Liz	0	0	185	22	73	274	0	158	8	0	25.69	
Net	0	0	0	720	0	0	0	0	0	0	100.00	
Repskin	0	0	15	0	429	103	0	0	170	3	59.58	
Scloth	0	0	73	48	77	57	0	361	28	76	7.92	
Straw	32	0	4	0	4	1	613	0	6	60	85.14	
Strcloth	0	0	75	297	0	18	0	330	0	0	45.83	
Water	0	0	2	0	108	161	0	0	449	0	62.36	
Weave	91	0	0	0	8	0	5	0	7	609	84.58	

TABLE 4.62 Euclidean Distance results for 'sym2' using samples of 64 by 64

Maximum Likelihood Classifier Results					Approximation Coefficients							
	Classified as											
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.	
Cork	24	106	75	121	67	78	45	102	50	52	3.33	
French	72	15	18	80	93	84	85	93	72	108	2.08	
Liz	157	75	14	44	35	31	153	40	63	108	1.94	
Net	86	8	89	6	31	75	135	9	77	204	0.83	
Repskin	88	104	30	83	48	68	83	92	64	60	6.67	
Scloth	123	113	42	64	36	46	114	72	54	56	6.39	
Straw	23	94	136	107	53	79	33	118	45	32	4.58	
Strcloth	134	33	27	28	66	58	142	36	63	133	5.00	
Water	99	88	19	85	54	56	97	67	63	92	8.75	
Weave	15	105	83	140	62	56	40	124	58	37	5.14	

TABLE 4.63 Maximum Likelihood results for 'sym2' using samples of 64 by 64

KNN K=3 Classifier Results				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	652	0	0	0	0	0	68	0	0	0	90.56		
French	2	634	67	3	0	0	0	14	0	0	88.06		
Liz	0	0	250	0	0	470	0	0	0	0	34.72		
Net	0	0	0	720	0	0	0	0	0	0	100.00		
Repskin	1	1	0	0	707	6	0	0	5	0	98.19		
Scloth	0	0	0	0	0	0	0	719	0	1	0.00		
Straw	0	0	0	0	0	0	720	0	0	0	100.00		
Strcloth	0	1	0	2	0	0	0	717	0	0	99.58		
Water	2	0	0	0	0	0	0	0	718	0	99.72		
Weave	0	0	0	0	0	0	0	0	0	720	100.00		

TABLE 4.64 K-Nearest Neighbor K=3 results for 'sym2' using samples of 64 by 64

KNN K=5 Classifier Results				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	718	0	0	0	0	0	2	0	0	0	99.72		
French	1	630	76	0	0	0	0	13	0	0	87.50		
Liz	2	0	362	0	0	356	0	0	0	0	50.28		
Net	0	0	0	720	0	0	0	0	0	0	100.00		
Repskin	2	2	0	0	637	21	0	0	58	0	88.47		
Scloth	0	0	0	0	0	0	0	719	0	1	0.00		
Straw	0	0	0	0	0	0	720	0	0	0	100.00		
Strcloth	0	1	0	2	0	0	0	717	0	0	99.58		
Water	1	0	0	0	1	0	0	0	718	0	99.72		
Weave	0	0	0	0	0	0	0	0	0	720	100.00		

TABLE 4.65 K-Nearest Neighbor K=5 results for 'sym2' using samples of 64 by 64

4.3.5 Results using Fusion Method for samples size of 64 by 64

The results obtained from the classifiers were saved for each sample. After all of the classifiers finished with the samples for each filter, we entered the data in a fusion center using majority voting. We will present the results as fusion results for samples of 64 by 64. In each of them we will present tables for each filter used.

Fusion Results 'db1' for 64 by 64				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	627	0	0	0	0	0	10	0	0	3	97.97		
French	1	637	2	0	0	0	0	0	0	0	99.53		
Liz	17	16	47	2	1	557	0	0	0	0	7.34		
Net	0	0	0	640	0	0	0	0	0	0	100.00		
Repskin	27	21	4	0	532	52	0	1	3	0	83.13		
Scloth	12	15	0	0	0	0	0	612	0	1	0.00		
Straw	3	5	0	0	0	0	632	0	0	0	98.75		
Strcloth	6	5	0	1	0	0	0	628	0	0	98.13		
Water	6	5	0	0	0	0	0	0	629	0	98.28		
Weave	0	0	0	0	0	0	0	0	0	640	100.00		

TABLE 4.66 Majority Voting results for 'db1' using samples of 64 by 64

Fusion Results 'coif1' for 64 by 64				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	754	0	0	0	0	0	4	0	0	2	99.21		
French	5	662	90	3	0	0	0	0	0	0	87.11		
Liz	63	55	298	18	1	318	0	7	0	0	39.21		
Net	0	0	0	760	0	0	0	0	0	0	100.00		
Repskin	16	9	0	0	729	0	0	0	6	0	95.92		
Scloth	14	14	0	0	0	0	0	731	0	1	0.00		
Straw	3	0	0	0	0	0	757	0	0	0	99.61		
Strcloth	5	1	0	4	0	0	0	750	0	0	98.68		
Water	18	16	0	0	0	4	0	0	722	0	95.00		
Weave	2	0	0	0	0	0	0	0	0	758	99.74		

TABLE 4.67 Majority Voting results for 'coif1' using samples of 64 by 64

Fusion Results 'sym2' for 64 by 64				Approximation Coefficients									
	Classified as												
Data	Cork	French	Liz	Net	Repskin	Scloth	Straw	Strcloth	Water	Weave	% Classif.		
Cork	672	1	1	0	0	2	36	0	0	8	93.33		
French	5	628	69	3	2	0	0	13	0	0	87.22		
Liz	19	18	275	4	0	400	0	4	0	0	38.19		
Net	0	0	0	720	0	0	0	0	0	0	100.00		
Repskin	24	16	0	0	666	7	0	0	7	0	92.50		
Scloth	7	4	0	0	0	0	0	708	0	1	0.00		
Straw	3	1	0	0	0	0	716	0	0	0	99.44		
Strcloth	6	3	0	1	0	0	0	710	0	0	98.61		
Water	13	10	0	0	1	0	0	0	696	0	96.67		
Weave	3	0	0	0	0	0	0	0	0	717	99.58		

TABLE 4.68 Majority Voting results for 'sym2' using samples of 64 by 64

4.3.6 Histograms of samples from the classes used

Here we present the histograms for some samples of the classes that we used from the Brodatz album. These histograms are for the approximation and diagonal coefficients given that these are the most valuable in information; because the approximation coefficients are the results that comes from the Low-Low filter thus are the DC values, and the diagonal coefficients comes from the High-High filter thus are the AC values where the texture information is stored. These histograms can help us to understand why the mismatch classification for some classes.

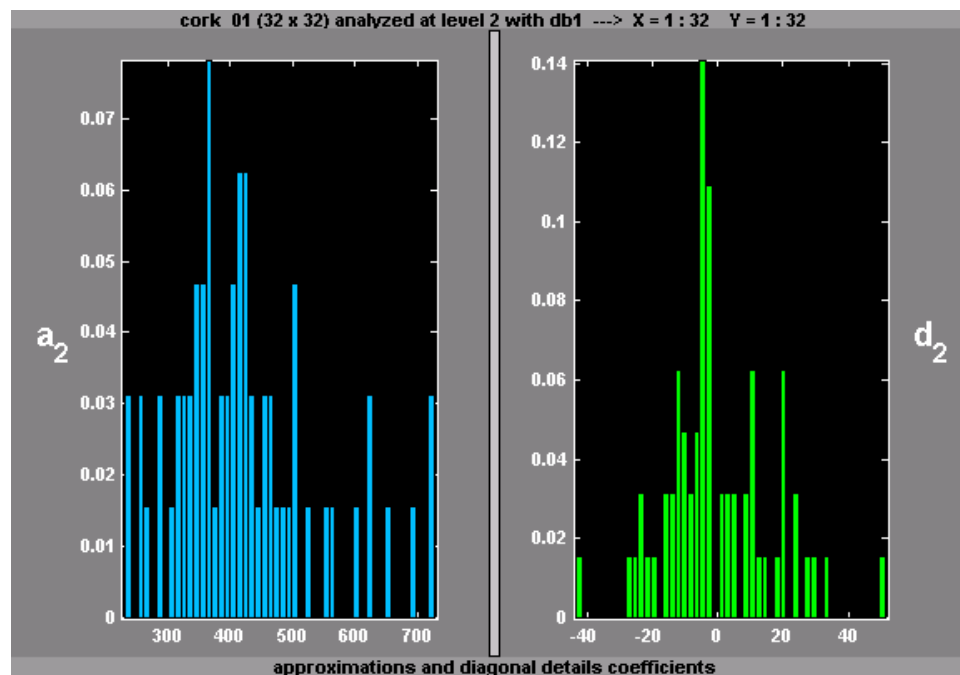


Figure 4.12 Histograms of Cork sample approximation and diagonal coefficients

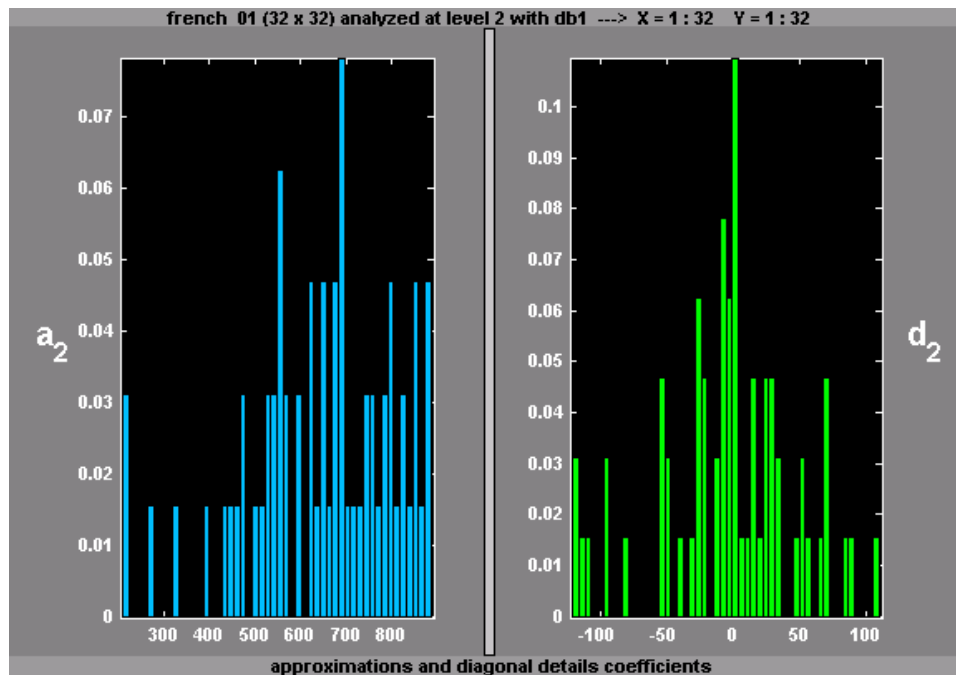


Figure 4.13 Histograms of French sample approximation and diagonal coefficients

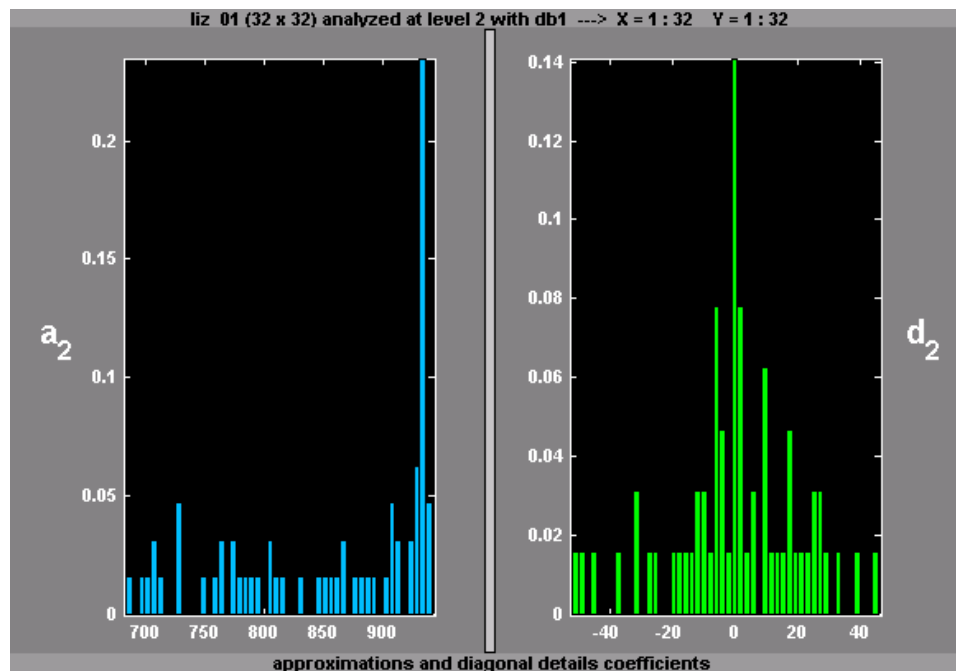


Figure 4.14 Histograms of Liz sample approximation and diagonal coefficients

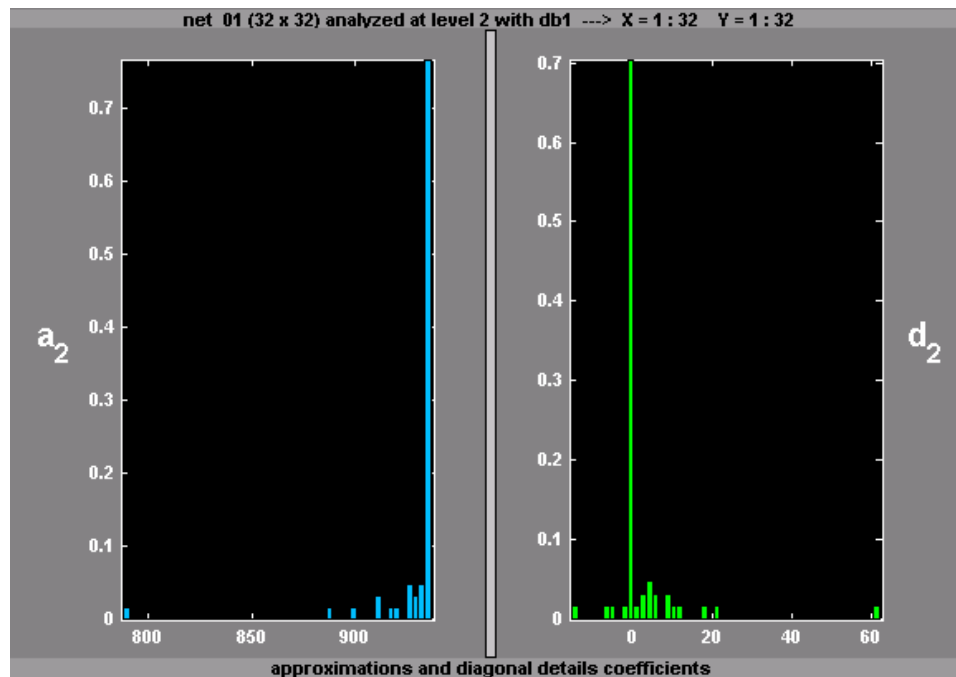


Figure 4.15 Histograms of Net sample approximation and diagonal coefficients

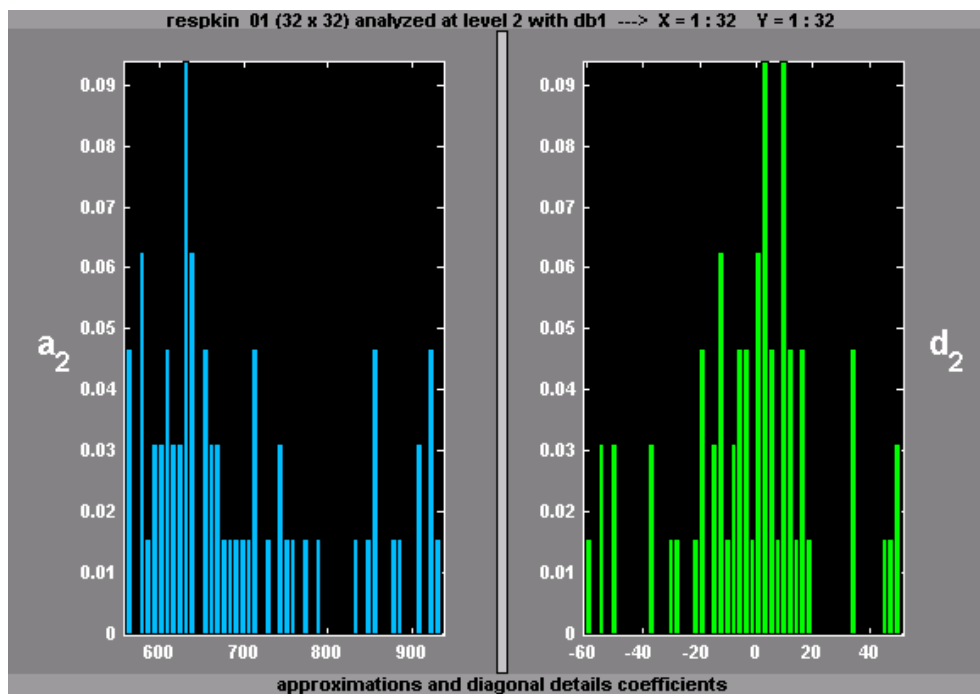


Figure 4.16 Histograms of Repskin sample approximation and diagonal coefficients

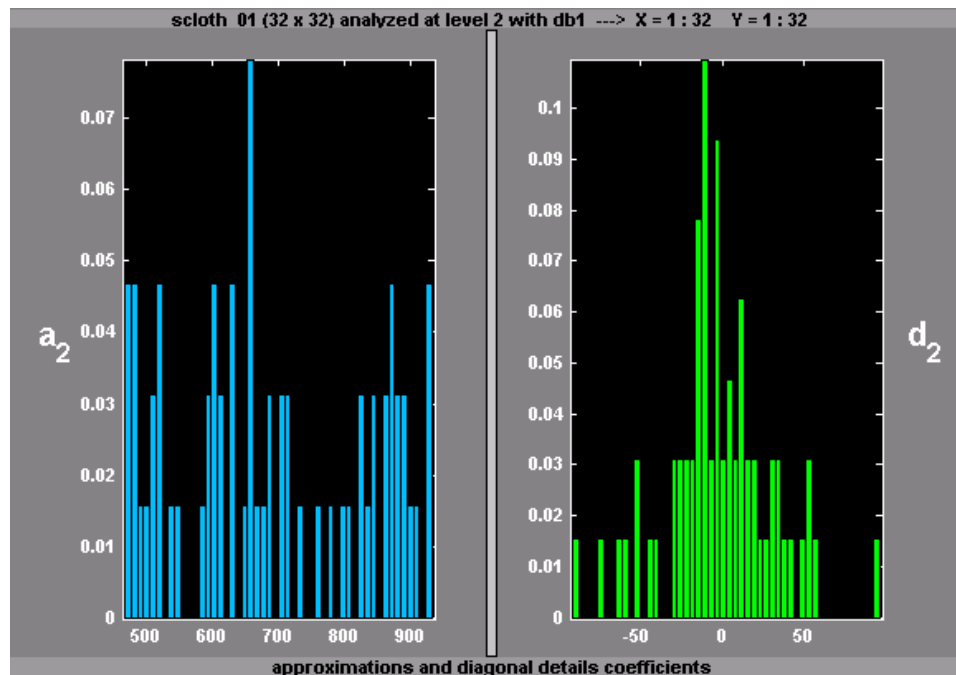


Figure 4.17 Histograms of Scloth sample approximation and diagonal coefficients

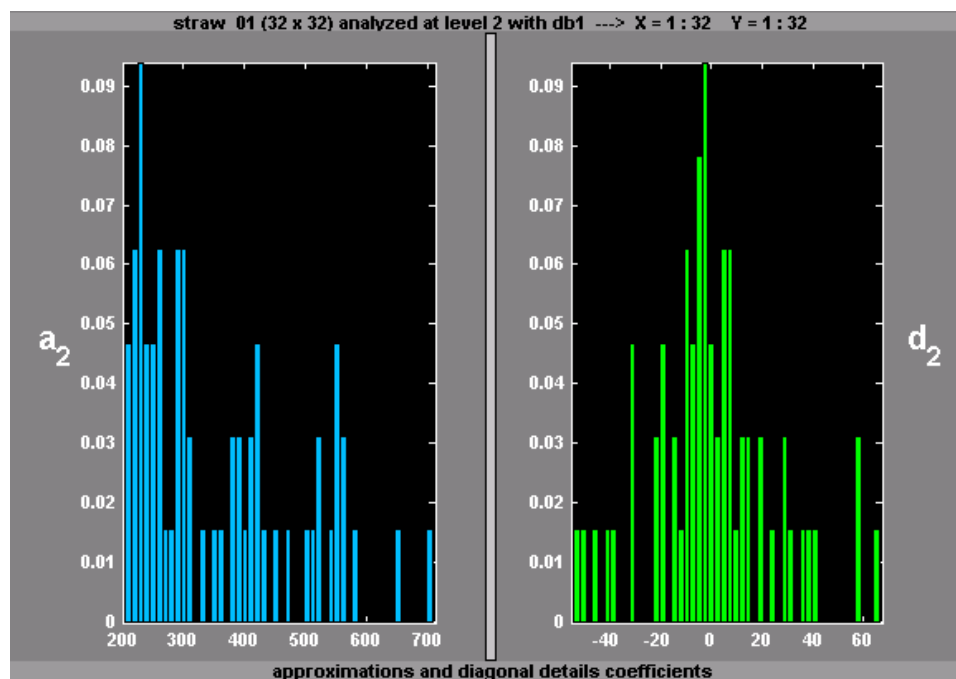


Figure 4.18 Histograms of Straw sample approximation and diagonal coefficients

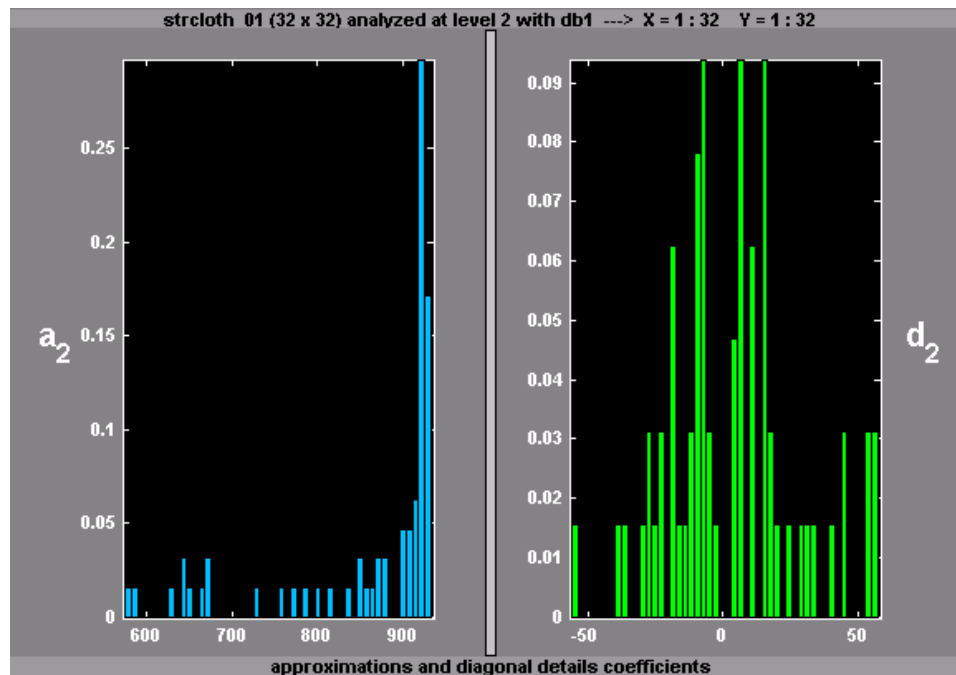


Figure 4.19 Histograms of Strcloth sample approximation and diagonal coefficients

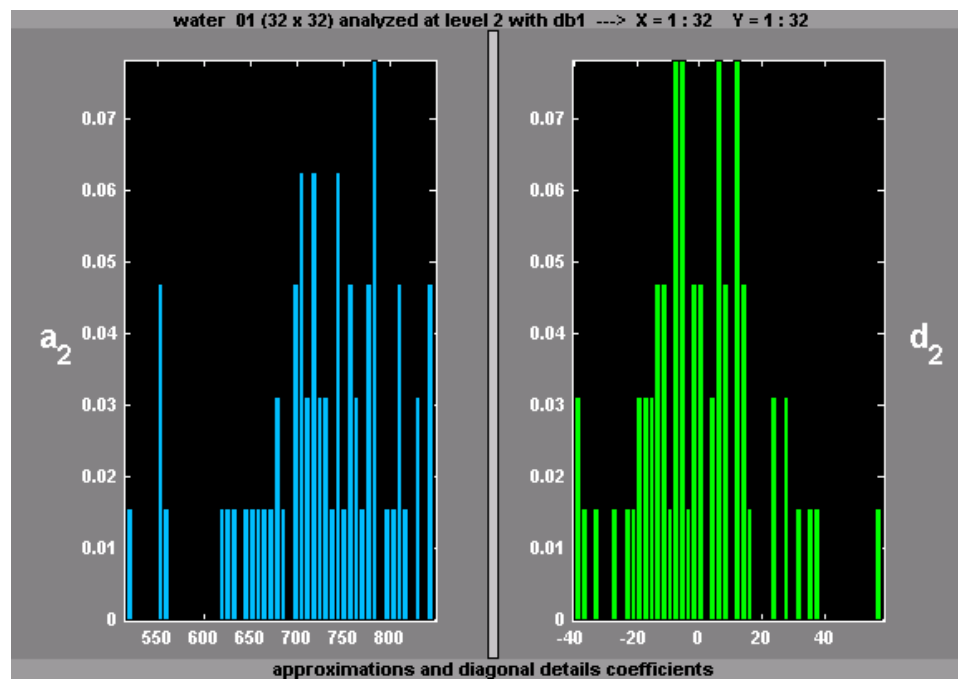


Figure 4.20 Histograms of Water sample approximation and diagonal coefficients

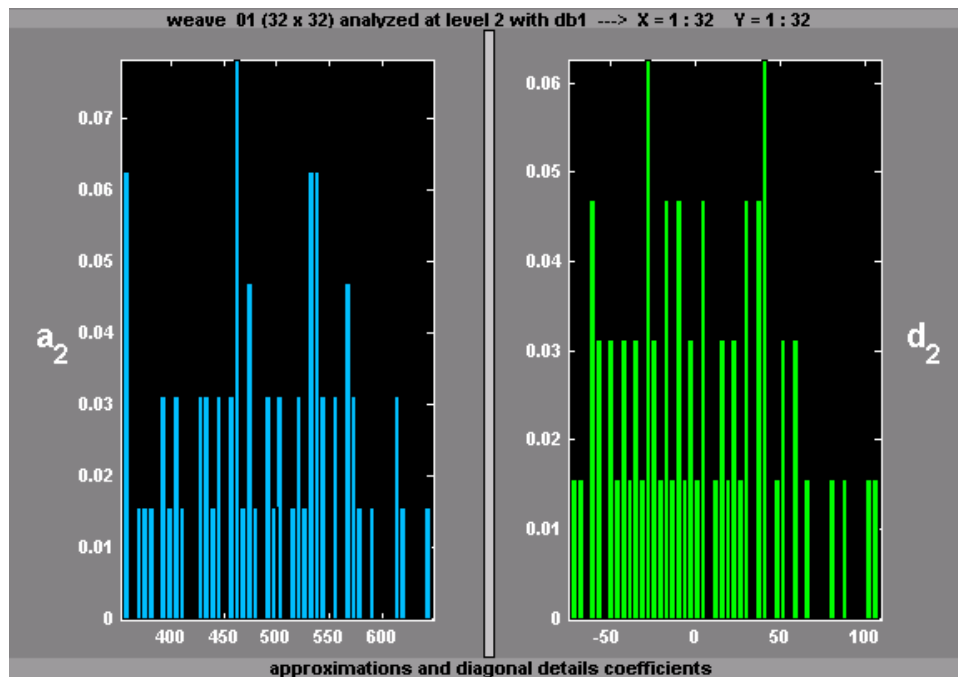


Figure 4.21 Histograms of Weave sample approximation and diagonal coefficients

5 CONCLUSIONS AND FUTURE WORK

For our work in image classification using wavelet filters and fusion methods we performed experiments using ten Brodatz album textures. We used algorithms with nine different wavelet filters to extract the features from the images. The classifiers used were Euclidean Distance, K-Nearest Neighbor for K=3, K-Nearest Neighbor for K=5 and Maximum Likelihood as a way to evaluate the features extracted with the filters.

K-Nearest Neighbor for K=3 and K=5 classifiers performed the best using wavelet filters for the images used in the experiments when using only a single coefficient. Maximum Likelihood did not perform well when using only one coefficient, because for the calculation of the covariance matrix we need at least two different coefficients, otherwise the covariance matrix becomes a singular matrix. For the case where these covariance matrices resulted singular, we tried using Singular Value Decomposition, QR Decomposition and Pseudo Inverse, but the inverse of the covariance matrix was not accurate thus the bad results. Euclidean Distance classifier gave really good results when using a single coefficient because in this case we used the approximation coefficients. These coefficients come from the Low-Low filtering thus the contained the DC values that are the means of these coefficients, that's why it gave those good classification results. Because if the coefficients values are the means when calculating the Euclidean Distance with the means calculated, there is a really good probability that the distance with the same class would be the nearest to zero.

We tried combining the diagonal coefficients the High-High with the approximation Low-Low, because the diagonal coefficients have the AC values which contains the valuable information about texture, they gave us good results for some cases. We got really good results when testing our algorithms with the image of Mayagüez Bay and using all of the features. We made different experiments that did not give us the results that we expected, for example the ones with the variance as a measure of activity, we could not get better classification than the 42.5%.

The Scloth and Liz image were the images that gave the worst results in all classifiers. As you can see in the histograms of those images in Figure 4.14 and Figure 4.17 these classes tends to covered by the histograms of the others, which justifies the mismatch. The Net image was the one that gave the best results in all classifiers, not including Maximum Likelihood when using only the approximation coefficient; it always was classified correct 100% no matter which filter was used for the feature selection and even for the cases using the four coefficients it gave us excellent results. This excellent result can be explained by its histogram see Figure 4.15, as can be seen the histogram really differs from the other ones of the different classes. We can watch how the results using twice as bigger the sample size (64 by 64) were better than the original size of 32 by 32 samples from the image because there was overlapping in these samples; some of them already had the exact same parts.

The results using Majority Voting are better overall than the single classifier for the case of 64 by 64 sample size. For the 32 by 32 the results weren't that great as the 64 by

64 because 5 out of ten images got really low percentage of correct classification. But overall for all the filters and classifier used the results from the fusion methods were actually improved and were better than that of any single classifier overall result.

For future work we propose trying different Fusion Methods and a different approach for image classification using wavelets for classifying the image into regularity, directionality, symmetry, etc., with this kind of feature we could separate the samples of the images before using classifiers. We propose also to attack the problem with a more mathematical approach.

6 USER GUIDE FOR THE ALGORITHMS

This is a user guide to the algorithms; we will present one of each kind for a particular case. All of the algorithms are well documented; we will explain some of the parts that we think may confuse the user.

1. Selection of Testing and Training Samples of 32 by 32 (*samples_fcn.m*)

- a) Change the name in “`x=imread('d4_cork.tif');`”, the part in ' ' is the file.tif, just change the file and the extension `_cork`.
- b) In “`x_mat`” we have the size of the window that we want to use, in our case 32 by 32.
- c) This algorithm must be in the same folder as the images that are going to be used are.
- d) You have to do this for the ten images used then save the results. We called the results “`data_double.mat`” because we had to convert into doubles the matrices that we rearranged.

For more info see the algorithm.

2. Filtering, Feature Selection, Training and Testing Samples Selection (*train_test_selec_n.m*)

- a) Every file that is loaded must be in the same folder as the algorithm.

- b) “*load data_double.mat*” loads the matrices that we got from the selection algorithm from before. In this “.mat” file we have the 10 matrices of the 10 images calculated.
 - c) In the for-loops we filter the samples using wavelet decomposition when calling the function “*wave_fcn_n.m*” (for more info look the “*wave_fcn_n.m*” algorithm).
 - d) There are 64 Samples we must decide which are going to be Testing and Training, that’s why we used the “*randperm*” command to generate a random vector that will help us to decide and select which are going to be testing and which training samples.
 - e) By changing the number of training samples we can have different solutions
 - f) “*training_testing_rand_dbl.mat*” is the name of the file that we save from the results. This file “.mat” have the arrays “*test_cork*” through “*test_weave*” and “*train_cork*” through “*train_weave*”.
3. Function for sample filtering wavelet decomposition (*wave_fcn_n.m*)
- a) Here we used the command “*dwt2*” to do the discrete wavelet transform.
 - b) You can change the filter used by changing the name of the filter between ‘’. Example: “*dwt2(x,'rbio3.1');*” the filter used in this case is the Reverse Biorthogonal 3.1 ‘*rbio3.1*’.
 - c) “[*cA1,cH1,cV1,cD1*]” and “[*cA2_L,cH2_L,cV2_L,cD2_L*]” “A” means the approximation coefficients, “1” in the first level decomposition, “2” in the

second level decomposition, “ H ” means the horizontal coefficients, “ V ” means the vertical coefficients and “ D ” means the diagonal coefficients.

- d) We then normalize the coefficients.
- e) The covariance matrix and mean vector are calculated.
- f) A new array is formed that has the mean vector, the covariance matrix and the coefficients from the second level decomposition for every sample that enters.

In our research we used different “*wave_fcn*” that had different final arrays depending on what we needed.

4. Euclidean Distance Classifier (*eu_app_test4.m*)

- a) First we load the file “*training_testing_rand_db1.mat*” created in (*train_test_selec_n.m*).
- b) Then we calculate the distances with respect to the training samples to the testing using “*eu_fcn*” that is a function that has the discriminant function for Euclidean Distance.
- c) We then classify the results.
- d) When its finished with the classification you can see in the variable “*rec_perc*” the confusion matrix and the percentage of the classification as the last column from left to righth.
- e) For later use in the fusion center you have to save the array “*results*”. This array contains the history of what sample was labeled to which class. This is very useful because we have to fusion the results of the classifier separately.

For more info see the algorithm (*eu_app_test4.m*) because is well commented.

5. Discriminant Function For Multivariate & Multiple Classes (*eu_fcn.m*) and (*hii_fcn.m*)

- a) These are function created to calculate the discriminant functions of Euclidean Distance and Maximum Likelihood.

For more reference see these functions that are well commented.

6. K-Nearest Neighbor K=3 and K=5 Classifiers

- a) There are almost the same as Euclidean Distance algorithms but with the difference of the part of the classification that compare the distances depending on the number K, to give a result from the classification.
- b) If its wanted the user can change the discriminant function, now is using the (*eu_fcn.m*) to calculate the distance, but if wanted just change it for another one. For example the (*hii_fcn.m*) to calculate Maximum Likelihood distance.

See the algorithm for more info.

7. Maximum Likelihood Classifier

- a) Same procedure as the other classifier algorithms, but with the difference that you have now the covariance matrix in the discriminant function.
- b) All of the other procedures like loading the data and everything still the same.

For more info see the algorithm.

8. Majority Voting Fusion Center

- a) We first load the results obtained by the individual classifiers.

- b) Then we do the majority voting and assign the labels to the pixels.
- c) Then we calculate the percentage of correct classification and we generate the confusion matrix along with the correct percentage results.

REFERENCES

- [1] B. E. Usevitch, "A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000", *IEEE Signal Processing Magazine*, 18(5), pp. 22-35, September 2001.
- [2] K. S. Thyagarajan, J. Nguyen and C. E. Persons, "A Maximum Likelihood Approach to Texture Classification Using Wavelet Transform", *Proceedings of IEEE International Conference on Image Processing "ICIP-II"*, VOL 94, pp. 640-644, Thessaloniki, Greece, October 2001.
- [3] L. Balmelli and A. Mojsilovic, "Wavelet Domain Features for Texture Description, Classification and Replicability Analysis", *IEEE International Conference on Image Processing, ICIP'99*, Kobe, Japan, October 24-28, 1999.
- [4] D. Costantino, A. Greco, F. C. Morabito and M. Versaci, "A Morlet Wavelet Classification Technique for ICA filtered sEMG experimental Data", *International Joint Conference on Neural Networks "IJCNN 2003"*, Special Session M3S: Knowledge Discovery, and Image and Signal Processing in Medicine, [#496], Portland, Oregon, July, 2003.
- [5] G. Loum, P. Provent, J. Lemoine and E. Petit "A New Method for Texture Classification Based on Wavelet Transform", *Proc. 3rd IEEE Symposium "Time-Frequency and Time Scale Analysis"*, Paris, France, June 1996.

- [6] Z. Jingyuan, J. Xingzhou and Y. Bingcheng “Adaptive Wavelets Classification of Transient Sonar Signals”, *Proceedings of IEEE Computer Society “4th International Conference on the Software Process” (ICSP)*, Brighton, United Kingdom, December 1996.
- [7] L. Pierce, G. Samples, M. C. Dobson and F. Ulaby, “An Automated Unsupervised/Supervised Classification Methodology”, *IEEE International Geoscience and Remote Sensing Symposium Proceedings (IGARSS)*, Seattle, Washington, pp.1781-1783, July 1998.
- [8] J. F. Liu and J. C. M. Lee, “An Efficient and Effective Texture Classification Approach Using a New Notion in Wavelet Theory”, *Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, Vienna, Austria, Vol. 2, pp. 820-824, August 1996.
- [9] T. Y. Chen and P. L. Poon, “Classification-Hierarchy Table: A Methodology for Constructing the Classification Tree”, *Proceedings of the Australian Software Engineering Conference (ASWEC 96)*, Melbourne, Australia, pp. 93-104, July 1996.
- [10] X. Jia and J. A. Richards, “Cluster-Space Representation for Hyperspectral Data Classification” *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40, pp. 593-598, March 2002.

- [11] W. Li, J. Han and J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules", *Proceedings 2001 International Conference on Data Mining (ICDM'01)*, San Jose, CA, November 2001.
- [12] B. Jeon and D. A. Landgrebe, "Decision Fusion Approach for Multitemporal Classification", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37, pp. 1227-1233, May 1999.
- [13] V. DeBrunner and M. Kadiyala, "Effect of Wavelet Bases in Texture Classification Using a Tree-Structured Wavelet Transform", *IEEE Signal Processing Society Conference record of the "Thirty-Third Asilomar Conference on Signals, Systems, and Computers"*, Vol. 2, pp. 1292 -1296, October 1999.
- [14] S. Luchian, H. Luchian and M. Petriuc, "Evolutionary Automated Classification", *Proceedings of the 1st IEEE World Congress on Computational Intelligence*, Vol. 2, pp. 585-588, 1994.
- [15] H.A. Kestler, M. Hoher and G. Palm, "Evolutionary Optimization of a Wavelet Classifier for the Categorization of Beat-to-Beat Variability Signals", *5th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA 2001)*, Czech Republic, April 2001.
- [16] C. Tremblay and P. Valin, "Experiments on Individual Classifiers and on Fusion of a Set of Classifiers", *Proc. 2002 International Conference on Information Fusion (ISIF 2002)*, Annapolis, MD, pp. 272-277, July 2002.

- [17] D. Endo, T. Hiyane, K. Atsuta and S. Kondo, "Fractal Image Compression by the Classification in the Wavelet Transform Domain", *Proceeding of 5th IEEE International Conference on Image Processing (ICIP98)*, pp.190-193, New York, August 1998.
- [18] J. Ediriwickrema and S. Khorram, "Hierarchical Maximum-Likelihood Classification for Improved Accuracies", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 35, No. 4, pp. 810-816, July 1997.
- [19] L. Roux and J. Desachy, "Information fusion for supervised classification in a satellite image", *Fourth IEEE International Conference on Fuzzy Systems FUZZ-IEEE'95*, Vol. 3, pp. 1119-1124, Yokohama, Japan, March 1995.
- [20] S. Liapis, N. Alvertos and G. Tziritas "Maximum Likelihood Texture Classification and Bayesian Texture Segmentation Using Discrete Wavelet Frames", *Intern. Conference on Digital Signal Processing*, Santorini, 1997.
- [21] N. Fletcher and A. Evans, "Minimum Distance Texture Classification of SAR Images Using Wavelet Packets", *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS'02)*, Toronto, Ontario Canada, 24-28, June 2002.
- [22] I. Jouny and M. Kanapathipillai, "Neural Networks Adaptive Wavelet Classification of Radar Targets", *IEEE Geoscience and Remote Sensing Symposium, 1994 (IGARSS '94)*, 'Surface and Atmospheric Remote Sensing:

- Technologies, Data Analysis and Interpretation', International, Vol. 4, 8-12 August 1994.
- [23] M. K. Tsatsanis and G. B. Giannakis, "Object and Texture Classification Using Higher Order Statistics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, Issue: 7, pp. 733 – 750, July 1992.
- [24] W.-S. Lu and A. Antoniou, "Optimized Orthogonal and Biorthogonal Wavelets Using Linear Parameterization of Halfband Filters", *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98)*, Vol. 5, pp. 102 – 105, 31 May-3 June 1998.
- [25] R. L. Joshi, T. R. Fischer and R. H. Bamberg, "Optimum Classification in Subband Coding of Images", *Proceedings IEEE International Conference Image Processing (ICIP-94)*, Vol. 2, pp. 883 - 887, 13-16 November 1994.
- [26] R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes", *IEE Proceedings Vision, Image and Signal Processing*, Vol. 144, Issue: 3, pp. 180 – 188, June 1997.
- [27] C.-M. Pun and M.-C. Lee, "Rotation-invariant texture classification using a two-stage wavelet packet feature approach", *IEE Proceedings Vision, Image and Signal Processing*, Vol. 148, Issue: 6, pp. 422 – 428, December 2001.
- [28] P. Hill, C. N. Canagarajah and D. R. Bull, "Statistical Wavelet Subband Modelling for Texture Classification", *Proceedings of 2001 International Conference on Image Processing*, Vol. 1 , pp. 165 – 168, 7-10 Oct. 2001

- [29] S. N. Talbar, R. S. Holambe and T. R. Sontakke, "Supervised Texture Classification Using Wavelet Transform", *1998 Fourth International Conference on Signal Processing Proceedings (ICSP '98)*, Vol. 2, pp. 1177 - 1180, 12-16 Oct. 1998
- [30] N. M. Rajpoot, "Texture Classification Using Discriminant Wavelet Packet Subbands", *The 2002 45th Midwest Symposium on Circuits and Systems (MWSCAS-2002)*, Vol. 3, pp. III-300 - III-303, 4-7 Aug. 2002.
- [31] Moon-Chuen Lee and Chi-Man Pun, "Texture Classification Using Dominant Wavelet Packet Energy Features", *Proceedings of 4th IEEE Southwest Symposium Image Analysis and Interpretation*, pp. 301 – 304, 2-4 April 2000.
- [32] S. Hatipoglu, S. K. Mitra and N. Kingsbury, "Texture Classification Using Dual-Tree Complex Wavelet Transform", *Seventh International Conference on Image Processing And Its Applications*, (Conf. Publ. No. 465), Vol. 1, pp. 344 – 347, 13-15 July 1999.
- [33] L. Hepplewhite and T. J. Stonham, "Texture Classification Using N-Tuple Pattern Recognition", *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 4, pp. 159 – 163, 25-29 August 1996.
- [34] K. Sam-Deuk and S. Udpa, "Texture Classification Using Rotated Wavelet Filters", *IEEE Transactions on Systems, Man and Cybernetics*, Part A, Vol. 30, Issue: 6, pp. 847 – 852, November 2000.

- [35] L. Wang, J. Liu and S. Z. Li, "Texture Classification Using Wavelet Decomposition with Markov Random Field Models", *Proceedings of Fourteenth International Conference on Pattern Recognition*, Vol. 2, pp. 1613 – 1615, 16-20 Aug. 1998.
- [36] A. Van Nevel, "Texture Classification Using Wavelet Frame Decomposition", *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, Vol. 1, pp. 311 – 314, 2-5 November 1997.
- [37] Tang Xiaou and W.K. Stewart, "Texture Classification Using Wavelet Packet and Fourier Transforms", *Conference Proceedings of OCEANS '95. MTS/IEEE. 'Challenges of Our Changing Global Environment'*, Vol. 1, pp. 387 – 396, 9-12 October 1995.
- [38] A. Busch and W. W. Boles, "Texture Classification Using Wavelet Scale Relationships", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, Vol. 4, pp. IV-3584 - IV-3587, 13-17 May 2002.
- [39] V. DeBrunner and M. Kadiyala, "Texture Classification Using Wavelet Transform", *42nd Midwest Symposium on Circuits and Systems*, Vol. 2, pp. 1053 – 1056, 8-11 August 1999.
- [40] T. Chang and C.-C.J. Kuo, "Texture Classification with Tree-Structured Wavelet Transform", *Proceeding of 11th IAPR International Conference on Pattern*

- Recognition, 'Vol.II. Conference B: Pattern Recognition Methodology and Systems', pp. 256 – 259, 30 Aug.-3 Sept. 1992.*
- [41] Luo Jiebo and A. E. Savakis, “Two Stage Texture Segmentation Using Complementary Features”, *Proceedings of 2000 International Conference on Image Processing*, Vol. 3, pp. 564 – 567, 10-13 Sept. 2000.
- [42] J. Puniene, V. Punys and J. Punys, “Ultrasound and Angiographic Image Compression by Cosine and Wavelet Transforms and its Approval in Clinical Environment”, *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis (ISPA2001)*, pp. 74 – 79, 19-21 June 2001.
- [43] G. F. McLean, “Vector Quantization for Texture Classification”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, Issue: 3, pp. 637 – 649, May-June 1993.
- [44] C. Reyes and M. Adjouadi, “A Clustering Technique for Random Data Classification”, *IEEE International Conference on Systems, Man and Cybernetics, 'Intelligent Systems for the 21st Century'*, Vol. 1, pp. 316 – 321, 22-25 Oct. 1995.
- [45] N. Sebe and M. S. Lew, “Wavelet Based Texture Classification”, *Proceedings of 15th International Conference on Pattern Recognition*, Vol. 3, pp. 947 – 950, 3-7 Sept. 2000.
- [46] Wu Gaohong, Zhang Yujin and Lin Xinggang “Wavelet Transform-based Texture Classification with Feature Weighting”, *Proceedings of 1999 International*

- Conference on Image Processing (ICIP 99)*, Vol. 4, Pages:435 – 439, 24-28 Oct. 1999.
- [47] D. Charalampidis and T. Kasparis, “Wavelet-Based Rotational Invariant Roughness Features for Texture Classification and Segmentation”, *IEEE Transactions on Image Processing*, Vol. 11, Issue: 8, pp. 825 – 837, Aug. 2002.
- [48] A. Tabesh, “Zero-crossing rates of wavelet frame representations for texture classification”, *Electronics Letters*, Vol. 38, Issue: 22, pp. 1340 – 1341, 24 Oct 2002.
- [49] L. I. Kuncheva, “A Theoretical Study on Six Classifier Fusion Strategies”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 24, NO. 2, February 2002.
- [50] D. A. Landgrebe, “Signal Theory Methods in Multispectral Remote Sensing”, New Jersey: John Wiley & Sons, 2003.
- [51] G. Strang and T. Nguyen, “Wavelet and Filter Banks”, Massachusetts: Wellesley-Cambridge Press, 1996.
- [52] J. A. Richards and X. Jia, “Remote Sensing Digital Image Analysis”, Germany: Springer-Verlag, Third revised and enlarged edition, 1999.
- [53] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, “On Combining Classifiers”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, NO. 3, March 1998.

- [54] L. O. Jimenez Rodriguez, "Laboratory for Applied Remote Sensing and Image Processing" University of Puerto Rico, 2002-2003.
- [55] M. Misiti, Y. Misiti, G. Oppenheim and J. M. Poggi, "Wavelet Toolbox for use with Matlab", The Math Works Inc., Version 2.2, July 2002.
- [56] V. Manian and R. Vasquez, "Texture discrimination based on neural dynamics of visual perception", *Proceedings of the International Joint Conference on Neural Networks*, Vol. 1 , pp. 113 – 118, 20-24 July 2003.
- [57] L. O. Jimenez, A. Morales-Morell and A. Creus, "Classification of Hyperdimensional Data Based on Feature and Decision Fusion Approaches Using Projection Pursuit, Majority Voting, and Neural Networks", *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 37, NO. 3, March 1999.
- [58] J. Malik, S. Belongie, J. Shi and T. Leung, "Textons, contours and regions cue integration in image segmentation", *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999. Volume 2, 20-27 Sept. 1999 Page(s):918 - 925 vol.
- [59] F. C. A. Fernandes, R. L. C. van Spaendonck, and C. Sidney Burrus, "Multidimensional, Mapping-Based Complex Wavelet Transforms", *IEEE Transactions on Image Processing*, Vol. 14, No. 1, January 2005.
- [60] L. Zhang, P. Bao, and X. Wu, "Multiscale LMMSE-Based Image Denoising With Optimal Wavelet Selection", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 4, April 2005.

- [61] A. Gupta, S. Dutt Joshi, and S. Prasad, "A New Approach for Estimation of Statistically Matched Wavelet", *IEEE Transactions on Signal Processing*, Vol. 53, No. 5, May 2005.
- [62] D. Li, W. Pedrycz, and N. J. Pizzi, "Fuzzy Wavelet Packet Based Feature Extraction Method and Its Application to Biomedical Signal Classification", *IEEE Transactions on Biomedical Engineering*, Vol. 52, No. 6, June 2005
- [63] X. Zhang, N. H. Younan, and C. G. O'Hara, "Wavelet Domain Statistical Hyperspectral Soil Texture Classification", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 3, March 2005.

APPENDIX A. SAMPLES SELECTION

APPENDIX A1 CODES FOR 32 BY 32 SAMPLE SIZE WITHOUT OVERLAPPING

```
% Selection of Testing and Training Samples of 32 by 32
% x=imread('d4_cork.tif'); the part in '' is the file.tif,
% just change the file and the extension _cork

xa=imread('d15_straw.tif');
x=xa(1:256,1:256);

for i=1:8
    for j=1:8
        x_mat=x((j-1)*32+1:j*32,(i-1)*32+1:i*32); % takes the part of the matrix for training sample
        x_col=im2col(x_mat,[32 32],'sliding'); % converts the matrix in to a column vector
        x_8element(j,:)=x_col';
    end
    x_straw((i-1)*8+1:i*8,:)=x_8element; % generates a matrix of row vector
end
xx_straw=double(x_straw);
```

APPENDIX A2 CODES FOR 64 BY 64 SAMPLE SIZE WITHOUT OVERLAPPING

```
% Selection of Testing and Training Samples of 64 by 64 out of an image of
% 512 by 512 without overlapping
% x=imread('d4_cork.tif'); the part in '' is the file.tif,
% just change the file and the extension _cork

xa=imread('d22_repskin.tif');
x=xa(1:512,1:512);

for i=1:8
    for j=1:8
        x_mat=x((j-1)*64+1:j*64,(i-1)*64+1:i*64); % takes the part of the matrix for training sample
        x_col=im2col(x_mat,[64 64],'sliding'); % converts the matrix in to a column vector
        x_8element(j,:)=x_col';
    end
    x_repskin((i-1)*8+1:i*8,:)=x_8element; % generates a matrix of row vector
end
xx_repskin=double(x_repskin);
```

APPENDIX A3 CODES FOR 64 BY 64 SAMPLE SIZE WITH OVERLAPPING

```
% Selection of Testing and Training Samples of 64 by 64
% x=imread('d4_cork.tif'); the part in '' is the file.tif,
% just change the file and the extension _cork

% This algorithm takes 64 samples of 64 by 64 from an image of 256 by 256
% with overlapping. In the first "for loop" it takes 52 samples, that's 13
% samples out of each "i-column" (for i=1:4), so 13*4=52. The second "for
% loop" takes 12 samples.

xa=imread('d15_straw.tif');
x=xa(1:256,1:256);

for i=1:4      % i ~ columns
    for j=1:13 % j ~ rows
        x_1mat=x((j-1)*16+1:(j-1)*16+64,(i-1)*64+1:i*64); % takes the part of the matrix for training
        sample
        x_1col=im2col(x_1mat,[64 64],'sliding'); % converts the matrix in to a column vector
        x_13element(j,:)=x_1col';
    end
    x_1straw((i-1)*13+1:i*13,:)=x_13element; % generates a matrix of row vector
end

for i=1:3      % i ~ columns
    for j=1:4   % j ~ rows
        x_2mat=x((j-1)*64+1:j*64,(2*i-1)*32+1:(2*i-1)*32+64); % takes the part of the matrix for
        training sample
        x_2col=im2col(x_2mat,[64 64],'sliding'); % converts the matrix in to a column vector
        x_3element(j,:)=x_2col';
    end
    x_2straw((i-1)*4+1:i*4,:)=x_3element; % generates a matrix of row vector
end

x_straw=[x_1straw ; x_2straw];

xx_straw=double(x_straw);
```

APPENDIX B PROGRAMS FOR FILTERING, TRAINING AND TESTING SAMPLES SELECTION

APPENDIX B1 PROGRAM FOR TRAINING AND TESTING SAMPLES SELECTION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Filtering, Feature Selection, Training and Testing Samples Selection %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% for high-high variance measure of activity  
% This algorithm generates the Training and Testing samples for image  
% classification, we will make different training and testing ".mat files"  
% The Filtering & Feature Selection is made by the "wave_fcn()" see  
% waveh_fcn.m for more info  
clear  
% loading the matrices of the images samples  
load data_double.mat  
% using the same random vectors as the first data selection  
load random_vectors.mat  
  
% wavelet decomposition  
% cork data  
for i=1:64  
    x=reshape(xx_cork(i,:),32,32);  
    cork_vec(i*4-3:i*4,:)=wave_fcn_n(x);
```

```

    % cork_vec(i,:)=wave_fcn_n(x);
end
% straw data
for i=1:64
    x=reshape(xx_straw(i,:),32,32);
    straw_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % straw_vec(i,:)=wave_fcn_n(x);
end
% repskin data
for i=1:64
    x=reshape(xx_repskin(i,:),32,32);
    repskin_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % repskin_vec(i,:)=wave_fcn_n(x);
end
% strcloth data
for i=1:64
    x=reshape(xx_strcloth(i,:),32,32);
    strcloth_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % strcloth_vec(i,:)=wave_fcn_n(x);
end
% weave data
for i=1:64
    x=reshape(xx_weave(i,:),32,32);
    weave_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % weave_vec(i,:)=wave_fcn_n(x);
end

```

```

% french data
for i=1:64
    x=reshape(xx_french(i,:),32,32);
    french_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % french_vec(i,:)=wave_fcn_n(x);
end

% net data
for i=1:64
    x=reshape(xx_net(i,:),32,32);
    net_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % net_vec(i,:)=wave_fcn_n(x);
end

% liz data
for i=1:64
    x=reshape(xx_liz(i,:),32,32);
    liz_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % liz_vec(i,:)=wave_fcn_n(x);
end

% water data
for i=1:64
    x=reshape(xx_water(i,:),32,32);
    water_vec(i*4-3:i*4,:)=wave_fcn_n(x);
    % water_vec(i,:)=wave_fcn_n(x);
end

% scloth data
for i=1:64

```

```

x=reshape(xx_scloth2(i,:),32,32);
scloth_vec(i*4-3:i*4,:)=wave_fcn_n(x);
% scloth_vec(i,:)=wave_fcn_n(x);
end

```

% There are 64 Samples we must decide which are going to be Testing and Training

```

training=24; % by changing the number of training samples we can have diferent solutions
% tt_cork=randperm(64); % this part is for randomizing the picking of the training and testing samples
% tt_straw=randperm(64);
% tt_repskin=randperm(64);
% tt_strcloth=randperm(64);
% tt_weave=randperm(64);
% tt_french=randperm(64);
% tt_net=randperm(64);
% tt_liz=randperm(64);
% tt_water=randperm(64);
% tt_scloth=randperm(64);
for i=1:training
    a=tt_cork(i,1);
    train_cork(i*4-3:i*4,:)=cork_vec(a*4-3:a*4,:);
    b=tt_straw(i,1);
    train_straw(i*4-3:i*4,:)=straw_vec(b*4-3:b*4,:);
    c=tt_repskin(i,1);
    train_repskin(i*4-3:i*4,:)=repskin_vec(c*4-3:c*4,:);

```

```

d=tt_strcloth(i,1);
train_strcloth(i*4-3:i*4,:)=strcloth_vec(d*4-3:d*4,:);
e=tt_weave(i,1);
train_weave(i*4-3:i*4,:)=weave_vec(d*4-3:d*4,:);
f=tt_french(i,1);
train_french(i*4-3:i*4,:)=french_vec(d*4-3:d*4,:);
g=tt_net(i,1);
train_net(i*4-3:i*4,:)=net_vec(d*4-3:d*4,:);
h=tt_liz(i,1);
train_liz(i*4-3:i*4,:)=liz_vec(d*4-3:d*4,:);
j=tt_water(i,1);
train_water(i*4-3:i*4,:)=water_vec(d*4-3:d*4,:);
k=tt_scloth(i,1);
train_scloth(i*4-3:i*4,:)=scloth_vec(d*4-3:d*4,:);
end
for i=training+1:64
    a=tt_cork(i,1);
    test_cork(i*4-3:i*4,:)=cork_vec(a*4-3:a*4,:);
    b=tt_straw(i,1);
    test_straw(i*4-3:i*4,:)=straw_vec(b*4-3:b*4,:);
    c=tt_repskin(i,1);
    test_repskin(i*4-3:i*4,:)=repskin_vec(c*4-3:c*4,:);
    d=tt_strcloth(i,1);
    test_strcloth(i*4-3:i*4,:)=strcloth_vec(d*4-3:d*4,:);
    e=tt_weave(i,1);
    test_weave(i*4-3:i*4,:)=weave_vec(d*4-3:d*4,:);

```

```

f=tt_french(i,1);
test_french(i*4-3:i*4,:)=french_vec(d*4-3:d*4,:);

g=tt_net(i,1);
test_net(i*4-3:i*4,:)=net_vec(d*4-3:d*4,:);

h=tt_liz(i,1);
test_liz(i*4-3:i*4,:)=liz_vec(d*4-3:d*4,:);

j=tt_water(i,1);
test_water(i*4-3:i*4,:)=water_vec(d*4-3:d*4,:);

k=tt_scloth(i,1);
test_scloth(i*4-3:i*4,:)=scloth_vec(d*4-3:d*4,:);

end

test_cork(1:training*4,:)=[];
test_straw(1:training*4,:)=[];
test_repskin(1:training*4,:)=[];
test_strcloth(1:training*4,:)=[];
test_weave(1:training*4,:)=[];
test_french(1:training*4,:)=[];
test_net(1:training*4,:)=[];
test_liz(1:training*4,:)=[];
test_water(1:training*4,:)=[];
test_scloth(1:training*4,:)=[];

```

APPENDIX B2 WAVELET FUNCTION

% function for sample filtering wavelet decomposition

% second level discrete 2-D wavelet transform

function [vec]=wave_fcn_n(x); % enter the image matrix plus the level decomp.

[cA1,cH1,cV1,cD1] = dwt2(x,'db1'); % first level discrete 2-D wavelet transform

[cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'db1'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA2_H,cH2_H,cV2_H,cD2_H] = dwt2(cD1,'db1'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA1,cH1,cV1,cD1] = dwt2(x,'coif1'); % first level discrete 2-D wavelet transform

% [cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'coif1'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA1,cH1,cV1,cD1] = dwt2(x,'db2'); % first level discrete 2-D wavelet transform

% [cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'db2'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA1,cH1,cV1,cD1] = dwt2(x,'sym2'); % first level discrete 2-D wavelet transform

% [cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'sym2'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA1,cH1,cV1,cD1] = dwt2(x,'bior3.1'); % first level discrete 2-D wavelet transform

% [cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'bior3.1'); % second level discrete 2-D wavelet transform for the High-High coefficient

% [cA1,cH1,cV1,cD1] = dwt2(x,'rbio3.1'); % first level discrete 2-D wavelet transform

% [cA2_L,cH2_L,cV2_L,cD2_L] = dwt2(cA1,'rbio3.1'); % second level discrete 2-D wavelet transform for the High-High coefficient

```

% [cA1,cH1,cV1,cD1] = dwt2(x,'bior1.1'); % first level discrete 2-D wavelet transform
% [cA2,cH2,cV2,cD2] = dwt2(cD1,'bior1.1'); % second level discrete 2-D wavelet transform for the
High-High coefficient

max_cA2_L=max(max(abs(cA2_L)));
max_cH2_L=max(max(abs(cH2_L)));
max_cV2_L=max(max(abs(cV2_L)));
max_cD2_L=max(max(abs(cD2_L)));

% so we dont have NaN
if (max_cA2_L==0)
    max_cA2_L=1;
end
if (max_cH2_L==0)
    max_cH2_L=1;
end
if (max_cV2_L==0)
    max_cV2_L=1;
end
if (max_cD2_L==0)
    max_cD2_L=1;
end

cA2_L_m=cA2_L/max_cA2_L;
cH2_L_m=cH2_L/max_cH2_L;
cV2_L_m=cV2_L/max_cV2_L;
cD2_L_m=cD2_L/max_cD2_L;

```

```

[M,N]=size(cA2_L);

% The New Way normalized
cA2_L_c=im2col(cA2_L_m',[M N],'sliding');
cH2_L_c=im2col(cH2_L_m',[M N],'sliding');
cV2_L_c=im2col(cV2_L_m',[M N],'sliding');
cD2_L_c=im2col(cD2_L_m',[M N],'sliding');

%% The New Way!!!!!!not normalized
% cA2_L_c=im2col(cA2_L,[M N],'sliding');
% cH2_L_c=im2col(cH2_L,[M N],'sliding');
% cV2_L_c=im2col(cV2_L,[M N],'sliding');
% cD2_L_c=im2col(cD2_L,[M N],'sliding');

cF2=[cA2_L_c ; cH2_L_c ; cV2_L_c ; cD2_L_c];
% cF2=[cA2_L_c ; cD2_L_c];

mean_cF2=mean(cF2,2);
cov_cF2=cov(cF2',1);

[vec]=[mean_cF2 cov_cF2 cF2];

```

APPENDIX B3 WAVELET FILTERS

Here we present the filters used with their low-pass and high-pass decomposition values. Also we present their representation in graphics.

Note: LO_D, the decomposition low-pass filter

HI_D, the decomposition high-pass filter

1. Daubechies

'db1' → LO_D = [0.70711 0.70711]

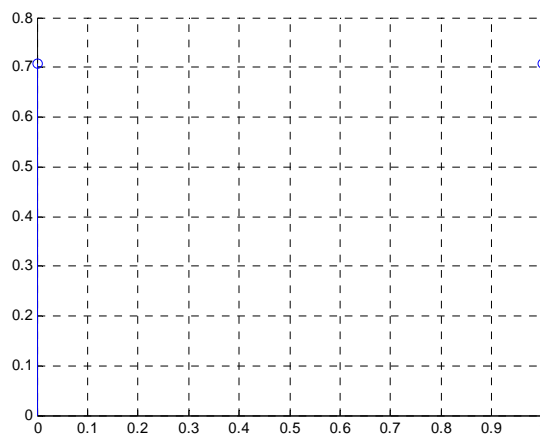
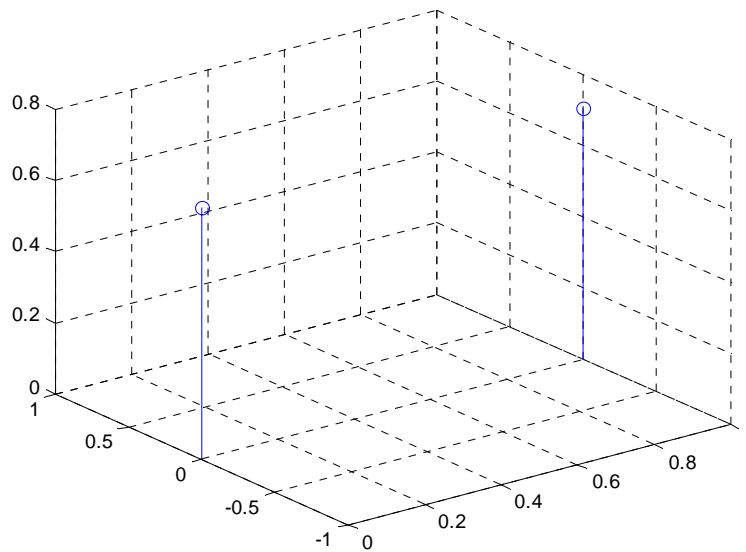


Figure 0.1 The decomposition low-pass filter for 'db1'

$$HI_D = [-0.70711 \quad 0.70711]$$

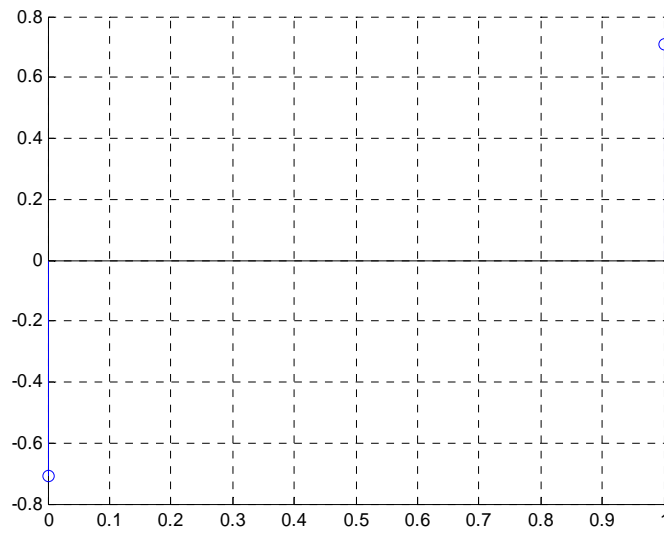
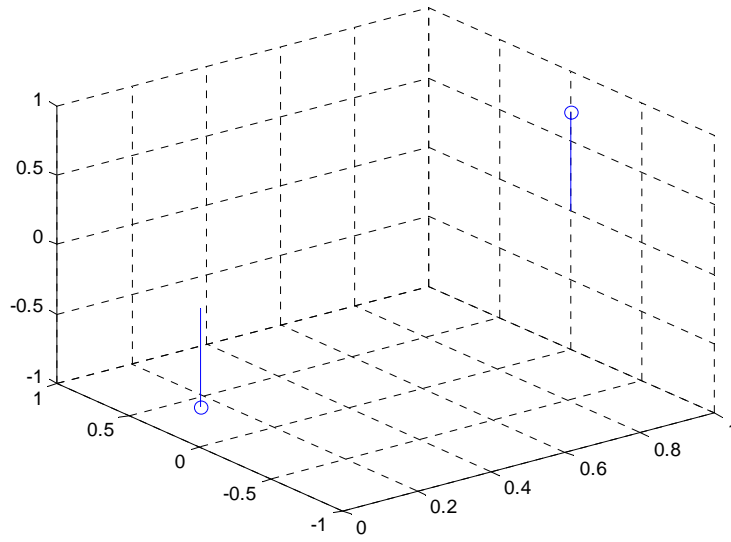


Figure 0.2 The decomposition high-pass filter for 'db1'

2. *Haar*:

'haar' \rightarrow LO_D = [0.70711 0.70711]

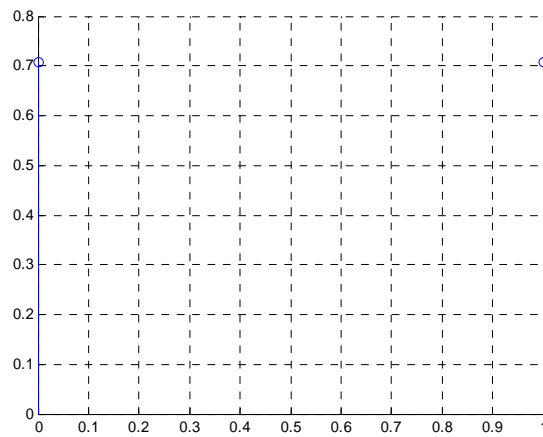
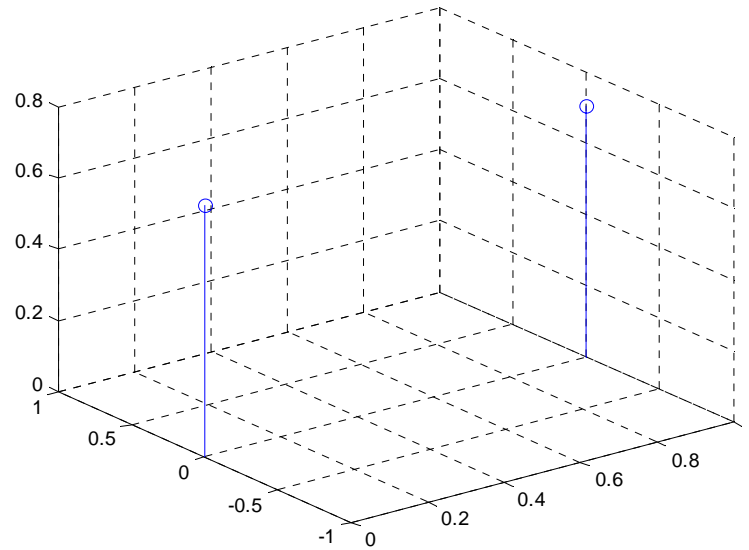


Figure 0.3 The decomposition low-pass filter for 'haar'

$$HI_D = [-0.70711 \quad 0.70711]$$

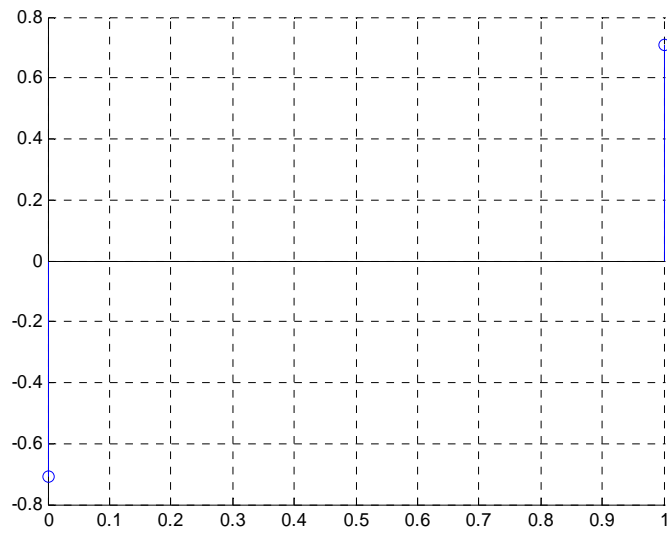
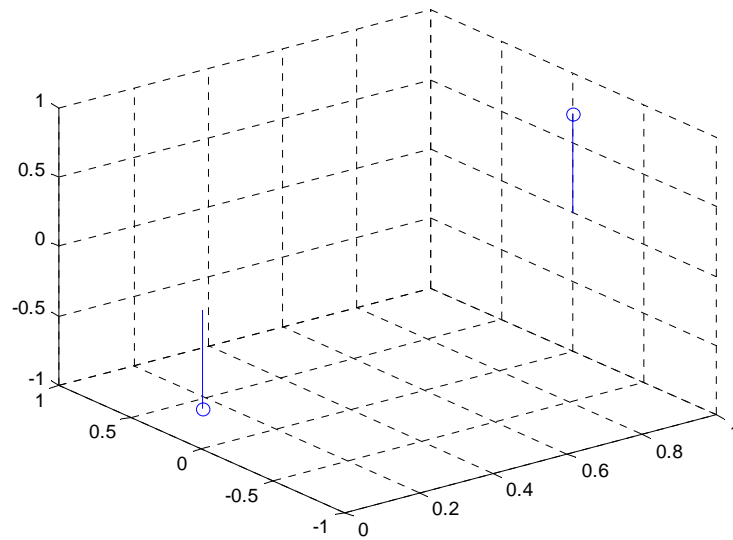


Figure 0.4 The decomposition high-pass filter for 'haar'

3. *Coiflets*:

'coif1' \rightarrow LO_D = [-0.015656 -0.072733 0.38486 0.85257 0.3379
-0.072733]

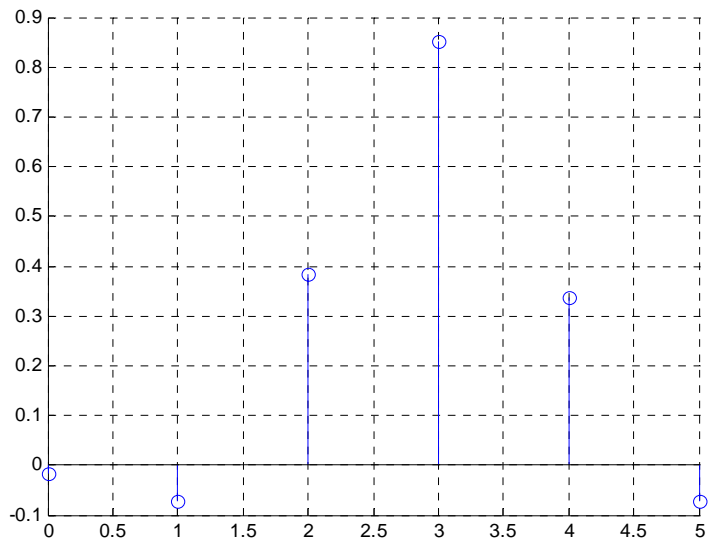
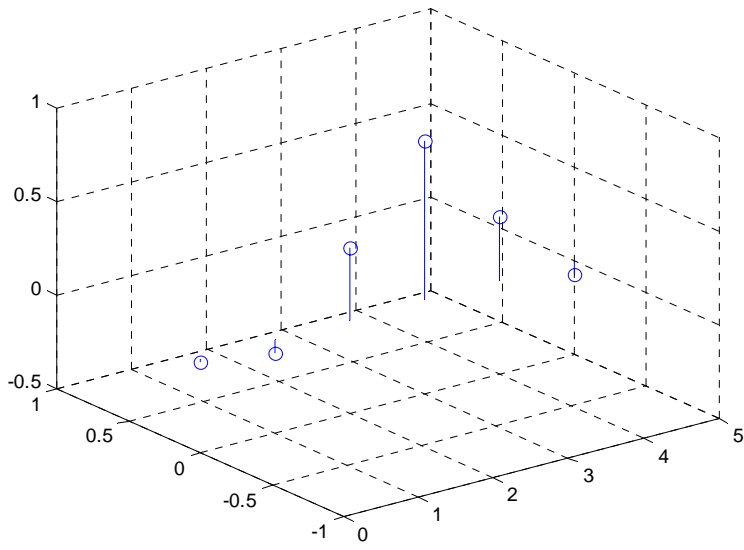


Figure 0.5 The decomposition low-pass filter for 'coif1'

HI_D = [0.072733 0.3379 -0.85257 0.38486 0.072733
-0.015656]

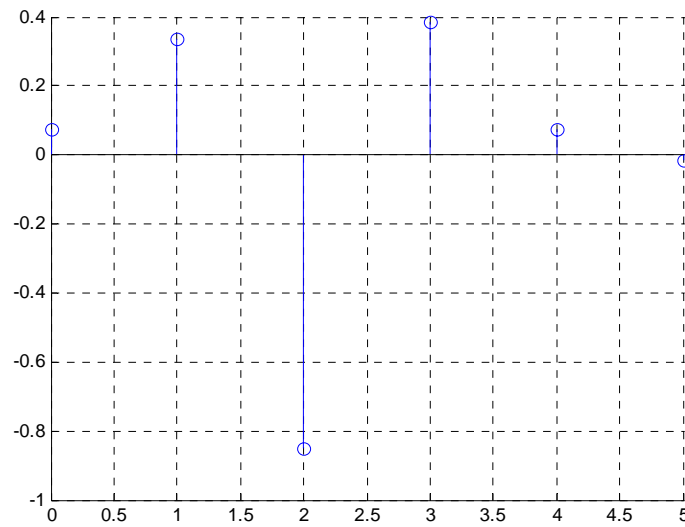
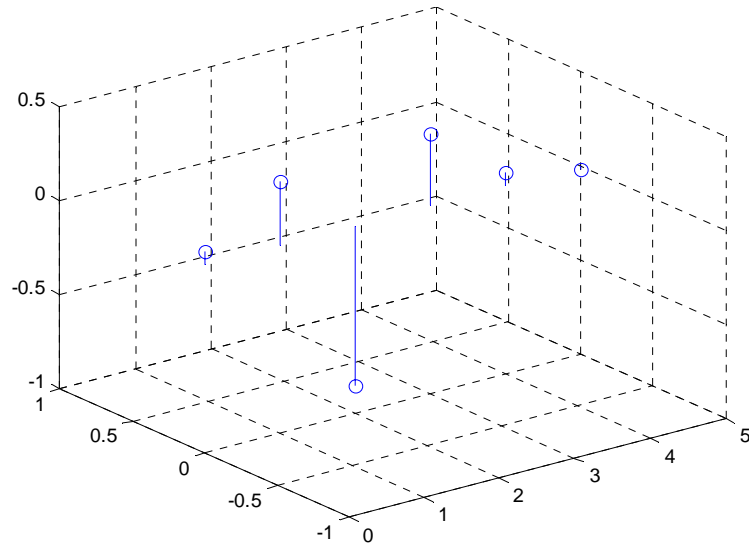


Figure 0.6 The decomposition high-pass filter for 'coif1'

4. Symlets:

'sym2' → LO_D = [-0.12941 0.22414 0.83652 0.48296]

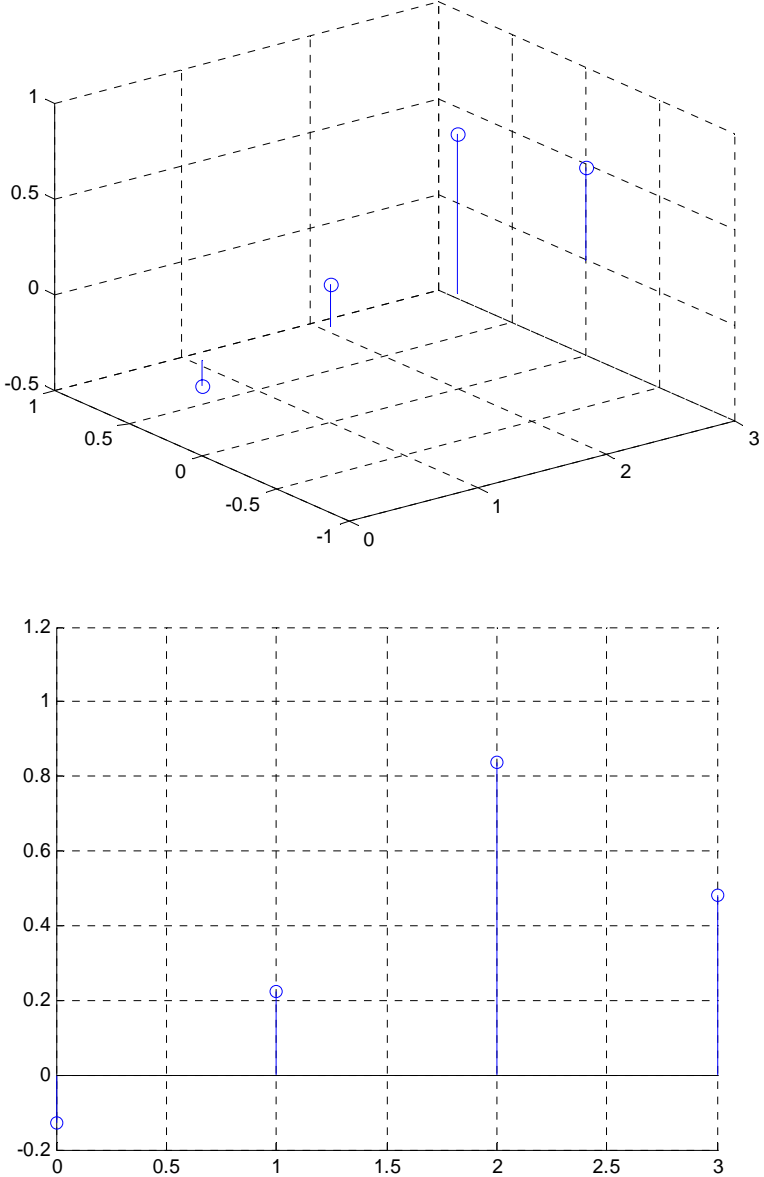


Figure 0.7 The decomposition low-pass filter for 'sym2'

$$HI_D = [-0.48296 \quad 0.83652 \quad -0.22414 \quad -0.12941]$$

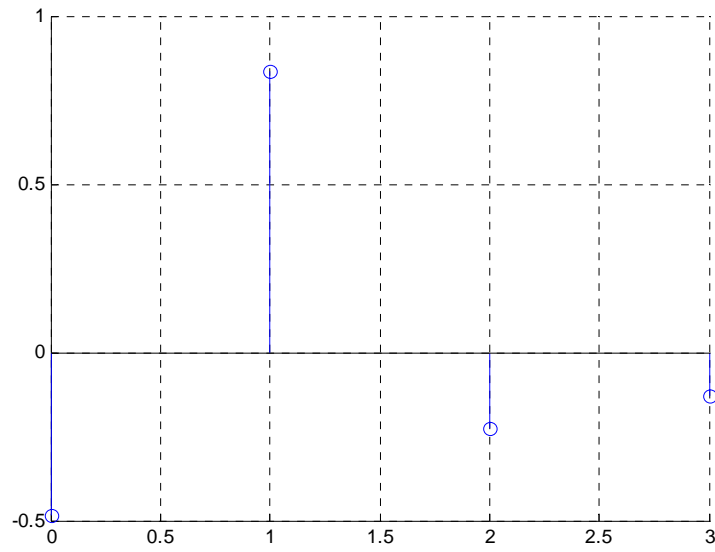
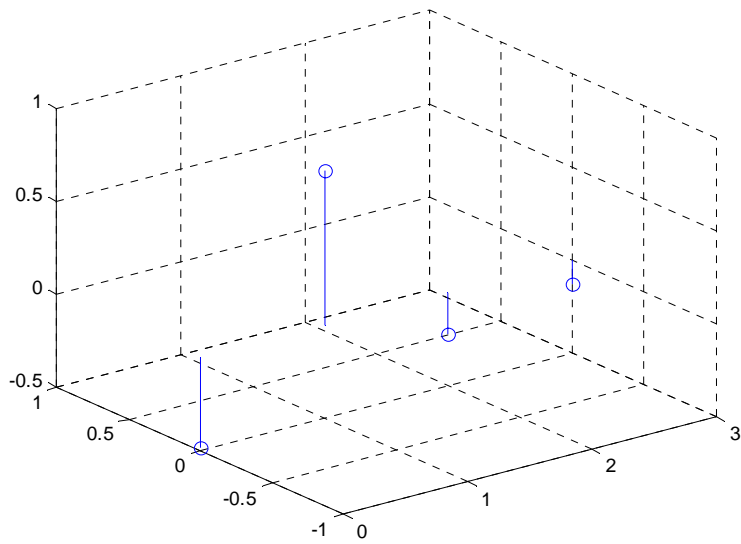


Figure 0.8 The decomposition high-pass filter for 'sym2'

5. Biorthogonal:

'bior1.1' \rightarrow LO_D = [0.70711 0.70711]

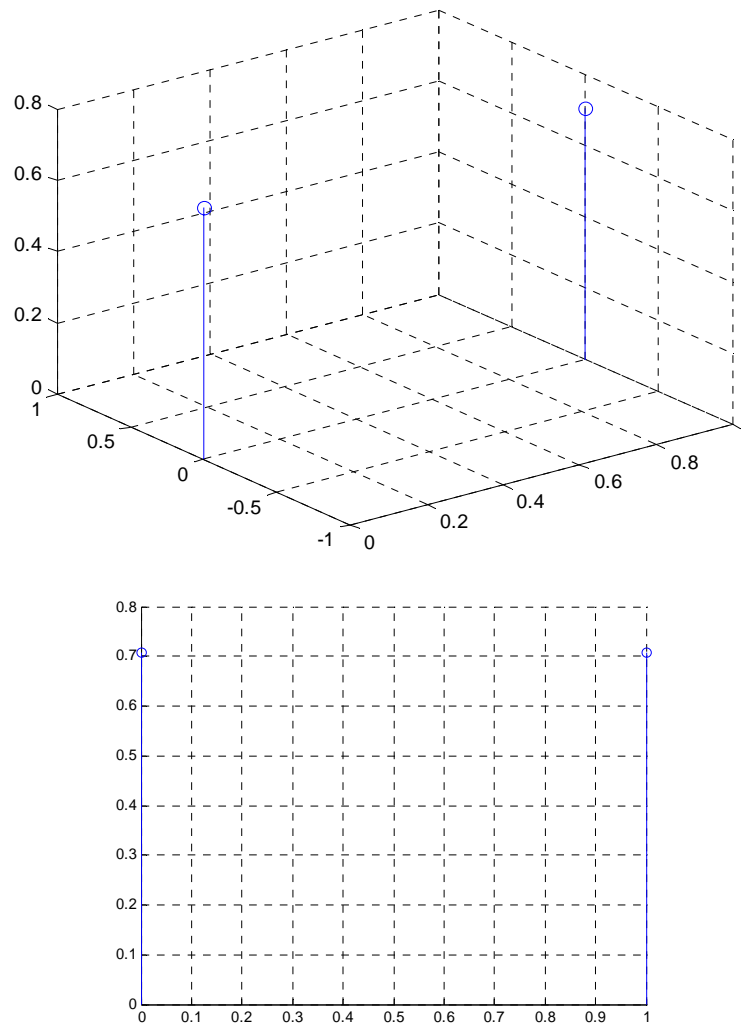


Figure 0.9 The decomposition low-pass filter for 'bior1.1'

$$HI_D = [-0.70711 \quad 0.70711]$$

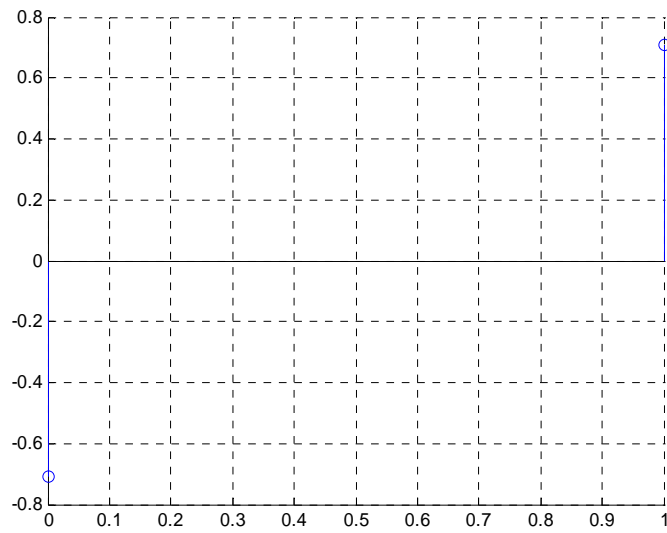
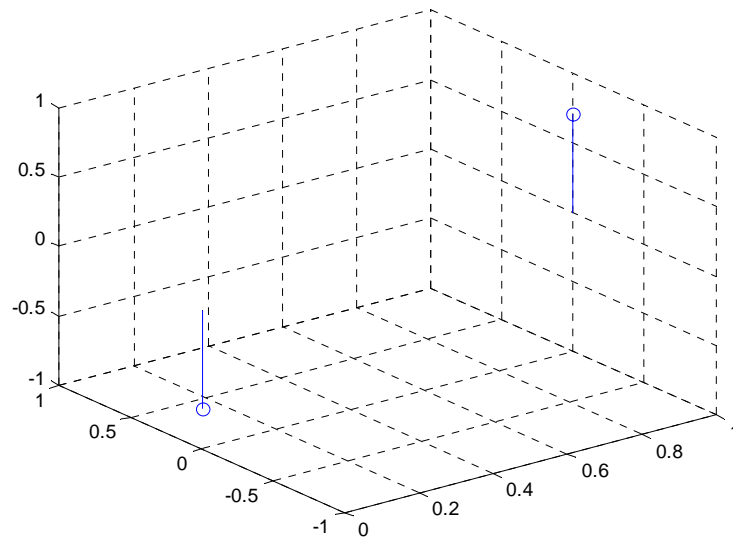


Figure 0.10 The decomposition high-pass filter for 'bior1.1'

6. Reverse Biorthogonal:

$$\text{'rbio1.1'} \rightarrow \text{LO_D} = [0.70711 \quad 0.70711]$$

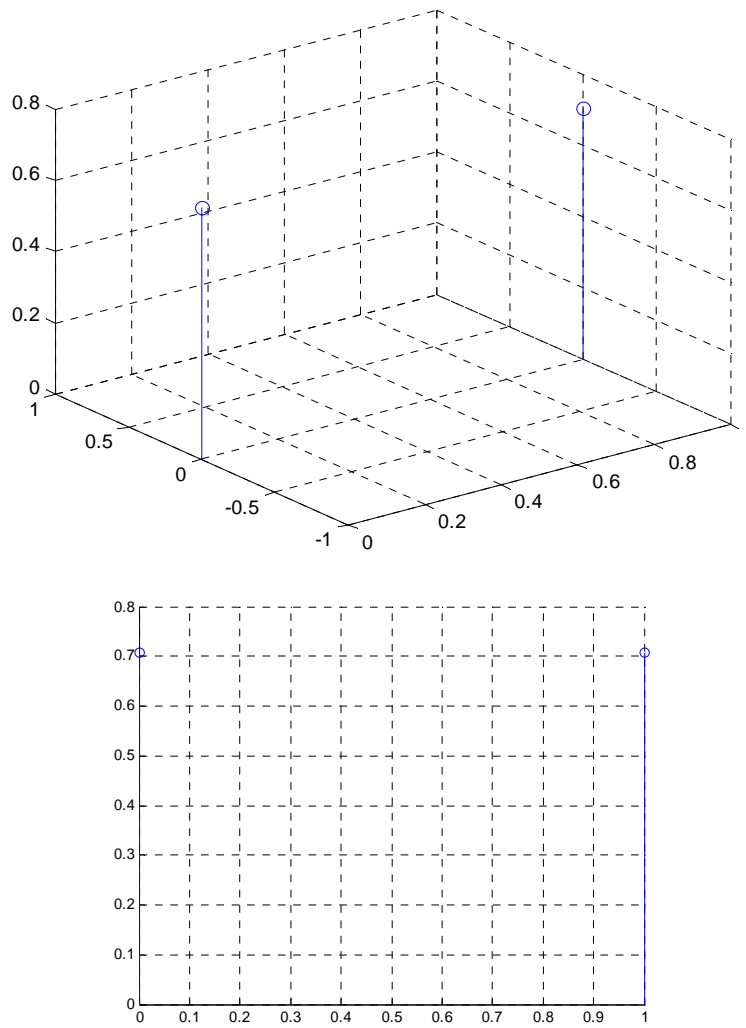


Figure 0.11 The decomposition low-pass filter for 'rbio1.1'

$$HI_D = [-0.70711 \quad 0.70711]$$

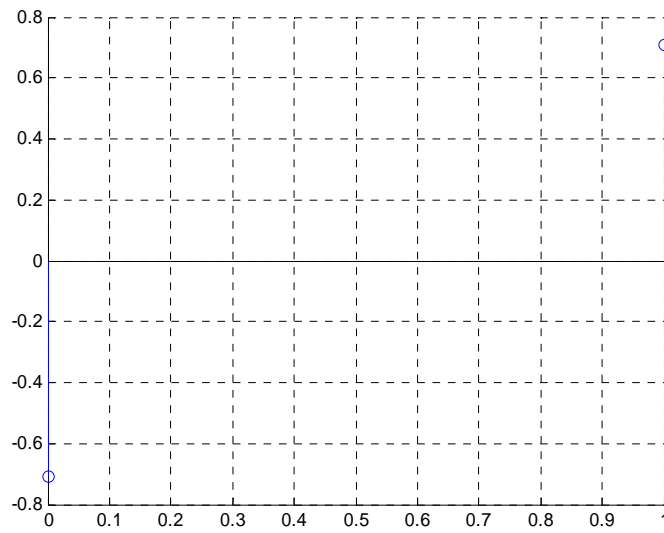
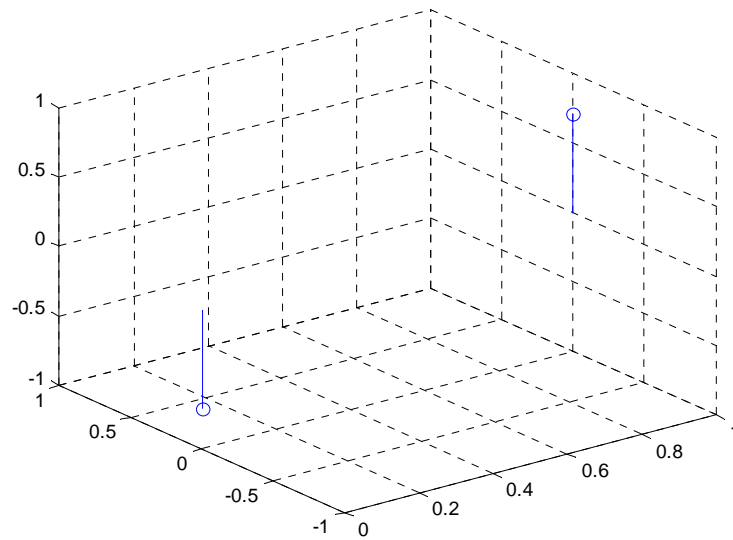


Figure 0.12 The decomposition high-pass filter for 'rbio1.1'

APPENDIX C CLASSIFIERS ALGORITHMS AND FUSION METHODS

APPENDIX C1 EUCLIDEAN DISTANCE CLASSIFIER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Euclidean Distance Classifier %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% All four Coeficient Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
tic
% loading the matrices of training & testing samples
load training_testing_rand_db1.mat
% load training_testing_rand_db1_hm.mat
% load training_testing_rand_haar.mat % using 'haar'
% load training_testing_rand_bior11.mat
% load training_testing_rand_rbio11.mat
[M N]=size(test_all);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X_DATA Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% By changing "x_data=..." we can classify any image can be training or
% testing, so this will wor for any type of data

% Discriminant function for x_data 01_data
for j=1:400 % 400 is the length of the testing samples matrix (for testing the classifier we will use 24
for the training samples)
    for i=1:24 % 24 is the length of the training samples matrix
```

```

for k=1:(N-5) % M by N-5 coefficient matrix
    x_data=test_all(j*4-3:j*4,5+k); % by changing x_data we will classify different images
    mean_01=train_cork(i*4-3:i*4,1);
    % cov_data1=train_cork(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
    eu_01(i,k)=eu_fcn(x_data,mean_01); % this saves for each coefficient matrix all the distances
between all the features of the other classes
    end
    eu_01=-sort(-eu_01); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x01(j*5-4:j*5,:)=eu_01(1:5,:); % So the minimum distance for the first coefficient matrix is the
first row
end
% Discriminant function for x_data 02_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_02=train_french(i*4-3:i*4,1);
            % cov_data1=train_repskin(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_02(i,k)=eu_fcn(x_data,mean_02); % this saves for each coefficient matrix all the distances
between all the features of the other classes
            end
            eu_02=-sort(-eu_02); % We first sort the columns in order to get the minimum distances for each
vector
            end
            eu_x02(j*5-4:j*5,:)=eu_02(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
        end
    end
end
% Discriminant function for x_data 03_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_03=train_liz(i*4-3:i*4,1);
            % cov_data1=train_straw(i*4-3:i*4,5:12); % if using ML classifier as the distance measure

```

```

        eu_03(i,k)=eu_fcn(x_data,mean_03); % this saves for each coefficient matrix all the distances
        between all the features of the other classes
    end
    eu_03=-sort(-eu_03); % We first sort the columns in order to get the minimum distances for each
    vector
    end
    eu_x03(j*5-4:j*5,:)=eu_03(1:5,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

```

% Discriminant function for x_data 04_data

```

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_04=train_net(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_04(i,k)=eu_fcn(x_data,mean_04); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_04=-sort(-eu_04); % We first sort the columns in order to get the minimum distances for each
        vector
        end
        eu_x04(j*5-4:j*5,:)=eu_04(1:5,:); % So the minimum distance for the first coefficient matrix is the first
        row
    end
end

```

```

% Discriminant function for x_data 05_data

```

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_05=train_repskin(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_05(i,k)=eu_fcn(x_data,mean_05); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
    end
end

```

```

    eu_05=-sort(-eu_05); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x05(j*5-4:j*5,:)=eu_05(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 06_data
for j=1:400    % 400 is the length of the testing samples matrix
    for i=1:24    % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_06=train_scloth(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_06(i,k)=eu_fcn(x_data,mean_06); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_06=-sort(-eu_06); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x06(j*5-4:j*5,:)=eu_06(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
    end

% Discriminant function for x_data 07_data
for j=1:400    % 400 is the length of the testing samples matrix
    for i=1:24    % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_07=train_straw(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_07(i,k)=eu_fcn(x_data,mean_07); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_07=-sort(-eu_07); % We first sort the columns in order to get the minimum distances for each
vector
        end
    end
end

```

```

    eu_x07(j*5-4:j*5,:)=eu_07(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

```

% Discriminant function for x_data 08_data

```

```

for j=1:400      % 400 is the length of the testing samples matrix
    for i=1:24   % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_08=train_strcloth(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_08(i,k)=eu_fcn(x_data,mean_08); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_08=-sort(-eu_08); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x08(j*5-4:j*5,:)=eu_08(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

```

% Discriminant function for x_data 09_data

```

```

for j=1:400      % 400 is the length of the testing samples matrix
    for i=1:24   % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_09=train_water(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_09(i,k)=eu_fcn(x_data,mean_09); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_09=-sort(-eu_09); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x09(j*5-4:j*5,:)=eu_09(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

```

% Discriminant function for x_data 10_data
for j=1:400    % 400 is the length of the testing samples matrix
    for i=1:24    % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_10=train_weave(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_10(i,k)=eu_fcn(x_data,mean_10); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_10=-sort(-eu_10); % We first sort the columns in order to get the minimum distances for each
        vector
    end
    eu_x10(j*5-4:j*5,)=eu_10(1:5,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

```

% Classification process %

```

```

count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

```

```

% Data classified

```

```

cork=1;
french=2;
liz=3;
net=4;

```

```

repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=1:40 % number of testing samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
            count_01=count_01+1;
            classif(j,:)=["cork 0"];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...

```

```

    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['scloth 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...

```

```

    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strecth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x01(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork liz'];
    else

```

```

        count_03=count_03+1;
        classif(j,:)=['liz cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;

```

```

        classif(j,:)=['cork straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork water'];
    else
        count_09=count_09+1;
        classif(j,:)=['water cork'];
    end
elseif (eu_x01(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork weave'];
    else
        count_10=count_10+1;
        classif(j,:)=['weave cork'];
    end
end
end
end

```

```

    results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Cork\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results cork/cork=%4.0f\n',count_01);
fprintf('Classification results cork/french=%4.0f\n',count_02);
fprintf('Classification results cork/liz=%4.0f\n',count_03);
fprintf('Classification results cork/net=%4.0f\n',count_04);
fprintf('Classification results cork/repskin=%4.0f\n',count_05);
fprintf('Classification results cork/scloth=%4.0f\n',count_06);
fprintf('Classification results cork/straw=%4.0f\n',count_07);
fprintf('Classification results cork/strcloth=%4.0f\n',count_08);
fprintf('Classification results cork/water=%4.0f\n',count_09);
fprintf('Classification results cork/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(1,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,1)=count_01*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

```

```

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclotch=6;
straw=7;
strelotch=8;
water=9;
weave=10;
random_selection=11;

% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=41:80 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...

```

```

    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['scloth 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...

```

```

    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x02(i*5-4,j)==eu_x01(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=['french cork'];
    else
        count_01=count_01+1;
        classif(j,:)=['cork french'];
    end
end

```

```

elseif (eu_x02(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french liz];
    else
        count_03=count_03+1;
        classif(j,:)=[liz french];
    end
elseif (eu_x02(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french net];
    else
        count_04=count_04+1;
        classif(j,:)=[net french];
    end
elseif (eu_x02(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french repskin];
    else
        count_05=count_05+1;
        classif(j,:)=[repskin french];
    end
elseif (eu_x02(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french scloth];
    else
        count_06=count_06+1;

```

```

        classif(j,:)=['sclotch french'];
    end
elseif (eu_x02(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=['french straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw french'];
    end
elseif (eu_x02(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=['french strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth french'];
    end
elseif (eu_x02(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=['french water'];
    else
        count_09=count_09+1;
        classif(j,:)=['water french'];
    end
elseif (eu_x02(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=['french weave'];

```

```

else
    count_10=count_10+1;
    classif(j,:)=['weave french'];
end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for French\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results french/cork=%4.0f\n',count_01);
fprintf('Classification results french/french=%4.0f\n',count_02);
fprintf('Classification results french/liz=%4.0f\n',count_03);
fprintf('Classification results french/net=%4.0f\n',count_04);
fprintf('Classification results french/repskin=%4.0f\n',count_05);
fprintf('Classification results french/scloth=%4.0f\n',count_06);
fprintf('Classification results french/straw=%4.0f\n',count_07);
fprintf('Classification results french/streloth=%4.0f\n',count_08);
fprintf('Classification results french/water=%4.0f\n',count_09);
fprintf('Classification results french/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(2,:)=['count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10'];
perc(1,2)=count_02*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,2));
%%%%%%%%%%
%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;

```

```

count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

```

```

% Data classified

```

```

cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;

```

```

random_selection=11;

```

```

% the only thing that is going to change according to what we are

```

```

% classifying is the first element of the array:

```

```

% i.e. classif(j,:)=["cork" cork]; only change the part between " "

```

```

% Depending on which class is going to be classified we must change the

```

```

% first "if" below, instead of eu_x01 be compared with every single eu_x we

```

```

% must change it for example if is french class it should be eu_02 instead

```

```

% of eu_01.

```

```

for i=81:120 % number of training samples

```

```

    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified

```

```

        if ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))

```

```

count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
count_02=count_02+1;
classif(j,:)=['french 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
&&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...

```

```

        &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
        &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
        &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
        &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
        &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
        &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
        &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
        &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
        &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
        &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
        &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
        &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
        &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
        &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
        &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
        &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
        &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x03(i,j)==eu_x02(i,j))
    random_pick=rand;        % We use rand to let random selection when
    if random_pick>.5

```

```

        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french liz];
    end
elseif (eu_x03(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    end
elseif (eu_x03(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net liz];
    end
elseif (eu_x03(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin liz];
    end
elseif (eu_x03(i,j)==eu_x06(i,j))

```

```

random_pick=rand;
if random_pick>.5
    count_03=count_03+1;
    classif(i,j*2-1:j*2)=[liz scloth];
else
    count_06=count_06+1;
    classif(i,j*2-1:j*2)=[scloth liz];
end
elseif (eu_x03(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw liz];
    end
elseif (eu_x03(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth liz];
    end
elseif (eu_x03(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water liz];
    end

```

```

        end
elseif (eu_x03(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave liz];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Liz\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results liz/cork=%4.0f\n',count_01);
fprintf('Classification results liz/french=%4.0f\n',count_02);
fprintf('Classification results liz/liz=%4.0f\n',count_03);
fprintf('Classification results liz/net=%4.0f\n',count_04);
fprintf('Classification results liz/repskin=%4.0f\n',count_05);
fprintf('Classification results liz/scloth=%4.0f\n',count_06);
fprintf('Classification results liz/straw=%4.0f\n',count_07);
fprintf('Classification results liz/strcloth=%4.0f\n',count_08);
fprintf('Classification results liz/water=%4.0f\n',count_09);
fprintf('Classification results liz/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(3,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,3)=count_03*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,3));
%%%%%%%%%%
%%%%%%%%%%

```

```

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=121:160 % number of training samples

```

```

for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
if ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
count_02=count_02+1;
classif(j,:)=['french 0'];
elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))

```

```

count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...

```

```

        &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=[weave 0];
elseif (eu_x04(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net french];
    else
        count_02=count_02+1;
        classif(j,:)=[french net];
    end
elseif (eu_x04(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net liz];
    else
        count_03=count_03+1;
        classif(j,:)=[liz net];
    end
elseif (eu_x04(i*5-4,j)==eu_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork net];
    end
elseif (eu_x04(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;

```

```

        classif(j,:)=['net repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin net'];
    end
elseif (eu_x04(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=['net scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth net'];
    end
elseif (eu_x04(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=['net straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw net'];
    end
elseif (eu_x04(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=['net strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth net'];
    end
elseif (eu_x04(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;

```

```

    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net water];
    else
        count_09=count_09+1;
        classif(j,:)=[water net];
    end
elseif (eu_x04(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave net];
    end
end
end
end
    results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Net\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results net/cork=%4.0f\n',count_01);
fprintf('Classification results net/french=%4.0f\n',count_02);
fprintf('Classification results net/liz=%4.0f\n',count_03);
fprintf('Classification results net/net=%4.0f\n',count_04);
fprintf('Classification results net/repskin=%4.0f\n',count_05);
fprintf('Classification results net/scloth=%4.0f\n',count_06);
fprintf('Classification results net/straw=%4.0f\n',count_07);
fprintf('Classification results net/strcloth=%4.0f\n',count_08);
fprintf('Classification results net/water=%4.0f\n',count_09);
fprintf('Classification results net/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
0;

```

```

fprintf('\nTotal samples=%4.0f\n',a);
rec(4,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,4)=count_04*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,4));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

```

```

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=161:200 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
            count_05=count_05+1;
            classif(j,:)=['repskin 0'];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=['french 0'];
        elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=['liz 0'];
        elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
            count_04=count_04+1;

```

```

classif(j,:)= [net 0];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)= [cork 0];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)= [scloth 0];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)= [straw 0];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)= [strcloth 0];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
&&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))

```

```

count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
&&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x05(i,j)==eu_x02(i,j))
random_pick=rand;      % We use rand to let random selection when
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin french];
else
count_02=count_02+1;
classif(i,j*2-1:j*2)=[french repskin];
end
elseif (eu_x05(i,j)==eu_x03(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin liz];
else
count_03=count_03+1;
classif(i,j*2-1:j*2)=[liz repskin];
end
elseif (eu_x05(i,j)==eu_x04(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin net];
else
count_04=count_04+1;

```

```

        classif(i,j*2-1:j*2)=[net repskin];
    end
elseif (eu_x05(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    end
elseif (eu_x05(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth repskin];
    end
elseif (eu_x05(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw repskin];
    end
elseif (eu_x05(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin stcloth];

```

```

else
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strcloth repskin];
end
elseif (eu_x05(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water repskin];
    end
elseif (eu_x05(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave repskin];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Repskin\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results repskin/cork=%4.0f\n',count_01);
fprintf('Classification results repskin/french=%4.0f\n',count_02);
fprintf('Classification results repskin/liz=%4.0f\n',count_03);
fprintf('Classification results repskin/net=%4.0f\n',count_04);
fprintf('Classification results repskin/repskin=%4.0f\n',count_05);
fprintf('Classification results repskin/scloth=%4.0f\n',count_06);

```

```

fprintf('Classification results repskin/straw=%4.0f\n',count_07);
fprintf('Classification results repskin/strcloth=%4.0f\n',count_08);
fprintf('Classification results repskin/water=%4.0f\n',count_09);
fprintf('Classification results repskin/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(5,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,5)=count_05*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,5));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;

```

```

water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=201:240 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
            count_06=count_06+1;
            classif(j,:)=["cork" 0];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french" 0];
        elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=["liz" 0];

```

```

elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;

```

```

    classif(j,:)=['strecloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
    count_10=count_10+1;
    classif(j,:)=['weave 0'];
elseif (eu_x06(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french scloth'];
    end
elseif (eu_x06(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz scloth'];
    end
elseif (eu_x06(i*5-4,j)==eu_x04(i*5-4,j))

```

```

random_pick=rand;
if random_pick>.5
    count_06=count_06+1;
    classif(j,:)=[scloth net];
else
    count_04=count_04+1;
    classif(j,:)=[net scloth];
end
elseif (eu_x06(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth repskin];
    else
        count_05=count_05+1;
        classif(j,:)=[repskin scloth];
    end
elseif (eu_x06(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork scloth];
    end
elseif (eu_x06(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw scloth];
    end

```

```

    end
elseif (eu_x06(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth strcloth];
    else
        count_08=count_08+1;
        classif(j,:)=[strcloth scloth];
    end
elseif (eu_x06(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth water];
    else
        count_09=count_09+1;
        classif(j,:)=[water scloth];
    end
elseif (eu_x06(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave scloth];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf("\n\nResults for Approximation Coefficients of Testing Samples for Scloth\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');

```

```

fprintf('Classification results scloth/cork=%4.0f\n',count_01);
fprintf('Classification results scloth/french=%4.0f\n',count_02);
fprintf('Classification results scloth/liz=%4.0f\n',count_03);
fprintf('Classification results scloth/net=%4.0f\n',count_04);
fprintf('Classification results scloth/repskin=%4.0f\n',count_05);
fprintf('Classification results scloth/scloth=%4.0f\n',count_06);
fprintf('Classification results scloth/straw=%4.0f\n',count_07);
fprintf('Classification results scloth/strcloth=%4.0f\n',count_08);
fprintf('Classification results scloth/water=%4.0f\n',count_09);
fprintf('Classification results scloth/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(6,:)= [count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,6)=count_06*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,6));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;

```

```

liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=241:280 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
            count_07=count_07+1;
            classif(j,:)=["straw 0"];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...

```

```

    &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['scloth 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];

```

```

elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x07(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;

```

```

        classif(j,:)=['straw liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x01(i*5-4,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_07=count_07+1;
    classif(j,:)=['straw cork'];
else
    count_01=count_01+1;
    classif(j,:)=['cork straw'];
end
elseif (eu_x07(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw water'];
    else
        count_09=count_09+1;
        classif(j,:)=['water straw'];
    end
elseif (eu_x07(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw weave'];
    else
        count_10=count_10+1;
        classif(j,:)=['weave straw'];
    end
end

```

```

    end
end
    results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Straw\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results straw/cork=%4.0f\n',count_01);
fprintf('Classification results straw/french=%4.0f\n',count_02);
fprintf('Classification results straw/liz=%4.0f\n',count_03);
fprintf('Classification results straw/net=%4.0f\n',count_04);
fprintf('Classification results straw/repskin=%4.0f\n',count_05);
fprintf('Classification results straw/scloth=%4.0f\n',count_06);
fprintf('Classification results straw/straw=%4.0f\n',count_07);
fprintf('Classification results straw/strcloth=%4.0f\n',count_08);
fprintf('Classification results straw/water=%4.0f\n',count_09);
fprintf('Classification results straw/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(7,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,7)=count_07*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,7));
%%%%%%%%%%
%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;

```

```

count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=281:320 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
            count_08=count_08+1;
            classif(j,:)=["strcloth 0"];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...

```

```

    &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
count_02=count_02+1;
classif(j,:)=['french 0'];
elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...

```

```

    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x08(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strecth french'];
    else
        count_02=count_02+1;

```

```

        classif(j,:)=['french strcloth'];
    end
elseif (eu_x08(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz strcloth'];
    end
elseif (eu_x08(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net strcloth'];
    end
elseif (eu_x08(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin strcloth'];
    end
elseif (eu_x08(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth scloth'];

```

```

else
    count_06=count_06+1;
    classif(j,:)=[scloth strcloth];
end
elseif (eu_x08(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw strcloth];
    end
elseif (eu_x08(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork strcloth];
    end
elseif (eu_x08(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth water];
    else
        count_09=count_09+1;
        classif(j,:)=[water strcloth];
    end
elseif (eu_x08(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5

```

```

        count_08=count_08+1;
        classif(j,:)=[strcloth weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave strcloth];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Strcloth\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results strcloth/cork=%4.0f\n',count_01);
fprintf('Classification results strcloth/french=%4.0f\n',count_02);
fprintf('Classification results strcloth/liz=%4.0f\n',count_03);
fprintf('Classification results strcloth/net=%4.0f\n',count_04);
fprintf('Classification results strcloth/repskin=%4.0f\n',count_05);
fprintf('Classification results strcloth/scloth=%4.0f\n',count_06);
fprintf('Classification results strcloth/straw=%4.0f\n',count_07);
fprintf('Classification results strcloth/strcloth=%4.0f\n',count_08);
fprintf('Classification results strcloth/water=%4.0f\n',count_09);
fprintf('Classification results strcloth/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(8,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,8)=count_08*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,8));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;

```

```

count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;

% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=321:360 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...

```

```

    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
count_02=count_02+1;
classif(j,:)=['french 0'];
elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...

```

```

    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];

```

```

elseif (eu_x09(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water french];
    else
        count_02=count_02+1;
        classif(j,:)=[french water];
    end
elseif (eu_x09(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water liz];
    else
        count_03=count_03+1;
        classif(j,:)=[liz water];
    end
elseif (eu_x09(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water net];
    else
        count_04=count_04+1;
        classif(j,:)=[net water];
    end
elseif (eu_x09(i*5-4,j)==eu_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water repskin];
    else
        count_05=count_05+1;

```

```

        classif(j,:)=['repskin water'];
    end
elseif (eu_x09(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth water'];
    end
elseif (eu_x09(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw water'];
    end
elseif (eu_x09(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth water'];
    end
elseif (eu_x09(i*5-4,j)==eu_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water cork'];

```

```

else
    count_01=count_01+1;
    classif(j,:)=['cork water'];
end
elseif (eu_x09(i*5-4,j)==eu_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water weave'];
    else
        count_10=count_10+1;
        classif(j,:)=['weave water'];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Water\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results water/cork=%4.0f\n',count_01);
fprintf('Classification results water/french=%4.0f\n',count_02);
fprintf('Classification results water/liz=%4.0f\n',count_03);
fprintf('Classification results water/net=%4.0f\n',count_04);
fprintf('Classification results water/repskin=%4.0f\n',count_05);
fprintf('Classification results water/scloth=%4.0f\n',count_06);
fprintf('Classification results water/straw=%4.0f\n',count_07);
fprintf('Classification results water/strcloth=%4.0f\n',count_08);
fprintf('Classification results water/water=%4.0f\n',count_09);
fprintf('Classification results water/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(9,:)=['count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10'];

```

```

perc(1,9)=count_09*100/a;
fprintf('\nPercentage of classification=%4.4f\n',perc(1,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;

% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we

```

```

% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for i=361:400 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((eu_x10(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x10(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x10(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x10(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x10(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x10(i*5-4,j)>eu_x01(i*5-4,j)))
            count_10=count_10+1;
            classif(j,:)=['weave 0'];
        elseif ((eu_x02(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x02(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x02(i*5-4,j)>eu_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=['french 0'];
        elseif ((eu_x03(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x01(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x03(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x03(i*5-4,j)>eu_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=['liz 0'];
        elseif ((eu_x04(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x05(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x07(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x04(i*5-4,j)>eu_x09(i*5-4,j))...
            &&(eu_x04(i*5-4,j)>eu_x10(i*5-4,j)))
            count_04=count_04+1;
            classif(j,:)=['net 0'];
        elseif ((eu_x05(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x03(i*5-4,j))...
            &&(eu_x05(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x01(i*5-4,j))...

```

```

    &&(eu_x05(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x05(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x05(i*5-4,j)>eu_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((eu_x06(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x06(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x06(i*5-4,j)>eu_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclcloth 0'];
elseif ((eu_x07(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x07(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x07(i*5-4,j)>eu_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((eu_x08(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x01(i*5-4,j))&&(eu_x08(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x08(i*5-4,j)>eu_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strelcloth 0'];
elseif ((eu_x09(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x03(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x09(i*5-4,j)>eu_x01(i*5-4,j))...
    &&(eu_x09(i*5-4,j)>eu_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((eu_x01(i*5-4,j)>eu_x02(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x03(i*5-4,j))...

```

```

    &&(eu_x01(i*5-4,j)>eu_x04(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x05(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x06(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x07(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x08(i*5-4,j))&&(eu_x01(i*5-4,j)>eu_x09(i*5-4,j))...
    &&(eu_x01(i*5-4,j)>eu_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif (eu_x10(i*5-4,j)==eu_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_02=count_02+1;
        classif(j,:)=['french weave'];
    end
elseif (eu_x10(i*5-4,j)==eu_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz weave'];
    end
elseif (eu_x10(i*5-4,j)==eu_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_04=count_04+1;
        classif(j,:)=['net weave'];
    end
elseif (eu_x10(i*5-4,j)==eu_x05(i*5-4,j))

```

```

random_pick=rand;
if random_pick>.5
    count_10=count_10+1;
    classif(j,:)=[weave random_selection];
else
    count_05=count_05+1;
    classif(j,:)=[repskin weave];
end
elseif (eu_x10(i*5-4,j)==eu_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_06=count_06+1;
        classif(j,:)=[scloth weave];
    end
elseif (eu_x10(i*5-4,j)==eu_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_07=count_07+1;
        classif(j,:)=[straw weave];
    end
elseif (eu_x10(i*5-4,j)==eu_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_08=count_08+1;
        classif(j,:)=[strecloth weave];
    end

```

```

    end
elseif (eu_x10(i*5-4,j)==eu_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_09=count_09+1;
        classif(j,:)=[water weave];
    end
elseif (eu_x10(i*5-4,j)==eu_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_01=count_01+1;
        classif(j,:)=[cork weave];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Weave\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results weave/cork=%4.0f\n',count_01);
fprintf('Classification results weave/french=%4.0f\n',count_02);
fprintf('Classification results weave/liz=%4.0f\n',count_03);
fprintf('Classification results weave/net=%4.0f\n',count_04);
fprintf('Classification results weave/repskin=%4.0f\n',count_05);
fprintf('Classification results weave/scloth=%4.0f\n',count_06);
fprintf('Classification results weave/straw=%4.0f\n',count_07);
fprintf('Classification results weave/strcloth=%4.0f\n',count_08);
fprintf('Classification results weave/water=%4.0f\n',count_09);

```

```

fprintf('Classification results weave/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(10,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,10)=count_10*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,10));
rec_perc=[rec perc'];
toc

```

APPENDIX C2 EUCLIDEAN DISTANCE FUNCTION

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Discriminant Function For Multivariate & Multiple Classes %

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% This function calculates the discriminant functions and th threshold using the data matrix and the mean
function ei_x=eu_fcn(x_data0,mean_data0) % x_data ~ is a vector from the matrix of coefficients
% mean_data ~ is the mean vector

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% This part is used for proof %

```

```

% load train_cork.mat %

```

```

% x_data0=train_cork(1:8,37); %

```

```

% mean_data0=train_cork(1:8,2); %

```

```

% cov_data0=train_cork(1:8,21:28); %

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% euclidean distance classification

```

```

% Example of how this works
% But we will use the general case multivariate and multiple classes
%  $e_i(X) = -(1/2)(X - \mu_i)'(X - \mu_i)$ 
%  $e_i(X) \geq e_j(X)$  (for all  $j \rightarrow X$  element of class  $i$ )

% difference of discriminant functions
ei_x = -0.5*(x_data0 - mean_data0)'*(x_data0 - mean_data0);

```

APPENDIX C2 K-NEAREST NEIGHBOR K=3 CLASSIFIER

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% K-Nearest Neighbor K=3 Classifier %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% All four Coefficient Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% KNN K=3 %

clear
tic
% loading the matrices of training & testing samples
load training_testing_rand_db1.mat
% load training_testing_rand_haar.mat
% load training_testing_rand_bior11.mat
% load training_testing_rand_rbio11.mat

[M N] = size(test_all);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X_DATA Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% By changing "x_data=..." we can classify any image can be training or
% testing, so this will wor for any type of data

% Discriminant function for x_data 01_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix (for testing the classifier we will use 24
for the training samples)
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k); % by changing x_data we will classify different images
            mean_01=train_cork(i*4-3:i*4,1);
            % cov_data1=train_cork(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_01(i,k)=eu_fcn(x_data,mean_01); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_01=-sort(-eu_01); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x01(j*3-2:j*3,:)=eu_01(1:3,:); % So the minimum distance for the first coefficient matrix is the
first row
end
% Discriminant function for x_data 02_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_02=train_french(i*4-3:i*4,1);
            % cov_data1=train_repskin(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_02(i,k)=eu_fcn(x_data,mean_02); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_02=-sort(-eu_02); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x02(j*3-2:j*3,:)=eu_02(1:3,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

```

% Discriminant function for x_data 03_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_03=train_liz(i*4-3:i*4,1);
            % cov_data1=train_straw(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_03(i,k)=eu_fcn(x_data,mean_03); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_03=-sort(-eu_03); % We first sort the columns in order to get the minimum distances for each
        vector
    end
    eu_x03(j*3-2:j*3,)=eu_03(1:3,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

```

% Discriminant function for x_data 04_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_04=train_net(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_04(i,k)=eu_fcn(x_data,mean_04); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_04=-sort(-eu_04); % We first sort the columns in order to get the minimum distances for each
        vector
    end
    eu_x04(j*3-2:j*3,)=eu_04(1:3,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

```

% Discriminant function for x_data 05_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix

```

```

for k=1:(N-5) % M by N-5 coefficient matrix
    x_data=test_all(j*4-3:j*4,5+k);
    mean_05=train_repskin(i*4-3:i*4,1);
    % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
    eu_05(i,k)=eu_fcn(x_data,mean_05); % this saves for each coefficient matrix all the distances
    between all the features of the other classes
end
eu_05=-sort(-eu_05); % We first sort the columns in order to get the minimum distances for each
vector
end
eu_x05(j*3-2:j*3,:)=eu_05(1:3,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

% Discriminant function for x_data 06_data

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_06=train_scloth(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_06(i,k)=eu_fcn(x_data,mean_06); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_06=-sort(-eu_06); % We first sort the columns in order to get the minimum distances for each
        vector
    end
    eu_x06(j*3-2:j*3,:)=eu_06(1:3,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

% Discriminant function for x_data 07_data

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_07=train_straw(i*4-3:i*4,1);

```

```

        % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
        eu_07(i,k)=eu_fcn(x_data,mean_07); % this saves for each coefficient matrix all the distances
        between all the features of the other classes
    end
    eu_07=-sort(-eu_07); % We first sort the columns in order to get the minimum distances for each
    vector
    end
    eu_x07(j*3-2:j*3,)=eu_07(1:3,:); % So the minimum distance for the first coefficient matrix is the first
    row
end

```

```

% Discriminant function for x_data 08_data

```

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_08=train_strcloth(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_08(i,k)=eu_fcn(x_data,mean_08); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
        eu_08=-sort(-eu_08); % We first sort the columns in order to get the minimum distances for each
        vector
        end
        eu_x08(j*3-2:j*3,)=eu_08(1:3,:); % So the minimum distance for the first coefficient matrix is the first
        row
    end
end

```

```

% Discriminant function for x_data 09_data

```

```

for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_09=train_water(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_09(i,k)=eu_fcn(x_data,mean_09); % this saves for each coefficient matrix all the distances
            between all the features of the other classes
        end
    end
end

```

```

    end
    eu_09=-sort(-eu_09); % We first sort the columns in order to get the minimum distances for each
vector
    end
    eu_x09(j*3-2:j*3,:)=eu_09(1:3,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 10_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_10=train_weave(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_10(i,k)=eu_fcn(x_data,mean_10); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_10=-sort(-eu_10); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x10(j*3-2:j*3,:)=eu_10(1:3,:); % So the minimum distance for the first coefficient matrix is the first
row
    end

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;

```

```

count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.

```

```

for j=1:(N-5) % number of training samples
for i=1:120 % number of distance calculated to be classified 5 for each 24 vectors equals 120
count_times=count_times+1; % thi is to count ech 5 times to classify
if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork 0];
elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
    &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
    &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
    &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
    &&(eu_x02(i,j)>eu_x10(i,j)))
count_02=count_02+1;
classif(i,j*2-1:j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1:j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1:j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...

```

```

    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1;j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...

```

```

    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_01=count_01+1;
    classif(i,j*2-1:j*2)=[cork repskin];
else
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin cork];
end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
end

```

```

elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))

```

```

count_k02=count_k02+1;
results(i_count,1:2)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
&&(count_03>count_04)&&(count_03>count_05)...
&&(count_03>count_06)&&(count_03>count_07)...
&&(count_03>count_08)&&(count_03>count_09)...
&&(count_03>count_10))
count_k03=count_k03+1;
results(i_count,1:2)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
&&(count_04>count_01)&&(count_04>count_05)...
&&(count_04>count_06)&&(count_04>count_07)...
&&(count_04>count_08)&&(count_04>count_09)...
&&(count_04>count_10))
count_k04=count_k04+1;
results(i_count,1:2)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
&&(count_05>count_04)&&(count_05>count_01)...
&&(count_05>count_06)&&(count_05>count_07)...
&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,1:2)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...
&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;
results(i_count,1:2)=[sclloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...

```

```

    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,1:2)=[strecloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,1:2)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,1:2)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,1:2)=[french cork];
    end
elseif (count_01==count_03)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,1:2)=[cork liz];
else
    count_k03=count_k03+1;
    results(i_count,1:2)=[liz cork];
end
elseif (count_01==count_04)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,1:2)=[cork net];
else
    count_k04=count_k04+1;
    results(i_count,1:2)=[net cork];
end
elseif (count_01==count_05)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,1:2)=[cork repskin];
else
    count_k05=count_k05+1;
    results(i_count,1:2)=[repskin cork];
end
elseif (count_01==count_06)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,1:2)=[cork scloth];
else
    count_k06=count_k06+1;
    results(i_count,1:2)=[scloth cork];

```

```

end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork strecth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strecth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,1:2)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,1:2)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;

```

```

        results(i_count,1:2)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,1:2)=[french weave];
else
    count_k10=count_k10+1;
    results(i_count,1:2)=[weave french];
end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,1:2)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth liz];
    end
end

```

```

elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz weave];
    else
        count_k10=count_k10+1;

```

```

        results(i_count,1:2)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net strcloth];

```

```

else
    count_k08=count_k08+1;
    results(i_count,1:2)=[strcloth net];
end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[sclotch repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin strecth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strecth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave repskin];
    end
elseif (count_06==count_07)

```

```

random_pick=rand;
if random_pick>.5
    count_k06=count_k06+1;
    results(i_count,1:2)=[scloth straw];
else
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw scloth];
end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[stcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave scloth];
    end

```

```

end
elseif (count_07==count_08)
random_pick=rand;
if random_pick>.5
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw strcloth];
else
    count_k08=count_k08+1;
    results(i_count,1:2)=[strcloth straw];
end
elseif (count_07==count_09)
random_pick=rand;
if random_pick>.5
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw water];
else
    count_k09=count_k09+1;
    results(i_count,1:2)=[water straw];
end
elseif (count_07==count_10)
random_pick=rand;
if random_pick>.5
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw weave];
else
    count_k10=count_k10+1;
    results(i_count,1:2)=[weave straw];
end
elseif (count_08==count_09)
random_pick=rand;
if random_pick>.5
    count_k08=count_k08+1;
    results(i_count,1:2)=[strcloth water];
else

```

```

        count_k09=count_k09+1;
        results(i_count,1:2)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,1:2)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave water];
    end
end
end
end
end
    % results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Cork\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results cork/cork=%4.0f\n',count_k01);
fprintf('Classification results cork/french=%4.0f\n',count_k02);
fprintf('Classification results cork/liz=%4.0f\n',count_k03);
fprintf('Classification results cork/net=%4.0f\n',count_k04);
fprintf('Classification results cork/repskin=%4.0f\n',count_k05);
fprintf('Classification results cork/scloth=%4.0f\n',count_k06);

```

```

fprintf('Classification results cork/straw=%4.0f\n',count_k07);
fprintf('Classification results cork/strcloth=%4.0f\n',count_k08);
fprintf('Classification results cork/water=%4.0f\n',count_k09);
fprintf('Classification results cork/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(1,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,1)=count_k01*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;

```

```

count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=121:240 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[sclotch 0];

```

```

elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x02(i,j)==eu_x01(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    else

```

```

        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    end
elseif (eu_x02(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz french];
    end
elseif (eu_x02(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net french];
    end
elseif (eu_x02(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin french];
    end
elseif (eu_x02(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;

```

```

        classif(i,j*2-1:j*2)=[french scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth french];
    end
elseif (eu_x02(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw french];
    end
elseif (eu_x02(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth french];
    end
elseif (eu_x02(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water french];
    end
elseif (eu_x02(i,j)==eu_x10(i,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_02=count_02+1;
    classif(i,j*2-1:j*2)=[french weave];
else
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave french];
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,3:4)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...

```

```

    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,3:4)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,3:4)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,3:4)=[sclloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,3:4)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,3:4)=[streloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...

```

```

    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,3:4)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,3:4)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,3:4)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k01=count_k01+1;
        results(i_count,3:4)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,3:4)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw cork];
    end
elseif (count_01==count_08)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,3:4)=[cork strcloth];
else
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth cork];
end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz french];
    end

```

```

end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,3:4)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[sclotch french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french straw];
    else

```

```

        count_k07=count_k07+1;
        results(i_count,3:4)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;

```

```

        results(i_count,3:4)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,3:4)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;

```

```

if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,3:4)=[liz strcloth];
else
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth liz];
end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin net];
    end
end

```

```

elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net water];
    else
        count_k09=count_k09+1;

```

```

        results(i_count,3:4)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin stcloth];

```

```

else
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth repskin];
end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth straw];
    end
elseif (count_07==count_09)

```

```

random_pick=rand;
if random_pick>.5
    count_k07=count_k07+1;
    results(i_count,3:4)=[straw water];
else
    count_k09=count_k09+1;
    results(i_count,3:4)=[water straw];
end
elseif (count_07==count_10)
    random_pick=rand;
if random_pick>.5
    count_k07=count_k07+1;
    results(i_count,3:4)=[straw weave];
else
    count_k10=count_k10+1;
    results(i_count,3:4)=[weave straw];
end
elseif (count_08==count_09)
    random_pick=rand;
if random_pick>.5
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth water];
else
    count_k09=count_k09+1;
    results(i_count,3:4)=[water strcloth];
end
elseif (count_08==count_10)
    random_pick=rand;
if random_pick>.5
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth weave];
else
    count_k10=count_k10+1;
    results(i_count,3:4)=[weave strcloth];

```

```

        end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,3:4)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave water];
    end
end
end
end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for French\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results french/cork=%4.0f\n',count_k01);
fprintf('Classification results french/french=%4.0f\n',count_k02);
fprintf('Classification results french/liz=%4.0f\n',count_k03);
fprintf('Classification results french/net=%4.0f\n',count_k04);
fprintf('Classification results french/repskin=%4.0f\n',count_k05);
fprintf('Classification results french/scloth=%4.0f\n',count_k06);
fprintf('Classification results french/straw=%4.0f\n',count_k07);
fprintf('Classification results french/streloth=%4.0f\n',count_k08);
fprintf('Classification results french/water=%4.0f\n',count_k09);
fprintf('Classification results french/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(2,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,2)=count_k02*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,2));

```

%%%%%%%%
%%%%%%%%

% Classification process %

count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;

% Data classified

cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;

```

weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=241:360 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))

```

```

count_01=count_01+1;
classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
&&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
&&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
&&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
&&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
&&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
&&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
&&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
&&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
&&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
&&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
&&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
&&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
&&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
&&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
&&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
&&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
&&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
&&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
&&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...

```

```

        &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
        &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
        &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
        &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
        &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
        &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
        &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
        &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
        &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x03(i,j)==eu_x02(i,j))
    random_pick=rand;        % We use rand to let random selection when
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french liz];
    end
elseif (eu_x03(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];

```

```

end
elseif (eu_x03(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net liz];
    end
elseif (eu_x03(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin liz];
    end
elseif (eu_x03(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth liz];
    end
elseif (eu_x03(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz straw];
    else

```

```

        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw liz];
    end
elseif (eu_x03(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth liz];
    end
elseif (eu_x03(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water liz];
    end
elseif (eu_x03(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave liz];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;

```

```

if ((count_01>count_02)&&(count_01>count_03)...
    &&(count_01>count_04)&&(count_01>count_05)...
    &&(count_01>count_06)&&(count_01>count_07)...
    &&(count_01>count_08)&&(count_01>count_09)...
    &&(count_01>count_10))
    count_k01=count_k01+1;
    results(i_count,5:6)=[cork 0];
elseif ((count_02>count_01)&&(count_02>count_03)...
    &&(count_02>count_04)&&(count_02>count_05)...
    &&(count_02>count_06)&&(count_02>count_07)...
    &&(count_02>count_08)&&(count_02>count_09)...
    &&(count_02>count_10))
    count_k02=count_k02+1;
    results(i_count,5:6)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
    &&(count_03>count_04)&&(count_03>count_05)...
    &&(count_03>count_06)&&(count_03>count_07)...
    &&(count_03>count_08)&&(count_03>count_09)...
    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,5:6)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,5:6)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;

```

```

    results(i_count,5:6)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,5:6)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,5:6)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,5:6)=[strelcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,5:6)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))

```

```

count_k10=count_k10+1;
results(i_count,5:6)=[weave 0];
elseif (count_01==count_02)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,5:6)=[cork french];
else
count_k02=count_k02+1;
results(i_count,5:6)=[french cork];
end
elseif (count_01==count_03)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,5:6)=[cork liz];
else
count_k03=count_k03+1;
results(i_count,5:6)=[liz cork];
end
elseif (count_01==count_04)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,5:6)=[cork net];
else
count_k04=count_k04+1;
results(i_count,5:6)=[net cork];
end
elseif (count_01==count_05)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,5:6)=[cork repskin];

```

```

else
    count_k05=count_k05+1;
    results(i_count,5:6)=[repskin cork];
end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[stcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k01=count_k01+1;
        results(i_count,5:6)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,5:6)=[net french];
    end
elseif (count_02==count_05)

```

```

random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,5:6)=[french repskin];
else
    count_k05=count_k05+1;
    results(i_count,5:6)=[repskin french];
end
elseif (count_02==count_06)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,5:6)=[french scloth];
else
    count_k06=count_k06+1;
    results(i_count,5:6)=[scloth french];
end
elseif (count_02==count_07)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,5:6)=[french straw];
else
    count_k07=count_k07+1;
    results(i_count,5:6)=[straw french];
end
elseif (count_02==count_08)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,5:6)=[french strcloth];
else
    count_k08=count_k08+1;
    results(i_count,5:6)=[strcloth french];

```

```

end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,5:6)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz repskin];
    else

```

```

        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;

```

```

        results(i_count,5:6)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;

```

```

if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,5:6)=[net straw];
else
    count_k07=count_k07+1;
    results(i_count,5:6)=[straw net];
end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net strecth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strecth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave net];
    end
end

```

```

elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[stcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin water];
    else
        count_k09=count_k09+1;

```

```

        results(i_count,5:6)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[stcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth water];
    end

```

```

else
    count_k09=count_k09+1;
    results(i_count,5:6)=[water scloth];
end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k07=count_k07+1;
        results(i_count,5:6)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,5:6)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave water];
    end
end
end

```

```

    end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Liz\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results liz/cork=%4.0f\n',count_k01);
fprintf('Classification results liz/french=%4.0f\n',count_k02);
fprintf('Classification results liz/liz=%4.0f\n',count_k03);
fprintf('Classification results liz/net=%4.0f\n',count_k04);
fprintf('Classification results liz/repskin=%4.0f\n',count_k05);
fprintf('Classification results liz/scloth=%4.0f\n',count_k06);
fprintf('Classification results liz/straw=%4.0f\n',count_k07);
fprintf('Classification results liz/strcloth=%4.0f\n',count_k08);
fprintf('Classification results liz/water=%4.0f\n',count_k09);
fprintf('Classification results liz/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(3,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,3)=count_k03*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,3));
%%%%%%%%%%
%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;

```

```

count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead

```

```

% of eu_01.
for j=1:(N-5)% number of training samples
  for i=361:480 % number of vectors for each coefficient matrix to be classified
    count_times=count_times+1; % thi is to count ech 5 times to classify
    if ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
        &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
        &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
        &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
        &&(eu_x04(i,j)>eu_x10(i,j)))
      count_04=count_04+1;
      classif(i,j*2-1:j*2)=[net 0];
    elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
        &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
        &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
        &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
        &&(eu_x02(i,j)>eu_x10(i,j)))
      count_02=count_02+1;
      classif(i,j*2-1:j*2)=[french 0];
    elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
        &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
        &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
        &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
        &&(eu_x03(i,j)>eu_x10(i,j)))
      count_03=count_03+1;
      classif(i,j*2-1:j*2)=[liz 0];
    elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
        &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
        &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
        &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
        &&(eu_x01(i,j)>eu_x10(i,j)))
      count_01=count_01+1;
      classif(i,j*2-1:j*2)=[cork 0];
    elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
        &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...

```

```

    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1;j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x04(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french net];
    end
elseif (eu_x04(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz net];
    end
elseif (eu_x04(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    end
elseif (eu_x04(i,j)==eu_x05(i,j))

```

```

random_pick=rand;
if random_pick>.5
    count_04=count_04+1;
    classif(i,j*2-1:j*2)=[net repskin];
else
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin net];
end
elseif (eu_x04(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth net];
    end
elseif (eu_x04(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw net];
    end
elseif (eu_x04(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth net];
    end

```

```

end
elseif (eu_x04(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water net];
    end
elseif (eu_x04(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave net];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...

```

```

    &&(count_02>count_10))
    count_k02=count_k02+1;
    results(i_count,7:8)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
    &&(count_03>count_04)&&(count_03>count_05)...
    &&(count_03>count_06)&&(count_03>count_07)...
    &&(count_03>count_08)&&(count_03>count_09)...
    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,7:8)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,7:8)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,7:8)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,7:8)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...

```

```

    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,7:8)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,7:8)=[strelcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,7:8)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,7:8)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,7:8)=[french cork];
    end
end

```

```

elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,7:8)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork scloth];
    else
        count_k06=count_k06+1;

```

```

        results(i_count,7:8)=[sclotch cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork strotch];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strotch cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork weave];

```

```

else
    count_k10=count_k10+1;
    results(i_count,7:8)=[weave cork];
end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,7:8)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k02=count_k02+1;
        results(i_count,7:8)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water french];
    end
elseif (count_02==count_10)

```

```

random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,7:8)=[french weave];
else
    count_k10=count_k10+1;
    results(i_count,7:8)=[weave french];
end
elseif (count_03==count_04)
    random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,7:8)=[liz net];
else
    count_k04=count_k04+1;
    results(i_count,7:8)=[net liz];
end
elseif (count_03==count_05)
    random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,7:8)=[liz repskin];
else
    count_k05=count_k05+1;
    results(i_count,7:8)=[repskin liz];
end
elseif (count_03==count_06)
    random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,7:8)=[liz scloth];
else
    count_k06=count_k06+1;
    results(i_count,7:8)=[scloth liz];

```

```

end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,7:8)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;

```

```

        results(i_count,7:8)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;

```

```

if random_pick>.5
    count_k05=count_k05+1;
    results(i_count,7:8)=[repskin straw];
else
    count_k07=count_k07+1;
    results(i_count,7:8)=[straw repskin];
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin strecth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strecth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave repskin];
    end
end

```

```

elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth strecloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strecloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth weave];
    else
        count_k10=count_k10+1;

```

```

        results(i_count,7:8)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth water];
    end
end
end

```

```

else
    count_k09=count_k09+1;
    results(i_count,7:8)=[water strcloth];
end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,7:8)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave water];
    end
end
end
end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Net\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results net/cork=%4.0f\n',count_k01);
fprintf('Classification results net/french=%4.0f\n',count_k02);
fprintf('Classification results net/liz=%4.0f\n',count_k03);
fprintf('Classification results net/net=%4.0f\n',count_k04);
fprintf('Classification results net/repskin=%4.0f\n',count_k05);

```

```

fprintf('Classification results net/scloth=%4.0f\n',count_k06);
fprintf('Classification results net/straw=%4.0f\n',count_k07);
fprintf('Classification results net/streloth=%4.0f\n',count_k08);
fprintf('Classification results net/water=%4.0f\n',count_k09);
fprintf('Classification results net/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(4,:)=[count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,4)=count_k04*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,4));
%%%%%%%%%%
%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;

```

```

count_k09=0;
count_k10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
    for i=481:600 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
            &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
            &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
            &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
            &&(eu_x05(i,j)>eu_x10(i,j)))
            count_05=count_05+1;
        end
    end
end

```

```

    classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
    &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
    &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
    &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
    &&(eu_x02(i,j)>eu_x10(i,j)))
    count_02=count_02+1;
    classif(i,j*2-1;j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
    count_03=count_03+1;
    classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
    count_04=count_04+1;
    classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
    count_01=count_01+1;
    classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))

```

```

count_06=count_06+1;
classif(i,j*2-1:j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
&&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
&&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
&&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
&&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
&&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
&&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
&&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
&&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
&&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
&&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
&&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
&&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
&&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
&&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
&&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
&&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x05(i,j)==eu_x02(i,j))
random_pick=rand;      % We use rand to let random selection when
if random_pick>.5
count_05=count_05+1;

```

```

        classif(i,j*2-1:j*2)=[repskin french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french repskin];
    end
elseif (eu_x05(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz repskin];
    end
elseif (eu_x05(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net repskin];
    end
elseif (eu_x05(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    end
elseif (eu_x05(i,j)==eu_x06(i,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin scloth];
else
    count_06=count_06+1;
    classif(i,j*2-1:j*2)=[scloth repskin];
end
elseif (eu_x05(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw repskin];
    end
elseif (eu_x05(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth repskin];
    end
elseif (eu_x05(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water repskin];
    end
end

```

```

elseif (eu_x05(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave repskin];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,9:10)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz 0];
    end
end

```

```

elseif((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,9:10)=[net 0];
elseif((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,9:10)=[repskin 0];
elseif((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth 0];
elseif((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,9:10)=[straw 0];
elseif((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;

```

```

    results(i_count,9:10)=[strelth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,9:10)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,9:10)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,9:10)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz cork];
    end
elseif (count_01==count_04)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,9:10)=[cork net];
else
    count_k04=count_k04+1;
    results(i_count,9:10)=[net cork];
end
elseif (count_01==count_05)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,9:10)=[cork repskin];
else
    count_k05=count_k05+1;
    results(i_count,9:10)=[repskin cork];
end
elseif (count_01==count_06)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,9:10)=[cork scloth];
else
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth cork];
end
elseif (count_01==count_07)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,9:10)=[cork straw];
else
    count_k07=count_k07+1;
    results(i_count,9:10)=[straw cork];

```

```

end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french liz];
    else

```

```

        count_k03=count_k03+1;
        results(i_count,9:10)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,9:10)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;

```

```

        results(i_count,9:10)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;

```

```

if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,9:10)=[liz net];
else
    count_k04=count_k04+1;
    results(i_count,9:10)=[net liz];
end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw liz];
    end
end

```

```

elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net repskin];
    else
        count_k05=count_k05+1;

```

```

        results(i_count,9:10)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net strecloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strecloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net water];

```

```

else
    count_k09=count_k09+1;
    results(i_count,9:10)=[water net];
end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw scloth];
    end
elseif (count_06==count_08)

```

```

random_pick=rand;
if random_pick>.5
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth strcloth];
else
    count_k08=count_k08+1;
    results(i_count,9:10)=[strcloth scloth];
end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth straw];
    end

```

```

end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,9:10)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,9:10)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave water];
    end
end
end
end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Repskin\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results repskin/cork=%4.0f\n',count_k01);
fprintf('Classification results repskin/french=%4.0f\n',count_k02);
fprintf('Classification results repskin/liz=%4.0f\n',count_k03);
fprintf('Classification results repskin/net=%4.0f\n',count_k04);
fprintf('Classification results repskin/repskin=%4.0f\n',count_k05);
fprintf('Classification results repskin/scloth=%4.0f\n',count_k06);
fprintf('Classification results repskin/straw=%4.0f\n',count_k07);
fprintf('Classification results repskin/strcloth=%4.0f\n',count_k08);
fprintf('Classification results repskin/water=%4.0f\n',count_k09);
fprintf('Classification results repskin/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(5,:)= [count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];

```

```
perc(1,5)=count_k05*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,5));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Classification process %
```

```
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;
```

```
count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
```

```
% Data classified
```

```
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
```

```

straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
    for i=601:720 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
            &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
            &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
            &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
            &&(eu_x06(i,j)>eu_x10(i,j)))
            count_06=count_06+1;
            classif(i,j*2-1:j*2)=[scloth 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...

```

```

    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x06(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[sclotch french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french sclotch];
    end
elseif (eu_x06(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[sclotch liz];

```

```

else
    count_03=count_03+1;
    classif(i,j*2-1:j*2)=[liz scloth];
end
elseif (eu_x06(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net scloth];
    end
elseif (eu_x06(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin scloth];
    end
elseif (eu_x06(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    end
elseif (eu_x06(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5

```

```

        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw scloth];
    end
elseif (eu_x06(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth scloth];
    end
elseif (eu_x06(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water scloth];
    end
elseif (eu_x06(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave scloth];
    end
end
end

```

```

if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,11:12)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;
        results(i_count,11:12)=[net 0];
    elseif ((count_05>count_02)&&(count_05>count_03)...
        &&(count_05>count_04)&&(count_05>count_01)...
        &&(count_05>count_06)&&(count_05>count_07)...

```

```

&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,11:12)=[repskin 0];
elseif((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...
&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;
results(i_count,11:12)=[scloth 0];
elseif((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...
&&(count_07>count_10))
count_k07=count_k07+1;
results(i_count,11:12)=[straw 0];
elseif((count_08>count_02)&&(count_08>count_03)...
&&(count_08>count_04)&&(count_08>count_05)...
&&(count_08>count_06)&&(count_08>count_07)...
&&(count_08>count_01)&&(count_08>count_09)...
&&(count_08>count_10))
count_k08=count_k08+1;
results(i_count,11:12)=[strcloth 0];
elseif((count_09>count_02)&&(count_09>count_03)...
&&(count_09>count_04)&&(count_09>count_05)...
&&(count_09>count_06)&&(count_09>count_07)...
&&(count_09>count_08)&&(count_09>count_01)...
&&(count_09>count_10))
count_k09=count_k09+1;
results(i_count,11:12)=[water 0];
elseif((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...

```

```

&&(count_10>count_06)&&(count_10>count_07)...
&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,11:12)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,11:12)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,11:12)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;

```

```

if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,11:12)=[cork repskin];
else
    count_k05=count_k05+1;
    results(i_count,11:12)=[repskin cork];
end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth cork];
    end
end

```

```

elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french net];
    else
        count_k04=count_k04+1;

```

```

        results(i_count,11:12)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french stcloth];

```

```

else
    count_k08=count_k08+1;
    results(i_count,11:12)=[strcloth french];
end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,11:12)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k03=count_k03+1;
        results(i_count,11:12)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth liz];
    end
elseif (count_03==count_09)

```

```

random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,11:12)=[liz water];
else
    count_k09=count_k09+1;
    results(i_count,11:12)=[water liz];
end
elseif(count_03==count_10)
    random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,11:12)=[liz weave];
else
    count_k10=count_k10+1;
    results(i_count,11:12)=[weave liz];
end
elseif(count_04==count_05)
    random_pick=rand;
if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,11:12)=[net repskin];
else
    count_k05=count_k05+1;
    results(i_count,11:12)=[repskin net];
end
elseif(count_04==count_06)
    random_pick=rand;
if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,11:12)=[net scloth];
else
    count_k06=count_k06+1;
    results(i_count,11:12)=[scloth net];

```

```

end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,11:12)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;

```

```

        results(i_count,11:12)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;

```

```

if random_pick>.5
    count_k06=count_k06+1;
    results(i_count,11:12)=[scloth water];
else
    count_k09=count_k09+1;
    results(i_count,11:12)=[water scloth];
end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water straw];
    end
end

```

```

elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,11:12)=[water weave];
    else
        count_k10=count_k10+1;

```

```

        results(i_count,11:12)=[weave water];
    end
end
end
end
    % results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Scloth\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results scloth/cork=%4.0f\n',count_k01);
fprintf('Classification results scloth/french=%4.0f\n',count_k02);
fprintf('Classification results scloth/liz=%4.0f\n',count_k03);
fprintf('Classification results scloth/net=%4.0f\n',count_k04);
fprintf('Classification results scloth/repskin=%4.0f\n',count_k05);
fprintf('Classification results scloth/scloth=%4.0f\n',count_k06);
fprintf('Classification results scloth/straw=%4.0f\n',count_k07);
fprintf('Classification results scloth/strcloth=%4.0f\n',count_k08);
fprintf('Classification results scloth/water=%4.0f\n',count_k09);
fprintf('Classification results scloth/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(6,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08
count_k09 count_k10];
perc(1,6)=count_k06*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,6));

%%%%%%%%%%
%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;

```

```

count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

```

```

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
  for i=721:840 % number of vectors for each coefficient matrix to be classified
    count_times=count_times+1; % thi is to count ech 5 times to classify
    if ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
        &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
        &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
        &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
        &&(eu_x07(i,j)>eu_x10(i,j)))
      count_07=count_07+1;
      classif(i,j*2-1:j*2)=[straw 0];
    elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
      count_02=count_02+1;
      classif(i,j*2-1:j*2)=[french 0];
    elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
      count_03=count_03+1;
      classif(i,j*2-1:j*2)=[liz 0];
    elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))

```

```

count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
&&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
&&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
&&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
&&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
&&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
&&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
&&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
&&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
&&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
&&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
&&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
&&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
&&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
&&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
&&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
&&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
&&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
&&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
&&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...

```

```

        &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
        &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
        &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
        &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
        &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x07(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french straw];
    end
elseif (eu_x07(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz straw];
    end
elseif (eu_x07(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw net];
    else

```

```

        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net straw];
    end
elseif (eu_x07(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin straw];
    end
elseif (eu_x07(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth straw];
    end
elseif (eu_x07(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    end
elseif (eu_x07(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;

```

```

        classif(i,j*2-1:j*2)=[straw strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth straw];
    end
elseif (eu_x07(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water straw];
    end
elseif (eu_x07(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave straw];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork 0];
    end
end

```

```

elseif((count_02>count_01)&&(count_02>count_03)...
&&(count_02>count_04)&&(count_02>count_05)...
&&(count_02>count_06)&&(count_02>count_07)...
&&(count_02>count_08)&&(count_02>count_09)...
&&(count_02>count_10))
count_k02=count_k02+1;
results(i_count,13:14)=[french 0];
elseif((count_03>count_02)&&(count_03>count_01)...
&&(count_03>count_04)&&(count_03>count_05)...
&&(count_03>count_06)&&(count_03>count_07)...
&&(count_03>count_08)&&(count_03>count_09)...
&&(count_03>count_10))
count_k03=count_k03+1;
results(i_count,13:14)=[liz 0];
elseif((count_04>count_02)&&(count_04>count_03)...
&&(count_04>count_01)&&(count_04>count_05)...
&&(count_04>count_06)&&(count_04>count_07)...
&&(count_04>count_08)&&(count_04>count_09)...
&&(count_04>count_10))
count_k04=count_k04+1;
results(i_count,13:14)=[net 0];
elseif((count_05>count_02)&&(count_05>count_03)...
&&(count_05>count_04)&&(count_05>count_01)...
&&(count_05>count_06)&&(count_05>count_07)...
&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,13:14)=[repskin 0];
elseif((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...
&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;

```

```

    results(i_count,13:14)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,13:14)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,13:14)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,13:14)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork french];

```

```

else
    count_k02=count_k02+1;
    results(i_count,13:14)=[french cork];
end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,13:14)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k01=count_k01+1;
        results(i_count,13:14)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water cork];
    end
elseif (count_01==count_10)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,13:14)=[cork weave];
else
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave cork];
end
elseif (count_02==count_03)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,13:14)=[french liz];
else
    count_k03=count_k03+1;
    results(i_count,13:14)=[liz french];
end
elseif (count_02==count_04)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,13:14)=[french net];
else
    count_k04=count_k04+1;
    results(i_count,13:14)=[net french];
end
elseif (count_02==count_05)
    random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,13:14)=[french repskin];
else
    count_k05=count_k05+1;
    results(i_count,13:14)=[repskin french];

```

```

end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french water];
    else

```

```

        count_k09=count_k09+1;
        results(i_count,13:14)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,13:14)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;

```

```

        results(i_count,13:14)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;

```

```

if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,13:14)=[liz weave];
else
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave liz];
end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw net];
    end
end

```

```

elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin scloth];
    else
        count_k06=count_k06+1;

```

```

        results(i_count,13:14)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin weave];

```

```

else
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave repskin];
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave straw];
    end
elseif (count_08==count_09)

```

```

    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,13:14)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave water];
    end
end
end
end
end

% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end

fprintf('\n\nResults for Approximation Coefficients of Training Samples for Straw\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results straw/cork=%4.0f\n',count_k01);

```

```

fprintf('Classification results straw/french=%4.0f\n',count_k02);
fprintf('Classification results straw/liz=%4.0f\n',count_k03);
fprintf('Classification results straw/net=%4.0f\n',count_k04);
fprintf('Classification results straw/repskin=%4.0f\n',count_k05);
fprintf('Classification results straw/scloth=%4.0f\n',count_k06);
fprintf('Classification results straw/straw=%4.0f\n',count_k07);
fprintf('Classification results straw/strcloth=%4.0f\n',count_k08);
fprintf('Classification results straw/water=%4.0f\n',count_k09);
fprintf('Classification results straw/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(7,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,7)=count_k07*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,7));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;

```

```

count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
    for i=841:960 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
            &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...

```

```

    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
    &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
    &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
    &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
    &&(eu_x02(i,j)>eu_x10(i,j)))
count_02=count_02+1;
classif(i,j*2-1;j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1;j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1;j*2)=[weave 0];

```

```

elseif (eu_x08(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french strcloth];
    end
elseif (eu_x08(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz strcloth];
    end
elseif (eu_x08(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net strcloth];
    end
elseif (eu_x08(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth repskin];
    else
        count_05=count_05+1;

```

```

        classif(i,j*2-1:j*2)=[repskin strcloth];
    end
elseif (eu_x08(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth strcloth];
    end
elseif (eu_x08(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw strcloth];
    end
elseif (eu_x08(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    end
elseif (eu_x08(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth water];

```

```

else
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water strcloth];
end
elseif (eu_x08(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave strcloth];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,15:16)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...

```

```

    &&(count_03>count_08)&&(count_03>count_09)...
    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,15:16)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,15:16)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,15:16)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,15:16)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,15:16)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...

```

```

    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,15:16)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,15:16)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,15:16)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,15:16)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork liz];
    else

```

```

        count_k03=count_k03+1;
        results(i_count,15:16)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,15:16)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;

```

```

        results(i_count,15:16)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,15:16)=[french liz];
else
    count_k03=count_k03+1;
    results(i_count,15:16)=[liz french];
end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,15:16)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth french];
    end
end

```

```

elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french weave];
    else
        count_k10=count_k10+1;

```

```

        results(i_count,15:16)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,15:16)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz straw];
    end

```

```

else
    count_k07=count_k07+1;
    results(i_count,15:16)=[straw liz];
end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k04=count_k04+1;
        results(i_count,15:16)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth net];
    end
elseif (count_04==count_09)

```

```

random_pick=rand;
if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,15:16)=[net water];
else
    count_k09=count_k09+1;
    results(i_count,15:16)=[water net];
end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw repskin];
    end

```

```

end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[sclloth straw];
    else

```

```

        count_k07=count_k07+1;
        results(i_count,15:16)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;

```

```

        results(i_count,15:16)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;

```



```

a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k0
9+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(8,:)=[count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08
count_k09 count_k10];
perc(1,8)=count_k08*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,8));

```

```

%%%%%%%%%
%%%%%%%%%

```

```

% Classification process %

```

```

count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

```

```

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;

```

```

% Data classified

```

```

cork=1;
french=2;

```

```

liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
    for i=961:1080 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
            &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
            &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
            &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
            &&(eu_x09(i,j)>eu_x10(i,j)))
            count_09=count_09+1;
            classif(i,j*2-1:j*2)=[water 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))

```

```

count_02=count_02+1;
classif(i,j*2-1;j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
&&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
&&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
&&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
&&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
&&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
&&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
&&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
&&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
&&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
&&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
&&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
&&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
&&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
&&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
&&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
&&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
&&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
&&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
&&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...

```

```

        &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
        &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
        &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
        &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
        &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
        &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
        &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
        &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
        &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
        &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
        &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
        &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
        &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x09(i,j)==eu_x02(i,j))
    random_pick=rand;        % We use rand to let random selection when
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french water];
    end
elseif (eu_x09(i,j)==eu_x03(i,j))

```

```

random_pick=rand;
if random_pick>.5
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water liz];
else
    count_03=count_03+1;
    classif(i,j*2-1:j*2)=[liz water];
end
elseif (eu_x09(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net water];
    end
elseif (eu_x09(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin water];
    end
elseif (eu_x09(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth water];
    end

```

```

end
elseif (eu_x09(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw water];
    end
elseif (eu_x09(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth water];
    end
elseif (eu_x09(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    end
elseif (eu_x09(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water weave];
    else

```

```

        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave water];
    end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,17:18)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;

```

```

    results(i_count,17:18)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,17:18)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,17:18)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,17:18)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,17:18)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))

```

```

count_k09=count_k09+1;
results(i_count,17:18)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...
&&(count_10>count_06)&&(count_10>count_07)...
&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,17:18)=[weave 0];
elseif (count_01==count_02)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork french];
else
count_k02=count_k02+1;
results(i_count,17:18)=[french cork];
end
elseif (count_01==count_03)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork liz];
else
count_k03=count_k03+1;
results(i_count,17:18)=[liz cork];
end
elseif (count_01==count_04)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork net];
else
count_k04=count_k04+1;

```

```

        results(i_count,17:18)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork streloth];
    end
end

```

```

else
    count_k08=count_k08+1;
    results(i_count,17:18)=[strcloth cork];
end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k02=count_k02+1;
        results(i_count,17:18)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,17:18)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw french];
    end
elseif (count_02==count_08)

```

```

random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,17:18)=[french strcloth];
else
    count_k08=count_k08+1;
    results(i_count,17:18)=[strcloth french];
end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,17:18)=[net liz];
    end

```

```

end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz strecloth];
    else

```

```

        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;

```

```

        results(i_count,17:18)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;

```

```

if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,17:18)=[net weave];
else
    count_k10=count_k10+1;
    results(i_count,17:18)=[weave net];
end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth repskin];
    end
end

```

```

elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth strecloth];
    else
        count_k08=count_k08+1;

```

```

        results(i_count,17:18)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[sclotch water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[sclotch weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw water];

```

```

else
    count_k09=count_k09+1;
    results(i_count,17:18)=[water straw];
end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k09=count_k09+1;
        results(i_count,17:18)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave water];
    end
end
end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Water\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results water/cork=%4.0f\n',count_k01);
fprintf('Classification results water/french=%4.0f\n',count_k02);
fprintf('Classification results water/liz=%4.0f\n',count_k03);
fprintf('Classification results water/net=%4.0f\n',count_k04);
fprintf('Classification results water/repskin=%4.0f\n',count_k05);
fprintf('Classification results water/scloth=%4.0f\n',count_k06);
fprintf('Classification results water/straw=%4.0f\n',count_k07);
fprintf('Classification results water/streloth=%4.0f\n',count_k08);
fprintf('Classification results water/water=%4.0f\n',count_k09);
fprintf('Classification results water/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(9,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,9)=count_k09*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,9));

%%%%%%%%%
%%%%%%%%%

% Classification process %

```

```
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;
```

```
count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
```

```

i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5)% number of training samples
    for i=1081:1200 % number of vectors for each coefficient matrix to be classified
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
            &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
            &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
            &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
            &&(eu_x10(i,j)>eu_x01(i,j)))
            count_10=count_10+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strecloth 0];

```

```

elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork 0];
elseif (eu_x10(i,j)==eu_x02(i,j))
random_pick=rand;      % We use rand to let random selection when
if random_pick>.5
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave random_selection];
else
    count_02=count_02+1;
    classif(i,j*2-1:j*2)=[french weave];
end
elseif (eu_x10(i,j)==eu_x03(i,j))
random_pick=rand;
if random_pick>.5
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave random_selection];
else
    count_03=count_03+1;
    classif(i,j*2-1:j*2)=[liz weave];
end
elseif (eu_x10(i,j)==eu_x04(i,j))
random_pick=rand;

```

```

if random_pick>.5
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave random_selection];
else
    count_04=count_04+1;
    classif(i,j*2-1:j*2)=[net weave];
end
elseif (eu_x10(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin weave];
    end
elseif (eu_x10(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth weave];
    end
elseif (eu_x10(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw weave];
    end
end

```

```

elseif (eu_x10(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth weave];
    end
elseif (eu_x10(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water weave];
    end
elseif (eu_x10(i,j)==eu_x01(i,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave random_selection];
    else
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    end
end
end
if (count_times==3) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...

```

```

&&(count_01>count_08)&&(count_01>count_09)...
&&(count_01>count_10))
count_k01=count_k01+1;
results(i_count,19:20)=[cork 0];
elseif((count_02>count_01)&&(count_02>count_03)...
&&(count_02>count_04)&&(count_02>count_05)...
&&(count_02>count_06)&&(count_02>count_07)...
&&(count_02>count_08)&&(count_02>count_09)...
&&(count_02>count_10))
count_k02=count_k02+1;
results(i_count,19:20)=[french 0];
elseif((count_03>count_02)&&(count_03>count_01)...
&&(count_03>count_04)&&(count_03>count_05)...
&&(count_03>count_06)&&(count_03>count_07)...
&&(count_03>count_08)&&(count_03>count_09)...
&&(count_03>count_10))
count_k03=count_k03+1;
results(i_count,19:20)=[liz 0];
elseif((count_04>count_02)&&(count_04>count_03)...
&&(count_04>count_01)&&(count_04>count_05)...
&&(count_04>count_06)&&(count_04>count_07)...
&&(count_04>count_08)&&(count_04>count_09)...
&&(count_04>count_10))
count_k04=count_k04+1;
results(i_count,19:20)=[net 0];
elseif((count_05>count_02)&&(count_05>count_03)...
&&(count_05>count_04)&&(count_05>count_01)...
&&(count_05>count_06)&&(count_05>count_07)...
&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,19:20)=[repskin 0];
elseif((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...

```

```

&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;
results(i_count,19:20)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...
&&(count_07>count_10))
count_k07=count_k07+1;
results(i_count,19:20)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
&&(count_08>count_04)&&(count_08>count_05)...
&&(count_08>count_06)&&(count_08>count_07)...
&&(count_08>count_01)&&(count_08>count_09)...
&&(count_08>count_10))
count_k08=count_k08+1;
results(i_count,19:20)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
&&(count_09>count_04)&&(count_09>count_05)...
&&(count_09>count_06)&&(count_09>count_07)...
&&(count_09>count_08)&&(count_09>count_01)...
&&(count_09>count_10))
count_k09=count_k09+1;
results(i_count,19:20)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...
&&(count_10>count_06)&&(count_10>count_07)...
&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,19:20)=[weave 0];
elseif (count_01==count_02)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,19:20)=[cork french];
else
    count_k02=count_k02+1;
    results(i_count,19:20)=[french cork];
end
elseif(count_01==count_03)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,19:20)=[cork liz];
else
    count_k03=count_k03+1;
    results(i_count,19:20)=[liz cork];
end
elseif(count_01==count_04)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,19:20)=[cork net];
else
    count_k04=count_k04+1;
    results(i_count,19:20)=[net cork];
end
elseif(count_01==count_05)
    random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,19:20)=[cork repskin];
else
    count_k05=count_k05+1;
    results(i_count,19:20)=[repskin cork];

```

```

end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork water];
    else

```

```

        count_k09=count_k09+1;
        results(i_count,19:20)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,19:20)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;

```

```

        results(i_count,19:20)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,19:20)=[french water];
else
    count_k09=count_k09+1;
    results(i_count,19:20)=[water french];
end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,19:20)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin liz];
    end
end

```

```

elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz water];
    else
        count_k09=count_k09+1;

```

```

        results(i_count,19:20)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net straw];

```

```

else
    count_k07=count_k07+1;
    results(i_count,19:20)=[straw net];
end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water repskin];
    end
elseif (count_05==count_10)

```

```

random_pick=rand;
if random_pick>.5
    count_k05=count_k05+1;
    results(i_count,19:20)=[repskin weave];
else
    count_k10=count_k10+1;
    results(i_count,19:20)=[weave repskin];
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water scloth];
    end

```

```

end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,19:20)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,19:20)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave water];
    end
end
end
end
% results(i*4-3:i*4,:)=classif; % this part is going to be used for fusion

```

```

end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Weave\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results weave/cork=%4.0f\n',count_k01);
fprintf('Classification results weave/french=%4.0f\n',count_k02);
fprintf('Classification results weave/liz=%4.0f\n',count_k03);
fprintf('Classification results weave/net=%4.0f\n',count_k04);
fprintf('Classification results weave/repskin=%4.0f\n',count_k05);
fprintf('Classification results weave/scloth=%4.0f\n',count_k06);
fprintf('Classification results weave/straw=%4.0f\n',count_k07);
fprintf('Classification results weave/strcloth=%4.0f\n',count_k08);
fprintf('Classification results weave/water=%4.0f\n',count_k09);
fprintf('Classification results weave/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(10,:)=[count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,10)=count_k10*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,10));
rec_perc=[rec perc'];
toc

```

APPENDIX C3 K-NEAREST NEIGHBOR K=5 CLASSIFIER

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% K-Nearest Neighbor K=5 Classifier %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% All four Coefficient Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% KNN K=5 %
clear
tic
% loading the matrices of training & testing samples
load training_testing_rand_db1.mat
% load training_testing_rand_haar.mat
% load training_testing_rand_bior11.mat
% load training_testing_rand_rbio11.mat

```

```

[M N]=size(test_all);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X_DATA Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% By changing "x_data=..." we can classify any image can be training or
% testing, so this will wor for any type of data

% Discriminant function for x_data 01_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix (for testing the classifier we will use 24
for the training samples)
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k); % by changing x_data we will classify different images
            mean_01=train_cork(i*4-3:i*4,1);
            % cov_data1=train_cork(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_01(i,k)=eu_fcn(x_data,mean_01); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_01=-sort(-eu_01); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x01(j*5-4:j*5,:)=eu_01(1:5,:); % So the minimum distance for the first coefficient matrix is the
first row
    end
% Discriminant function for x_data 02_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_02=train_french(i*4-3:i*4,1);
            % cov_data1=train_repskin(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_02(i,k)=eu_fcn(x_data,mean_02); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_02=-sort(-eu_02); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x02(j*5-4:j*5,:)=eu_02(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
    end
% Discriminant function for x_data 03_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_03=train_liz(i*4-3:i*4,1);
            % cov_data1=train_straw(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_03(i,k)=eu_fcn(x_data,mean_03); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
    end
end

```

```

end
eu_03=-sort(-eu_03); % We first sort the columns in order to get the minimum distances for each
vector
end
eu_x03(j*5-4:j*5,:)=eu_03(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 04_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_04=train_net(i*4-3:i*4,1);
% cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
eu_04(i,k)=eu_fcn(x_data,mean_04); % this saves for each coefficient matrix all the distances
between all the features of the other classes
end
eu_04=-sort(-eu_04); % We first sort the columns in order to get the minimum distances for each
vector
end
eu_x04(j*5-4:j*5,:)=eu_04(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 05_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_05=train_repskin(i*4-3:i*4,1);
% cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
eu_05(i,k)=eu_fcn(x_data,mean_05); % this saves for each coefficient matrix all the distances
between all the features of the other classes
end
eu_05=-sort(-eu_05); % We first sort the columns in order to get the minimum distances for each
vector
end
eu_x05(j*5-4:j*5,:)=eu_05(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 06_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_06=train_scloth(i*4-3:i*4,1);
% cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
eu_06(i,k)=eu_fcn(x_data,mean_06); % this saves for each coefficient matrix all the distances
between all the features of the other classes
end

```

```

    eu_06=-sort(-eu_06); % We first sort the columns in order to get the minimum distances for each
vector
end
    eu_x06(j*5-4:j*5,:)=eu_06(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 07_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_07=train_straw(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_07(i,k)=eu_fcn(x_data,mean_07); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_07=-sort(-eu_07); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x07(j*5-4:j*5,:)=eu_07(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
        end

% Discriminant function for x_data 08_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_08=train_strcloth(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_08(i,k)=eu_fcn(x_data,mean_08); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_08=-sort(-eu_08); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x08(j*5-4:j*5,:)=eu_08(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
        end

% Discriminant function for x_data 09_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_09=train_water(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_09(i,k)=eu_fcn(x_data,mean_09); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end

```

```

    eu_09=-sort(-eu_09); % We first sort the columns in order to get the minimum distances for each
vector
end
    eu_x09(j*5-4:j*5,:)=eu_09(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 10_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_10=train_weave(i*4-3:i*4,1);
            % cov_data1=train_strcloth(i*4-3:i*4,5:12); % if using ML classifier as the distance measure
            eu_10(i,k)=eu_fcn(x_data,mean_10); % this saves for each coefficient matrix all the distances
between all the features of the other classes
        end
        eu_10=-sort(-eu_10); % We first sort the columns in order to get the minimum distances for each
vector
        end
        eu_x10(j*5-4:j*5,:)=eu_10(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
        end

% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;

```

```

repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=1:200 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))
            count_04=count_04+1;
            classif(i,j*2-1:j*2)=[net 0];
        elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
            &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
            &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
            &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
            &&(eu_x05(i,j)>eu_x10(i,j)))
            count_05=count_05+1;
            classif(i,j*2-1:j*2)=[net 0];
        end
    end
end

```

```

    &&(eu_x05(i,j)>eu_x10(i,j)))
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
    count_06=count_06+1;
    classif(i,j*2-1:j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
    count_07=count_07+1;
    classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand; % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];

```

```

else
    count_03=count_03+1;
    classif(i,j*2-1:j*2)=[liz cork];
end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork stcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5

```

```

        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,1:2)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;
        results(i_count,1:2)=[net 0];
    elseif ((count_05>count_02)&&(count_05>count_03)...
        &&(count_05>count_04)&&(count_05>count_01)...
        &&(count_05>count_06)&&(count_05>count_07)...
        &&(count_05>count_08)&&(count_05>count_09)...
        &&(count_05>count_10))

```

```

    count_k05=count_k05+1;
    results(i_count,1:2)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,1:2)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,1:2)=[strelcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,1:2)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,1:2)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,1:2)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork liz];
    else

```

```

        count_k03=count_k03+1;
        results(i_count,1:2)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,1:2)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;

```

```

        results(i_count,1:2)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,1:2)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,1:2)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[sclotch french];
    end
elseif (count_02==count_07)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,1:2)=[french straw];
else
    count_k07=count_k07+1;
    results(i_count,1:2)=[straw french];
end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,1:2)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,1:2)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin liz];
    end
end

```

```

elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[stcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,1:2)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net repskin];
    else
        count_k05=count_k05+1;

```

```

        results(i_count,1:2)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,1:2)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin scloth];

```

```

else
    count_k06=count_k06+1;
    results(i_count,1:2)=[scloth repskin];
end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw repskin];
    end
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin strecth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strecth repskin];
    end
end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water repskin];
    end
end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,1:2)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave repskin];
    end
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw scloth];
    end
end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,1:2)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,1:2)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,1:2)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,1:2)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,1:2)=[weave straw];
    end
elseif (count_08==count_09)

```

```

        random_pick=rand;
        if random_pick>.5
            count_k08=count_k08+1;
            results(i_count,1:2)=[strcloth water];
        else
            count_k09=count_k09+1;
            results(i_count,1:2)=[water strcloth];
        end
    elseif (count_08==count_10)
        random_pick=rand;
        if random_pick>.5
            count_k08=count_k08+1;
            results(i_count,1:2)=[strcloth weave];
        else
            count_k10=count_k10+1;
            results(i_count,1:2)=[weave strcloth];
        end
    elseif (count_09==count_10)
        random_pick=rand;
        if random_pick>.5
            count_k09=count_k09+1;
            results(i_count,1:2)=[water weave];
        else
            count_k10=count_k10+1;
            results(i_count,1:2)=[weave water];
        end
    end
end
end
end
    % results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Cork\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results cork/cork=%4.0f\n',count_k01);
fprintf('Classification results cork/french=%4.0f\n',count_k02);
fprintf('Classification results cork/liz=%4.0f\n',count_k03);
fprintf('Classification results cork/net=%4.0f\n',count_k04);
fprintf('Classification results cork/repskin=%4.0f\n',count_k05);
fprintf('Classification results cork/scloth=%4.0f\n',count_k06);
fprintf('Classification results cork/straw=%4.0f\n',count_k07);
fprintf('Classification results cork/strcloth=%4.0f\n',count_k08);
fprintf('Classification results cork/water=%4.0f\n',count_k09);
fprintf('Classification results cork/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(1,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,1)=count_k01*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,1));

```



```

    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
    count_01=count_01+1;
    classif(i,j*2-1;j*2)=[cork 0];
elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
    &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
    &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
    &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
    &&(eu_x02(i,j)>eu_x10(i,j)))
    count_02=count_02+1;
    classif(i,j*2-1;j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
    count_03=count_03+1;
    classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
    count_04=count_04+1;
    classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
    count_05=count_05+1;
    classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
    count_06=count_06+1;
    classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
    count_07=count_07+1;
    classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
    count_08=count_08+1;

```

```

    classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))

```

```

random_pick=rand;
if random_pick>.5
    count_01=count_01+1;
    classif(i,j*2-1:j*2)=[cork scloth];
else
    count_06=count_06+1;
    classif(i,j*2-1:j*2)=[scloth cork];
end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...

```

```

    &&(count_01>count_10))
    count_k01=count_k01+1;
    results(i_count,3:4)=[cork 0];
elseif ((count_02>count_01)&&(count_02>count_03)...
    &&(count_02>count_04)&&(count_02>count_05)...
    &&(count_02>count_06)&&(count_02>count_07)...
    &&(count_02>count_08)&&(count_02>count_09)...
    &&(count_02>count_10))
    count_k02=count_k02+1;
    results(i_count,3:4)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
    &&(count_03>count_04)&&(count_03>count_05)...
    &&(count_03>count_06)&&(count_03>count_07)...
    &&(count_03>count_08)&&(count_03>count_09)...
    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,3:4)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,3:4)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,3:4)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,3:4)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,3:4)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,3:4)=[streloth 0];

```

```

elseif((count_09>count_02)&&(count_09>count_03)...
&&(count_09>count_04)&&(count_09>count_05)...
&&(count_09>count_06)&&(count_09>count_07)...
&&(count_09>count_08)&&(count_09>count_01)...
&&(count_09>count_10))
count_k09=count_k09+1;
results(i_count,3:4)=[water 0];
elseif((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...
&&(count_10>count_06)&&(count_10>count_07)...
&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,3:4)=[weave 0];
elseif(count_01==count_02)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,3:4)=[cork french];
else
count_k02=count_k02+1;
results(i_count,3:4)=[french cork];
end
elseif(count_01==count_03)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,3:4)=[cork liz];
else
count_k03=count_k03+1;
results(i_count,3:4)=[liz cork];
end
elseif(count_01==count_04)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,3:4)=[cork net];
else
count_k04=count_k04+1;
results(i_count,3:4)=[net cork];
end
elseif(count_01==count_05)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,3:4)=[cork repskin];
else
count_k05=count_k05+1;
results(i_count,3:4)=[repskin cork];
end
elseif(count_01==count_06)
random_pick=rand;

```

```

if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,3:4)=[cork scloth];
else
    count_k06=count_k06+1;
    results(i_count,3:4)=[scloth cork];
end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,3:4)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz french];
    end
end

```

```

elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,3:4)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french water];
    else
        count_k09=count_k09+1;

```

```

        results(i_count,3:4)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,3:4)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,3:4)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz strcloth];
    end

```

```

else
    count_k08=count_k08+1;
    results(i_count,3:4)=[strcloth liz];
end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water liz];
    end
end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,3:4)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave liz];
    end
end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin net];
    end
end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth net];
    end
end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw net];
    end
end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k04=count_k04+1;
        results(i_count,3:4)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,3:4)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,3:4)=[sclotch repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth repskin];
    end
elseif (count_05==count_09)

```

```

random_pick=rand;
if random_pick>.5
    count_k05=count_k05+1;
    results(i_count,3:4)=[repskin water];
else
    count_k09=count_k09+1;
    results(i_count,3:4)=[water repskin];
end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,3:4)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,3:4)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,3:4)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,3:4)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,3:4)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,3:4)=[weave scloth];
    end

```

```

end
elseif (count_07==count_08)
random_pick=rand;
if random_pick>.5
count_k07=count_k07+1;
results(i_count,3:4)=[straw strcloth];
else
count_k08=count_k08+1;
results(i_count,3:4)=[strcloth straw];
end
elseif (count_07==count_09)
random_pick=rand;
if random_pick>.5
count_k07=count_k07+1;
results(i_count,3:4)=[straw water];
else
count_k09=count_k09+1;
results(i_count,3:4)=[water straw];
end
elseif (count_07==count_10)
random_pick=rand;
if random_pick>.5
count_k07=count_k07+1;
results(i_count,3:4)=[straw weave];
else
count_k10=count_k10+1;
results(i_count,3:4)=[weave straw];
end
elseif (count_08==count_09)
random_pick=rand;
if random_pick>.5
count_k08=count_k08+1;
results(i_count,3:4)=[strcloth water];
else
count_k09=count_k09+1;
results(i_count,3:4)=[water strcloth];
end
elseif (count_08==count_10)
random_pick=rand;
if random_pick>.5
count_k08=count_k08+1;
results(i_count,3:4)=[strcloth weave];
else
count_k10=count_k10+1;
results(i_count,3:4)=[weave strcloth];
end
elseif (count_09==count_10)
random_pick=rand;
if random_pick>.5
count_k09=count_k09+1;
results(i_count,3:4)=[water weave];
else

```

```

        count_k10=count_k10+1;
        results(i_count,3:4)=[weave water];
    end
end
end
end
    % results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for French\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results french/cork=%4.0f\n',count_k01);
fprintf('Classification results french/french=%4.0f\n',count_k02);
fprintf('Classification results french/liz=%4.0f\n',count_k03);
fprintf('Classification results french/net=%4.0f\n',count_k04);
fprintf('Classification results french/repskin=%4.0f\n',count_k05);
fprintf('Classification results french/scloth=%4.0f\n',count_k06);
fprintf('Classification results french/straw=%4.0f\n',count_k07);
fprintf('Classification results french/streloth=%4.0f\n',count_k08);
fprintf('Classification results french/water=%4.0f\n',count_k09);
fprintf('Classification results french/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(2,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,2)=count_k02*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,2));

%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;

```

```

count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=401:600 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))

```

```

count_04=count_04+1;
classif(i,j*2-1:j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
&&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
&&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
&&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
&&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
&&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
&&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
&&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
&&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1:j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
&&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
&&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
&&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
&&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
&&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
&&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
&&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
&&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
&&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
&&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
&&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
&&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
&&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
&&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
&&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
&&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
random_pick=rand; % We use rand to let random selection when
if random_pick>.5
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork french];
else
count_02=count_02+1;
classif(i,j*2-1:j*2)=[french cork];

```

```

end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else

```

```

        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,5:6)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;

```

```

    results(i_count,5:6)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,5:6)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,5:6)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,5:6)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,5:6)=[strelotch 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,5:6)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,5:6)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,5:6)=[french cork];
    end
end

```

```

elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,5:6)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork streloth];
    else
        count_k08=count_k08+1;

```

```

        results(i_count,5:6)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,5:6)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,5:6)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french scloth];
    end

```

```

else
    count_k06=count_k06+1;
    results(i_count,5:6)=[scloth french];
end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,5:6)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,5:6)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k03=count_k03+1;
        results(i_count,5:6)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[sclloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,5:6)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave liz];
    end
elseif (count_04==count_05)

```

```

random_pick=rand;
if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,5:6)=[net repskin];
else
    count_k05=count_k05+1;
    results(i_count,5:6)=[repskin net];
end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,5:6)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave net];
    end

```

```

end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth repskin];
    end
end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw repskin];
    end
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth repskin];
    end
end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water repskin];
    end
end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,5:6)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave repskin];
    end
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth straw];
    else

```

```

        count_k07=count_k07+1;
        results(i_count,5:6)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,5:6)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,5:6)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;

```

```

        results(i_count,5:6)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,5:6)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,5:6)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,5:6)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,5:6)=[weave water];
    end
end
end
end
end
% results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Liz\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results liz/cork=%4.0f\n',count_k01);
fprintf('Classification results liz/french=%4.0f\n',count_k02);
fprintf('Classification results liz/liz=%4.0f\n',count_k03);
fprintf('Classification results liz/net=%4.0f\n',count_k04);
fprintf('Classification results liz/repskin=%4.0f\n',count_k05);
fprintf('Classification results liz/scloth=%4.0f\n',count_k06);
fprintf('Classification results liz/straw=%4.0f\n',count_k07);
fprintf('Classification results liz/strcloth=%4.0f\n',count_k08);
fprintf('Classification results liz/water=%4.0f\n',count_k09);
fprintf('Classification results liz/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;

```

```

fprintf('\nTotal samples=%4.0f\n',a);
rec(3,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08
count_k09 count_k10];
perc(1,3)=count_k03*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,3));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples

```

```

for i=601:800 % number of distance calculated to be classified 5 for each 24 vectors equals 120
count_times=count_times+1; % thi is to count ech 5 times to classify
if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
    &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
    &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
    &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
    &&(eu_x01(i,j)>eu_x10(i,j)))
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork 0];
elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
    &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
    &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
    &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
    &&(eu_x02(i,j)>eu_x10(i,j)))
count_02=count_02+1;
classif(i,j*2-1:j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
    &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
    &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1:j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1:j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1:j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];

```

```

else
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin cork];
end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5)    % Classification of K Nearest Neighborhood
    count_times=0;

```

```

i_count=i_count+1;
if ((count_01>count_02)&&(count_01>count_03)...
    &&(count_01>count_04)&&(count_01>count_05)...
    &&(count_01>count_06)&&(count_01>count_07)...
    &&(count_01>count_08)&&(count_01>count_09)...
    &&(count_01>count_10))
    count_k01=count_k01+1;
    results(i_count,7:8)=[cork 0];
elseif ((count_02>count_01)&&(count_02>count_03)...
    &&(count_02>count_04)&&(count_02>count_05)...
    &&(count_02>count_06)&&(count_02>count_07)...
    &&(count_02>count_08)&&(count_02>count_09)...
    &&(count_02>count_10))
    count_k02=count_k02+1;
    results(i_count,7:8)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
    &&(count_03>count_04)&&(count_03>count_05)...
    &&(count_03>count_06)&&(count_03>count_07)...
    &&(count_03>count_08)&&(count_03>count_09)...
    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,7:8)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,7:8)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,7:8)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,7:8)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,7:8)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...

```

```

    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,7:8)=[strecth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,7:8)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,7:8)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,7:8)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,7:8)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork repskin];
    else

```

```

        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork streloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[streloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,7:8)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;

```

```

        results(i_count,7:8)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,7:8)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,7:8)=[french water];
else
    count_k09=count_k09+1;
    results(i_count,7:8)=[water french];
end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,7:8)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,7:8)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw liz];
    end
end

```

```

elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,7:8)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net straw];
    else
        count_k07=count_k07+1;

```

```

        results(i_count,7:8)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,7:8)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin strcloth];

```

```

else
    count_k08=count_k08+1;
    results(i_count,7:8)=[strcloth repskin];
end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water repskin];
    end
end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,7:8)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave repskin];
    end
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw scloth];
    end
end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth scloth];
    end
end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water scloth];
    end
end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k06=count_k06+1;
        results(i_count,7:8)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,7:8)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,7:8)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,7:8)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,7:8)=[weave strcloth];
    end
elseif (count_09==count_10)

```

```

        random_pick=rand;
        if random_pick>.5
            count_k09=count_k09+1;
            results(i_count,7:8)=[water weave];
        else
            count_k10=count_k10+1;
            results(i_count,7:8)=[weave water];
        end
    end
end
end
end
% results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Net\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results net/cork=%4.0f\n',count_k01);
fprintf('Classification results net/french=%4.0f\n',count_k02);
fprintf('Classification results net/liz=%4.0f\n',count_k03);
fprintf('Classification results net/net=%4.0f\n',count_k04);
fprintf('Classification results net/repskin=%4.0f\n',count_k05);
fprintf('Classification results net/scloth=%4.0f\n',count_k06);
fprintf('Classification results net/straw=%4.0f\n',count_k07);
fprintf('Classification results net/streloth=%4.0f\n',count_k08);
fprintf('Classification results net/water=%4.0f\n',count_k09);
fprintf('Classification results net/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(4,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,4)=count_k04*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,4));
%%%%%%%%%%
%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;

```

```

count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=801:1000 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...

```

```

    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
    count_04=count_04+1;
    classif(i,j*2-1:j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
    count_05=count_05+1;
    classif(i,j*2-1:j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
    count_06=count_06+1;
    classif(i,j*2-1:j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
    count_07=count_07+1;
    classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strecloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand; % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;

```

```

        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_01=count_01+1;
    classif(i,j*2-1:j*2)=[cork strcloth];
else
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strcloth cork];
end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,9:10)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...

```

```

    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,9:10)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,9:10)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,9:10)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,9:10)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,9:10)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,9:10)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork french];

```

```

else
    count_k02=count_k02+1;
    results(i_count,9:10)=[french cork];
end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,9:10)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k01=count_k01+1;
        results(i_count,9:10)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,9:10)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,9:10)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin french];
    end
elseif (count_02==count_06)

```

```

random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,9:10)=[french scloth];
else
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth french];
end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,9:10)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,9:10)=[net liz];
    end

```

```

end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,9:10)=[liz weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,9:10)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,9:10)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;

```

```

        results(i_count,9:10)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,9:10)=[sclotch repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,9:10)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;

```

```

if random_pick>.5
    count_k06=count_k06+1;
    results(i_count,9:10)=[scloth straw];
else
    count_k07=count_k07+1;
    results(i_count,9:10)=[straw scloth];
end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,9:10)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water straw];
    end
end

```

```

elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,9:10)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,9:10)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,9:10)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,9:10)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,9:10)=[weave water];
    end
end
end
end
end
% results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Repskin\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results repskin/cork=%4.0f\n',count_k01);
fprintf('Classification results repskin/french=%4.0f\n',count_k02);
fprintf('Classification results repskin/liz=%4.0f\n',count_k03);
fprintf('Classification results repskin/net=%4.0f\n',count_k04);
fprintf('Classification results repskin/repskin=%4.0f\n',count_k05);
fprintf('Classification results repskin/scloth=%4.0f\n',count_k06);
fprintf('Classification results repskin/straw=%4.0f\n',count_k07);
fprintf('Classification results repskin/strcloth=%4.0f\n',count_k08);
fprintf('Classification results repskin/water=%4.0f\n',count_k09);

```

```

fprintf('Classification results repskin/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k0
9+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(5,:)=[count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08
count_k09 count_k10];
perc(1,5)=count_k05*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,5));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

```

```

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=1001:1200 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))
            count_04=count_04+1;
            classif(i,j*2-1:j*2)=[net 0];
        elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
            &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
            &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
            &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
            &&(eu_x05(i,j)>eu_x10(i,j)))
            count_05=count_05+1;
            classif(i,j*2-1:j*2)=[repskin 0];
        elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
            &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
            &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
            &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
            &&(eu_x06(i,j)>eu_x10(i,j)))
            count_06=count_06+1;
            classif(i,j*2-1:j*2)=[scloth 0];
        elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
            &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
            &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...

```

```

        &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
        &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
        &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
        &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
        &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
        &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
        &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
        &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
        &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
        &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
        &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
        &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
        &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
        &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;          % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
end

```

```

elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;

```

```

        classif(i,j*2-1:j*2)=[weave cork];
    end
end
if (count_times==5) % Classification of K Nearest Neighborhood
count_times=0;
i_count=i_count+1;
if ((count_01>count_02)&&(count_01>count_03)...
&&(count_01>count_04)&&(count_01>count_05)...
&&(count_01>count_06)&&(count_01>count_07)...
&&(count_01>count_08)&&(count_01>count_09)...
&&(count_01>count_10))
count_k01=count_k01+1;
results(i_count,11:12)=[cork 0];
elseif ((count_02>count_01)&&(count_02>count_03)...
&&(count_02>count_04)&&(count_02>count_05)...
&&(count_02>count_06)&&(count_02>count_07)...
&&(count_02>count_08)&&(count_02>count_09)...
&&(count_02>count_10))
count_k02=count_k02+1;
results(i_count,11:12)=[french 0];
elseif ((count_03>count_02)&&(count_03>count_01)...
&&(count_03>count_04)&&(count_03>count_05)...
&&(count_03>count_06)&&(count_03>count_07)...
&&(count_03>count_08)&&(count_03>count_09)...
&&(count_03>count_10))
count_k03=count_k03+1;
results(i_count,11:12)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
&&(count_04>count_01)&&(count_04>count_05)...
&&(count_04>count_06)&&(count_04>count_07)...
&&(count_04>count_08)&&(count_04>count_09)...
&&(count_04>count_10))
count_k04=count_k04+1;
results(i_count,11:12)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
&&(count_05>count_04)&&(count_05>count_01)...
&&(count_05>count_06)&&(count_05>count_07)...
&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,11:12)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...
&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;
results(i_count,11:12)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...

```

```

    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,11:12)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,11:12)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,11:12)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,11:12)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,11:12)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,11:12)=[net cork];
    end
elseif (count_01==count_05)

```

```

random_pick=rand;
if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,11:12)=[cork repskin];
else
    count_k05=count_k05+1;
    results(i_count,11:12)=[repskin cork];
end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,11:12)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave cork];
    end

```

```

end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,11:12)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french strecloth];
    else

```

```

        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,11:12)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,11:12)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;

```

```

        results(i_count,11:12)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,11:12)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;

```

```

if random_pick>.5
    count_k04=count_k04+1;
    results(i_count,11:12)=[net straw];
else
    count_k07=count_k07+1;
    results(i_count,11:12)=[straw net];
end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,11:12)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw repskin];
    end
end

```

```

elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,11:12)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[sclloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw sclloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[sclloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth sclloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[sclloth water];
    else
        count_k09=count_k09+1;

```

```

        results(i_count,11:12)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,11:12)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,11:12)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,11:12)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,11:12)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,11:12)=[strcloth weave];
    end

```



```

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=1201:1400 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...

```

```

    &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
    &&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
    &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
    &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
    &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
    &&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
    &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
    &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
    &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
    &&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
    &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1;j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;

```

```

    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else

```

```

        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
if (count_times==5)    % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,13:14)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...

```

```

    &&(count_03>count_10))
    count_k03=count_k03+1;
    results(i_count,13:14)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
    &&(count_04>count_01)&&(count_04>count_05)...
    &&(count_04>count_06)&&(count_04>count_07)...
    &&(count_04>count_08)&&(count_04>count_09)...
    &&(count_04>count_10))
    count_k04=count_k04+1;
    results(i_count,13:14)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
    &&(count_05>count_04)&&(count_05>count_01)...
    &&(count_05>count_06)&&(count_05>count_07)...
    &&(count_05>count_08)&&(count_05>count_09)...
    &&(count_05>count_10))
    count_k05=count_k05+1;
    results(i_count,13:14)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
    &&(count_06>count_04)&&(count_06>count_05)...
    &&(count_06>count_01)&&(count_06>count_07)...
    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,13:14)=[sclotch 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,13:14)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,13:14)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,13:14)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave 0];

```

```

elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,13:14)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork net];
    else
        count_k04=count_k04+1;
        results(i_count,13:14)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork straw];
    else
        count_k07=count_k07+1;

```

```

        results(i_count,13:14)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,13:14)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,13:14)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french repskin];
    end

```

```

else
    count_k05=count_k05+1;
    results(i_count,13:14)=[repskin french];
end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,13:14)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k03=count_k03+1;
        results(i_count,13:14)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,13:14)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,13:14)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water liz];
    end
elseif (count_03==count_10)

```

```

random_pick=rand;
if random_pick>.5
    count_k03=count_k03+1;
    results(i_count,13:14)=[liz weave];
else
    count_k10=count_k10+1;
    results(i_count,13:14)=[weave liz];
end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[sclotch net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water net];
    end

```

```

end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,13:14)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave net];
    end
end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,13:14)=[sclotch repskin];
    end
end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw repskin];
    end
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth repskin];
    end
end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water repskin];
    end
end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,13:14)=[repskin weave];
    else

```

```

        count_k10=count_k10+1;
        results(i_count,13:14)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,13:14)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;

```

```

        results(i_count,13:14)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,13:14)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,13:14)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,13:14)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,13:14)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,13:14)=[weave water];
    end
end
end
end
end
% results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Straw\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results straw/cork=%4.0f\n',count_k01);
fprintf('Classification results straw/french=%4.0f\n',count_k02);
fprintf('Classification results straw/liz=%4.0f\n',count_k03);
fprintf('Classification results straw/net=%4.0f\n',count_k04);

```

```

fprintf('Classification results straw/repskin=%4.0f\n',count_k05);
fprintf('Classification results straw/scloth=%4.0f\n',count_k06);
fprintf('Classification results straw/straw=%4.0f\n',count_k07);
fprintf('Classification results straw/strcloth=%4.0f\n',count_k08);
fprintf('Classification results straw/water=%4.0f\n',count_k09);
fprintf('Classification results straw/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(7,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,7)=count_k07*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,7));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;

```

```

% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=1401:1600 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;
            classif(i,j*2-1:j*2)=[french 0];
        elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
            count_03=count_03+1;
            classif(i,j*2-1:j*2)=[liz 0];
        elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))
            count_04=count_04+1;
            classif(i,j*2-1:j*2)=[net 0];
        elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
            &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
            &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
            &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
            &&(eu_x05(i,j)>eu_x10(i,j)))
            count_05=count_05+1;
            classif(i,j*2-1:j*2)=[repskin 0];
        elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
            &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
            &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
            &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
            &&(eu_x06(i,j)>eu_x10(i,j)))
            count_06=count_06+1;
    end
end

```

```

    classif(i,j*2-1:j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
    count_07=count_07+1;
    classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
    count_08=count_08+1;
    classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
    count_09=count_09+1;
    classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];

```

```

else
    count_04=count_04+1;
    classif(i,j*2-1:j*2)=[net cork];
end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5

```

```

        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,15:16)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;
        results(i_count,15:16)=[net 0];
    elseif ((count_05>count_02)&&(count_05>count_03)...
        &&(count_05>count_04)&&(count_05>count_01)...
        &&(count_05>count_06)&&(count_05>count_07)...
        &&(count_05>count_08)&&(count_05>count_09)...
        &&(count_05>count_10))
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin 0];
    elseif ((count_06>count_02)&&(count_06>count_03)...
        &&(count_06>count_04)&&(count_06>count_05)...
        &&(count_06>count_01)&&(count_06>count_07)...
        &&(count_06>count_08)&&(count_06>count_09)...
        &&(count_06>count_10))
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth 0];

```

```

elseif ((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...
&&(count_07>count_10))
count_k07=count_k07+1;
results(i_count,15:16)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
&&(count_08>count_04)&&(count_08>count_05)...
&&(count_08>count_06)&&(count_08>count_07)...
&&(count_08>count_01)&&(count_08>count_09)...
&&(count_08>count_10))
count_k08=count_k08+1;
results(i_count,15:16)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
&&(count_09>count_04)&&(count_09>count_05)...
&&(count_09>count_06)&&(count_09>count_07)...
&&(count_09>count_08)&&(count_09>count_01)...
&&(count_09>count_10))
count_k09=count_k09+1;
results(i_count,15:16)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...
&&(count_10>count_06)&&(count_10>count_07)...
&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,15:16)=[weave 0];
elseif (count_01==count_02)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,15:16)=[cork french];
else
count_k02=count_k02+1;
results(i_count,15:16)=[french cork];
end
elseif (count_01==count_03)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,15:16)=[cork liz];
else
count_k03=count_k03+1;
results(i_count,15:16)=[liz cork];
end
elseif (count_01==count_04)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,15:16)=[cork net];
else

```

```

        count_k04=count_k04+1;
        results(i_count,15:16)=[net cork];
    end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork stcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[stcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,15:16)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;

```

```

        results(i_count,15:16)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,15:16)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;

```

```

if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,15:16)=[french stcloth];
else
    count_k08=count_k08+1;
    results(i_count,15:16)=[strcloth french];
end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,15:16)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,15:16)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth liz];
    end
end

```

```

elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,15:16)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net scloth];
    else
        count_k06=count_k06+1;

```

```

        results(i_count,15:16)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,15:16)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin straw];
    end

```

```

else
    count_k07=count_k07+1;
    results(i_count,15:16)=[straw repskin];
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,15:16)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[sclloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw sclloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[sclloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth sclloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,15:16)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth straw];
    end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,15:16)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,15:16)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,15:16)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,15:16)=[water strcloth];
    end
elseif (count_08==count_10)

```



```

count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
    for i=1601:1800 % number of distance calculated to be classified 5 for each 24 vectors equals 120
        count_times=count_times+1; % thi is to count ech 5 times to classify
        if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
            &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
            &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
            &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
            &&(eu_x01(i,j)>eu_x10(i,j)))
            count_01=count_01+1;
            classif(i,j*2-1:j*2)=[cork 0];
        elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
            count_02=count_02+1;

```

```

classif(i,j*2-1;j*2)=[french 0];
elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
&&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
&&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
&&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
&&(eu_x03(i,j)>eu_x10(i,j)))
count_03=count_03+1;
classif(i,j*2-1;j*2)=[liz 0];
elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
&&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
&&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
&&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
&&(eu_x04(i,j)>eu_x10(i,j)))
count_04=count_04+1;
classif(i,j*2-1;j*2)=[net 0];
elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
&&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
&&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
&&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
&&(eu_x05(i,j)>eu_x10(i,j)))
count_05=count_05+1;
classif(i,j*2-1;j*2)=[repskin 0];
elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
&&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...
&&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
&&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
&&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1;j*2)=[scloth 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
&&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
&&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
&&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
&&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1;j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
&&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
&&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
&&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
&&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1;j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
&&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
&&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
&&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
&&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1;j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
&&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...

```

```

    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
    count_10=count_10+1;
    classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand; % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork net];
    else
        count_04=count_04+1;
        classif(i,j*2-1:j*2)=[net cork];
    end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;

```

```

if random_pick>.5
    count_01=count_01+1;
    classif(i,j*2-1:j*2)=[cork straw];
else
    count_07=count_07+1;
    classif(i,j*2-1:j*2)=[straw cork];
end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,17:18)=[french 0];
    end
end

```

```

elseif ((count_03>count_02)&&(count_03>count_01)...
&&(count_03>count_04)&&(count_03>count_05)...
&&(count_03>count_06)&&(count_03>count_07)...
&&(count_03>count_08)&&(count_03>count_09)...
&&(count_03>count_10))
count_k03=count_k03+1;
results(i_count,17:18)=[liz 0];
elseif ((count_04>count_02)&&(count_04>count_03)...
&&(count_04>count_01)&&(count_04>count_05)...
&&(count_04>count_06)&&(count_04>count_07)...
&&(count_04>count_08)&&(count_04>count_09)...
&&(count_04>count_10))
count_k04=count_k04+1;
results(i_count,17:18)=[net 0];
elseif ((count_05>count_02)&&(count_05>count_03)...
&&(count_05>count_04)&&(count_05>count_01)...
&&(count_05>count_06)&&(count_05>count_07)...
&&(count_05>count_08)&&(count_05>count_09)...
&&(count_05>count_10))
count_k05=count_k05+1;
results(i_count,17:18)=[repskin 0];
elseif ((count_06>count_02)&&(count_06>count_03)...
&&(count_06>count_04)&&(count_06>count_05)...
&&(count_06>count_01)&&(count_06>count_07)...
&&(count_06>count_08)&&(count_06>count_09)...
&&(count_06>count_10))
count_k06=count_k06+1;
results(i_count,17:18)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
&&(count_07>count_04)&&(count_07>count_05)...
&&(count_07>count_06)&&(count_07>count_01)...
&&(count_07>count_08)&&(count_07>count_09)...
&&(count_07>count_10))
count_k07=count_k07+1;
results(i_count,17:18)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
&&(count_08>count_04)&&(count_08>count_05)...
&&(count_08>count_06)&&(count_08>count_07)...
&&(count_08>count_01)&&(count_08>count_09)...
&&(count_08>count_10))
count_k08=count_k08+1;
results(i_count,17:18)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
&&(count_09>count_04)&&(count_09>count_05)...
&&(count_09>count_06)&&(count_09>count_07)...
&&(count_09>count_08)&&(count_09>count_01)...
&&(count_09>count_10))
count_k09=count_k09+1;
results(i_count,17:18)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
&&(count_10>count_04)&&(count_10>count_05)...
&&(count_10>count_06)&&(count_10>count_07)...

```

```

&&(count_10>count_08)&&(count_10>count_09)...
&&(count_10>count_01))
count_k10=count_k10+1;
results(i_count,17:18)=[weave 0];
elseif (count_01==count_02)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork french];
else
count_k02=count_k02+1;
results(i_count,17:18)=[french cork];
end
elseif (count_01==count_03)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork liz];
else
count_k03=count_k03+1;
results(i_count,17:18)=[liz cork];
end
elseif (count_01==count_04)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork net];
else
count_k04=count_k04+1;
results(i_count,17:18)=[net cork];
end
elseif (count_01==count_05)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork repskin];
else
count_k05=count_k05+1;
results(i_count,17:18)=[repskin cork];
end
elseif (count_01==count_06)
random_pick=rand;
if random_pick>.5
count_k01=count_k01+1;
results(i_count,17:18)=[cork scloth];
else
count_k06=count_k06+1;
results(i_count,17:18)=[scloth cork];
end
elseif (count_01==count_07)
random_pick=rand;
if random_pick>.5

```

```

        count_k01=count_k01+1;
        results(i_count,17:18)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water cork];
    end
elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,17:18)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,17:18)=[net french];
    end
elseif (count_02==count_05)

```

```

random_pick=rand;
if random_pick>.5
    count_k02=count_k02+1;
    results(i_count,17:18)=[french repskin];
else
    count_k05=count_k05+1;
    results(i_count,17:18)=[repskin french];
end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,17:18)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave french];
    end

```

```

end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,17:18)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth liz];
    end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw liz];
    end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth liz];
    end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz water];
    else

```

```

        count_k09=count_k09+1;
        results(i_count,17:18)=[water liz];
    end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,17:18)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave liz];
    end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin net];
    end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;

```

```

        results(i_count,17:18)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,17:18)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth repskin];
    end
elseif (count_05==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw repskin];
    end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,17:18)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;

```

```

if random_pick>.5
    count_k05=count_k05+1;
    results(i_count,17:18)=[repskin weave];
else
    count_k10=count_k10+1;
    results(i_count,17:18)=[weave repskin];
end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth scloth];
    end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water scloth];
    end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,17:18)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave scloth];
    end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth straw];
    end
end

```

```

elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water straw];
    end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,17:18)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave straw];
    end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth water];
    else
        count_k09=count_k09+1;
        results(i_count,17:18)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,17:18)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,17:18)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,17:18)=[weave water];
    end
end
end
end
end
    % results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Water\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');

```

```

fprintf('Classification results water/cork=%4.0f\n',count_k01);
fprintf('Classification results water/french=%4.0f\n',count_k02);
fprintf('Classification results water/liz=%4.0f\n',count_k03);
fprintf('Classification results water/net=%4.0f\n',count_k04);
fprintf('Classification results water/repskin=%4.0f\n',count_k05);
fprintf('Classification results water/scloth=%4.0f\n',count_k06);
fprintf('Classification results water/straw=%4.0f\n',count_k07);
fprintf('Classification results water/strcloth=%4.0f\n',count_k08);
fprintf('Classification results water/water=%4.0f\n',count_k09);
fprintf('Classification results water/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(9,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10];
perc(1,9)=count_k09*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,9));
%%%%%%%%%%
%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

count_k01=0;
count_k02=0;
count_k03=0;
count_k04=0;
count_k05=0;
count_k06=0;
count_k07=0;
count_k08=0;
count_k09=0;
count_k10=0;
% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;

```

```

weave=10;
random_selection=11;
count_times=0;
i_count=0;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of eu_x01 be compared with every single eu_x we
% must change it for example if is french class it should be eu_02 instead
% of eu_01.
for j=1:(N-5) % number of training samples
  for i=1801:2000 % number of distance calculated to be classified 5 for each 24 vectors equals 120
    count_times=count_times+1; % thi is to count ech 5 times to classify
    if ((eu_x01(i,j)>eu_x02(i,j))&&(eu_x01(i,j)>eu_x03(i,j))...
        &&(eu_x01(i,j)>eu_x04(i,j))&&(eu_x01(i,j)>eu_x05(i,j))...
        &&(eu_x01(i,j)>eu_x06(i,j))&&(eu_x01(i,j)>eu_x07(i,j))...
        &&(eu_x01(i,j)>eu_x08(i,j))&&(eu_x01(i,j)>eu_x09(i,j))...
        &&(eu_x01(i,j)>eu_x10(i,j)))
      count_01=count_01+1;
      classif(i,j*2-1:j*2)=[cork 0];
    elseif ((eu_x02(i,j)>eu_x01(i,j))&&(eu_x02(i,j)>eu_x03(i,j))...
            &&(eu_x02(i,j)>eu_x04(i,j))&&(eu_x02(i,j)>eu_x05(i,j))...
            &&(eu_x02(i,j)>eu_x06(i,j))&&(eu_x02(i,j)>eu_x07(i,j))...
            &&(eu_x02(i,j)>eu_x08(i,j))&&(eu_x02(i,j)>eu_x09(i,j))...
            &&(eu_x02(i,j)>eu_x10(i,j)))
      count_02=count_02+1;
      classif(i,j*2-1:j*2)=[french 0];
    elseif ((eu_x03(i,j)>eu_x02(i,j))&&(eu_x03(i,j)>eu_x01(i,j))...
            &&(eu_x03(i,j)>eu_x04(i,j))&&(eu_x03(i,j)>eu_x05(i,j))...
            &&(eu_x03(i,j)>eu_x06(i,j))&&(eu_x03(i,j)>eu_x07(i,j))...
            &&(eu_x03(i,j)>eu_x08(i,j))&&(eu_x03(i,j)>eu_x09(i,j))...
            &&(eu_x03(i,j)>eu_x10(i,j)))
      count_03=count_03+1;
      classif(i,j*2-1:j*2)=[liz 0];
    elseif ((eu_x04(i,j)>eu_x02(i,j))&&(eu_x04(i,j)>eu_x03(i,j))...
            &&(eu_x04(i,j)>eu_x01(i,j))&&(eu_x04(i,j)>eu_x05(i,j))...
            &&(eu_x04(i,j)>eu_x06(i,j))&&(eu_x04(i,j)>eu_x07(i,j))...
            &&(eu_x04(i,j)>eu_x08(i,j))&&(eu_x04(i,j)>eu_x09(i,j))...
            &&(eu_x04(i,j)>eu_x10(i,j)))
      count_04=count_04+1;
      classif(i,j*2-1:j*2)=[net 0];
    elseif ((eu_x05(i,j)>eu_x02(i,j))&&(eu_x05(i,j)>eu_x03(i,j))...
            &&(eu_x05(i,j)>eu_x04(i,j))&&(eu_x05(i,j)>eu_x01(i,j))...
            &&(eu_x05(i,j)>eu_x06(i,j))&&(eu_x05(i,j)>eu_x07(i,j))...
            &&(eu_x05(i,j)>eu_x08(i,j))&&(eu_x05(i,j)>eu_x09(i,j))...
            &&(eu_x05(i,j)>eu_x10(i,j)))
      count_05=count_05+1;
      classif(i,j*2-1:j*2)=[repskin 0];
    elseif ((eu_x06(i,j)>eu_x02(i,j))&&(eu_x06(i,j)>eu_x03(i,j))...
            &&(eu_x06(i,j)>eu_x04(i,j))&&(eu_x06(i,j)>eu_x05(i,j))...

```

```

    &&(eu_x06(i,j)>eu_x01(i,j))&&(eu_x06(i,j)>eu_x07(i,j))...
    &&(eu_x06(i,j)>eu_x08(i,j))&&(eu_x06(i,j)>eu_x09(i,j))...
    &&(eu_x06(i,j)>eu_x10(i,j)))
count_06=count_06+1;
classif(i,j*2-1:j*2)=[sclotch 0];
elseif ((eu_x07(i,j)>eu_x02(i,j))&&(eu_x07(i,j)>eu_x03(i,j))...
    &&(eu_x07(i,j)>eu_x04(i,j))&&(eu_x07(i,j)>eu_x05(i,j))...
    &&(eu_x07(i,j)>eu_x06(i,j))&&(eu_x07(i,j)>eu_x01(i,j))...
    &&(eu_x07(i,j)>eu_x08(i,j))&&(eu_x07(i,j)>eu_x09(i,j))...
    &&(eu_x07(i,j)>eu_x10(i,j)))
count_07=count_07+1;
classif(i,j*2-1:j*2)=[straw 0];
elseif ((eu_x08(i,j)>eu_x02(i,j))&&(eu_x08(i,j)>eu_x03(i,j))...
    &&(eu_x08(i,j)>eu_x04(i,j))&&(eu_x08(i,j)>eu_x05(i,j))...
    &&(eu_x08(i,j)>eu_x06(i,j))&&(eu_x08(i,j)>eu_x07(i,j))...
    &&(eu_x08(i,j)>eu_x01(i,j))&&(eu_x08(i,j)>eu_x09(i,j))...
    &&(eu_x08(i,j)>eu_x10(i,j)))
count_08=count_08+1;
classif(i,j*2-1:j*2)=[strcloth 0];
elseif ((eu_x09(i,j)>eu_x02(i,j))&&(eu_x09(i,j)>eu_x03(i,j))...
    &&(eu_x09(i,j)>eu_x04(i,j))&&(eu_x09(i,j)>eu_x05(i,j))...
    &&(eu_x09(i,j)>eu_x06(i,j))&&(eu_x09(i,j)>eu_x07(i,j))...
    &&(eu_x09(i,j)>eu_x08(i,j))&&(eu_x09(i,j)>eu_x01(i,j))...
    &&(eu_x09(i,j)>eu_x10(i,j)))
count_09=count_09+1;
classif(i,j*2-1:j*2)=[water 0];
elseif ((eu_x10(i,j)>eu_x02(i,j))&&(eu_x10(i,j)>eu_x03(i,j))...
    &&(eu_x10(i,j)>eu_x04(i,j))&&(eu_x10(i,j)>eu_x05(i,j))...
    &&(eu_x10(i,j)>eu_x06(i,j))&&(eu_x10(i,j)>eu_x07(i,j))...
    &&(eu_x10(i,j)>eu_x08(i,j))&&(eu_x10(i,j)>eu_x09(i,j))...
    &&(eu_x10(i,j)>eu_x01(i,j)))
count_10=count_10+1;
classif(i,j*2-1:j*2)=[weave 0];
elseif (eu_x01(i,j)==eu_x02(i,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork french];
    else
        count_02=count_02+1;
        classif(i,j*2-1:j*2)=[french cork];
    end
elseif (eu_x01(i,j)==eu_x03(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork liz];
    else
        count_03=count_03+1;
        classif(i,j*2-1:j*2)=[liz cork];
    end
elseif (eu_x01(i,j)==eu_x04(i,j))

```

```

random_pick=rand;
if random_pick>.5
    count_01=count_01+1;
    classif(i,j*2-1:j*2)=[cork net];
else
    count_04=count_04+1;
    classif(i,j*2-1:j*2)=[net cork];
end
elseif (eu_x01(i,j)==eu_x05(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork repskin];
    else
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin cork];
    end
elseif (eu_x01(i,j)==eu_x06(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork scloth];
    else
        count_06=count_06+1;
        classif(i,j*2-1:j*2)=[scloth cork];
    end
elseif (eu_x01(i,j)==eu_x07(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork straw];
    else
        count_07=count_07+1;
        classif(i,j*2-1:j*2)=[straw cork];
    end
elseif (eu_x01(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth cork];
    end
elseif (eu_x01(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water cork];
    end

```

```

end
elseif (eu_x01(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(i,j*2-1:j*2)=[cork weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave cork];
    end
end
end
if (count_times==5) % Classification of K Nearest Neighborhood
    count_times=0;
    i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03)...
        &&(count_01>count_04)&&(count_01>count_05)...
        &&(count_01>count_06)&&(count_01>count_07)...
        &&(count_01>count_08)&&(count_01>count_09)...
        &&(count_01>count_10))
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork 0];
    elseif ((count_02>count_01)&&(count_02>count_03)...
        &&(count_02>count_04)&&(count_02>count_05)...
        &&(count_02>count_06)&&(count_02>count_07)...
        &&(count_02>count_08)&&(count_02>count_09)...
        &&(count_02>count_10))
        count_k02=count_k02+1;
        results(i_count,19:20)=[french 0];
    elseif ((count_03>count_02)&&(count_03>count_01)...
        &&(count_03>count_04)&&(count_03>count_05)...
        &&(count_03>count_06)&&(count_03>count_07)...
        &&(count_03>count_08)&&(count_03>count_09)...
        &&(count_03>count_10))
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz 0];
    elseif ((count_04>count_02)&&(count_04>count_03)...
        &&(count_04>count_01)&&(count_04>count_05)...
        &&(count_04>count_06)&&(count_04>count_07)...
        &&(count_04>count_08)&&(count_04>count_09)...
        &&(count_04>count_10))
        count_k04=count_k04+1;
        results(i_count,19:20)=[net 0];
    elseif ((count_05>count_02)&&(count_05>count_03)...
        &&(count_05>count_04)&&(count_05>count_01)...
        &&(count_05>count_06)&&(count_05>count_07)...
        &&(count_05>count_08)&&(count_05>count_09)...
        &&(count_05>count_10))
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin 0];
    elseif ((count_06>count_02)&&(count_06>count_03)...
        &&(count_06>count_04)&&(count_06>count_05)...
        &&(count_06>count_01)&&(count_06>count_07)...

```

```

    &&(count_06>count_08)&&(count_06>count_09)...
    &&(count_06>count_10))
    count_k06=count_k06+1;
    results(i_count,19:20)=[scloth 0];
elseif ((count_07>count_02)&&(count_07>count_03)...
    &&(count_07>count_04)&&(count_07>count_05)...
    &&(count_07>count_06)&&(count_07>count_01)...
    &&(count_07>count_08)&&(count_07>count_09)...
    &&(count_07>count_10))
    count_k07=count_k07+1;
    results(i_count,19:20)=[straw 0];
elseif ((count_08>count_02)&&(count_08>count_03)...
    &&(count_08>count_04)&&(count_08>count_05)...
    &&(count_08>count_06)&&(count_08>count_07)...
    &&(count_08>count_01)&&(count_08>count_09)...
    &&(count_08>count_10))
    count_k08=count_k08+1;
    results(i_count,19:20)=[strcloth 0];
elseif ((count_09>count_02)&&(count_09>count_03)...
    &&(count_09>count_04)&&(count_09>count_05)...
    &&(count_09>count_06)&&(count_09>count_07)...
    &&(count_09>count_08)&&(count_09>count_01)...
    &&(count_09>count_10))
    count_k09=count_k09+1;
    results(i_count,19:20)=[water 0];
elseif ((count_10>count_02)&&(count_10>count_03)...
    &&(count_10>count_04)&&(count_10>count_05)...
    &&(count_10>count_06)&&(count_10>count_07)...
    &&(count_10>count_08)&&(count_10>count_09)...
    &&(count_10>count_01))
    count_k10=count_k10+1;
    results(i_count,19:20)=[weave 0];
elseif (count_01==count_02)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork french];
    else
        count_k02=count_k02+1;
        results(i_count,19:20)=[french cork];
    end
elseif (count_01==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork liz];
    else
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz cork];
    end
elseif (count_01==count_04)
    random_pick=rand;

```

```

if random_pick>.5
    count_k01=count_k01+1;
    results(i_count,19:20)=[cork net];
else
    count_k04=count_k04+1;
    results(i_count,19:20)=[net cork];
end
elseif (count_01==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin cork];
    end
elseif (count_01==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth cork];
    end
elseif (count_01==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw cork];
    end
elseif (count_01==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth cork];
    end
elseif (count_01==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water cork];
    end
end

```

```

elseif (count_01==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k01=count_k01+1;
        results(i_count,19:20)=[cork weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave cork];
    end
elseif (count_02==count_03)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french liz];
    else
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz french];
    end
elseif (count_02==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french net];
    else
        count_k04=count_k04+1;
        results(i_count,19:20)=[net french];
    end
elseif (count_02==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin french];
    end
elseif (count_02==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth french];
    end
elseif (count_02==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french straw];
    else
        count_k07=count_k07+1;

```

```

        results(i_count,19:20)=[straw french];
    end
elseif (count_02==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth french];
    end
elseif (count_02==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water french];
    end
elseif (count_02==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k02=count_k02+1;
        results(i_count,19:20)=[french weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave french];
    end
elseif (count_03==count_04)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz net];
    else
        count_k04=count_k04+1;
        results(i_count,19:20)=[net liz];
    end
elseif (count_03==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin liz];
    end
elseif (count_03==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz scloth];

```

```

else
    count_k06=count_k06+1;
    results(i_count,19:20)=[scloth liz];
end
elseif (count_03==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw liz];
    end
end
elseif (count_03==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth liz];
    end
end
elseif (count_03==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water liz];
    end
end
elseif (count_03==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k03=count_k03+1;
        results(i_count,19:20)=[liz weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave liz];
    end
end
elseif (count_04==count_05)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net repskin];
    else
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin net];
    end
end
elseif (count_04==count_06)
    random_pick=rand;
    if random_pick>.5

```

```

        count_k04=count_k04+1;
        results(i_count,19:20)=[net scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth net];
    end
elseif (count_04==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw net];
    end
elseif (count_04==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth net];
    end
elseif (count_04==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water net];
    end
elseif (count_04==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k04=count_k04+1;
        results(i_count,19:20)=[net weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave net];
    end
elseif (count_05==count_06)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin scloth];
    else
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth repskin];
    end
elseif (count_05==count_07)

```

```

random_pick=rand;
if random_pick>.5
    count_k05=count_k05+1;
    results(i_count,19:20)=[repskin straw];
else
    count_k07=count_k07+1;
    results(i_count,19:20)=[straw repskin];
end
elseif (count_05==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth repskin];
    end
elseif (count_05==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water repskin];
    end
elseif (count_05==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k05=count_k05+1;
        results(i_count,19:20)=[repskin weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave repskin];
    end
elseif (count_06==count_07)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth straw];
    else
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw scloth];
    end
elseif (count_06==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth scloth];
    end

```

```

end
elseif (count_06==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water scloth];
    end
end
elseif (count_06==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k06=count_k06+1;
        results(i_count,19:20)=[scloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave scloth];
    end
end
elseif (count_07==count_08)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw strcloth];
    else
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth straw];
    end
end
elseif (count_07==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw water];
    else
        count_k09=count_k09+1;
        results(i_count,19:20)=[water straw];
    end
end
elseif (count_07==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k07=count_k07+1;
        results(i_count,19:20)=[straw weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave straw];
    end
end
elseif (count_08==count_09)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth water];
    else

```

```

        count_k09=count_k09+1;
        results(i_count,19:20)=[water strcloth];
    end
elseif (count_08==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k08=count_k08+1;
        results(i_count,19:20)=[strcloth weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave strcloth];
    end
elseif (count_09==count_10)
    random_pick=rand;
    if random_pick>.5
        count_k09=count_k09+1;
        results(i_count,19:20)=[water weave];
    else
        count_k10=count_k10+1;
        results(i_count,19:20)=[weave water];
    end
end
end
end
end
    % results(i*10-9:i*10,:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Training Samples for Weave\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results weave/cork=%4.0f\n',count_k01);
fprintf('Classification results weave/french=%4.0f\n',count_k02);
fprintf('Classification results weave/liz=%4.0f\n',count_k03);
fprintf('Classification results weave/net=%4.0f\n',count_k04);
fprintf('Classification results weave/repskin=%4.0f\n',count_k05);
fprintf('Classification results weave/scloth=%4.0f\n',count_k06);
fprintf('Classification results weave/straw=%4.0f\n',count_k07);
fprintf('Classification results weave/strcloth=%4.0f\n',count_k08);
fprintf('Classification results weave/water=%4.0f\n',count_k09);
fprintf('Classification results weave/weave=%4.0f\n',count_k10);
% Percentaje of classification
a=count_k01+count_k02+count_k03+count_k04+count_k05+count_k06+count_k07+count_k08+count_k09+count_k10;
fprintf('\nTotal samples=%4.0f\n',a);
perc(1,10)=count_k10*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,10));
rec(10,:)=count_k01 count_k02 count_k03 count_k04 count_k05 count_k06 count_k07 count_k08 count_k09 count_k10;
rec_perc=[rec perc];
toc

```

APPENDIX C4 MAXIMUM LIKELIHOOD CLASSIFIER

%%%

```

% Maximum Likelihood Classifier %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% All four Coefficient Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
tic
% loading the matrices of training & testing samples
load training_testing_rand_db1_n.mat
% load training_testing_rand_haar.mat
% load training_testing_rand_bior11.mat
% load training_testing_rand_rbio11.mat
warning off
[M N]=size(test_all);
% warning off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X_DATA Classification %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% By changing "x_data=..." we can classify any image can be training or
% testing, so this will wor for any type of data

% Discriminant function for x_data 01_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix (for testing the classifier we will use 24
        for the training samples)
            for k=1:(N-5) % M by N-5 coefficient matrix
                x_data=test_all(j*4-3:j*4,5+k); % by changing x_data we will classify different images
                mean_01=train_cork(i*4-3:i*4,1);
                cov_01=train_cork(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
                ml_01(i,k)=hii_fcn(x_data,mean_01,cov_01); % this saves for each coefficient matrix all the
                distances between all the features of the other classes
            end
            ml_01=-sort(-ml_01); % We first sort the columns in order to get the minimum distances for each
            vector
            end
            ml_x01(j*5-4:j*5,:)=ml_01(1:5,:); % So the minimum distance for the first coefficient matrix is the
            first row
        end
    end
% Discriminant function for x_data 02_data
for j=1:400 % 400 is the length of the testing samples matrix
    for i=1:24 % 24 is the length of the training samples matrix
        for k=1:(N-5) % M by N-5 coefficient matrix
            x_data=test_all(j*4-3:j*4,5+k);
            mean_02=train_french(i*4-3:i*4,1);
            cov_02=train_french(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
            ml_02(i,k)=hii_fcn(x_data,mean_02,cov_02); % this saves for each coefficient matrix all the
            distances between all the features of the other classes
        end
        ml_02=-sort(-ml_02); % We first sort the columns in order to get the minimum distances for each
        vector
    end
end

```

```

end
ml_x02(j*5-4:j*5,:)=ml_02(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 03_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_03=train_liz(i*4-3:i*4,1);
cov_03=train_liz(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_03(i,k)=hii_fcn(x_data,mean_03,cov_03); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_03=-sort(-ml_03); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x03(j*5-4:j*5,:)=ml_03(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 04_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_04=train_net(i*4-3:i*4,1);
cov_04=train_net(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_04(i,k)=hii_fcn(x_data,mean_04,cov_04); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_04=-sort(-ml_04); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x04(j*5-4:j*5,:)=ml_04(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 05_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_05=train_repskin(i*4-3:i*4,1);
cov_05=train_repskin(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_05(i,k)=hii_fcn(x_data,mean_05,cov_05); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_05=-sort(-ml_05); % We first sort the columns in order to get the minimum distances for each
vector
end

```

```

ml_x05(j*5-4:j*5,:)=ml_05(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 06_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_06=train_scloth(i*4-3:i*4,1);
cov_06=train_scloth(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_06(i,k)=hii_fcn(x_data,mean_06,cov_06); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_06=-sort(-ml_06); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x06(j*5-4:j*5,:)=ml_06(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 07_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_07=train_straw(i*4-3:i*4,1);
cov_07=train_straw(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_07(i,k)=hii_fcn(x_data,mean_07,cov_07); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_07=-sort(-ml_07); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x07(j*5-4:j*5,:)=ml_07(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 08_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_08=train_strcloth(i*4-3:i*4,1);
cov_08=train_strcloth(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_08(i,k)=hii_fcn(x_data,mean_08,cov_08); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_08=-sort(-ml_08); % We first sort the columns in order to get the minimum distances for each
vector
end

```

```

ml_x08(j*5-4:j*5,:)=ml_08(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 09_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_09=train_water(i*4-3:i*4,1);
cov_09=train_water(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_09(i,k)=hii_fcn(x_data,mean_09,cov_09); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_09=-sort(-ml_09); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x09(j*5-4:j*5,:)=ml_09(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

% Discriminant function for x_data 10_data
for j=1:400 % 400 is the length of the testing samples matrix
for i=1:24 % 24 is the length of the training samples matrix
for k=1:(N-5) % M by N-5 coefficient matrix
x_data=test_all(j*4-3:j*4,5+k);
mean_10=train_weave(i*4-3:i*4,1);
cov_10=train_weave(i*4-3:i*4,2:5); % if using ML classifier as the distance measure
ml_10(i,k)=hii_fcn(x_data,mean_10,cov_10); % this saves for each coefficient matrix all the
distances between all the features of the other classes
end
ml_10=-sort(-ml_10); % We first sort the columns in order to get the minimum distances for each
vector
end
ml_x10(j*5-4:j*5,:)=ml_10(1:5,:); % So the minimum distance for the first coefficient matrix is the first
row
end

```

```

% Classification process %

```

```

count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

```

```

% Data classified

```

```

cork=1;
french=2;
liz=3;
net=4;
repskin=5;
sclloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=1:40 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
            count_01=count_01+1;
            classif(j,:)=["cork 0"];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=["liz 0"];
        elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
            count_04=count_04+1;
            classif(j,:)=["net 0"];
        elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
            count_05=count_05+1;
            classif(j,:)=["straw 0"];
        else
            count_06=count_06+1;
            classif(j,:)=["strcloth 0"];
        end
    end
end

```

```

        &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
    count_05=count_05+1;
    classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
    count_06=count_06+1;
    classif(j,:)=['sclotch 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
    count_07=count_07+1;
    classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
    count_08=count_08+1;
    classif(j,:)=['strelotch 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
    count_10=count_10+1;
    classif(j,:)=['weave 0'];
elseif (ml_x01(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand; % We use rand to let random selection when
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;

```

```

        classif(j,:)=['cork liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=['cork stcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['stcloth cork'];
    end
elseif (ml_x01(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;

```

```

    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=[cork water];
    else
        count_09=count_09+1;
        classif(j,:)=[water cork];
    end
elseif (ml_x01(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_01=count_01+1;
        classif(j,:)=[cork weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave cork];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf("\n\nResults for Approximation Coefficients of Testing Samples for Cork\n\n");
fprintf("Results will be expressed as: ('original class'/'class classified as')\n");
fprintf('Classification results cork/cork=%4.0f\n',count_01);
fprintf('Classification results cork/french=%4.0f\n',count_02);
fprintf('Classification results cork/liz=%4.0f\n',count_03);
fprintf('Classification results cork/net=%4.0f\n',count_04);
fprintf('Classification results cork/repskin=%4.0f\n',count_05);
fprintf('Classification results cork/scloth=%4.0f\n',count_06);
fprintf('Classification results cork/straw=%4.0f\n',count_07);
fprintf('Classification results cork/strcloth=%4.0f\n',count_08);
fprintf('Classification results cork/water=%4.0f\n',count_09);
fprintf('Classification results cork/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf("\nTotal samples=%4.0f\n",a);
rec(1,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,1)=count_01*100/a;
fprintf("\nPercentaje of classification=%4.4f\n",perc(1,1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;

```

```

count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=41:80 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=['french 0'];
        elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
            count_01=count_01+1;
            classif(j,:)=['cork 0'];
        elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=['liz 0'];
        elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
            count_04=count_04+1;
            classif(j,:)=['net 0'];
    end
end

```

```

elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strecloth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (ml_x02(i*5-4,j)==ml_x01(i*5-4,j))
random_pick=rand; % We use rand to let random selection when
if random_pick>.5
count_02=count_02+1;
classif(j,:)=['french cork'];
else
count_01=count_01+1;
classif(j,:)=['cork french'];
end
elseif (ml_x02(i*5-4,j)==ml_x03(i*5-4,j))

```

```

random_pick=rand;
if random_pick>.5
    count_02=count_02+1;
    classif(j,:)=[french liz];
else
    count_03=count_03+1;
    classif(j,:)=[liz french];
end
elseif (ml_x02(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french net];
    else
        count_04=count_04+1;
        classif(j,:)=[net french];
    end
elseif (ml_x02(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french repskin];
    else
        count_05=count_05+1;
        classif(j,:)=[repskin french];
    end
elseif (ml_x02(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french scloth];
    else
        count_06=count_06+1;
        classif(j,:)=[scloth french];
    end
elseif (ml_x02(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw french];
    end
elseif (ml_x02(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french strcloth];
    else
        count_08=count_08+1;
        classif(j,:)=[strcloth french];
    end

```

```

end
elseif (ml_x02(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french water];
    else
        count_09=count_09+1;
        classif(j,:)=[water french];
    end
elseif (ml_x02(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_02=count_02+1;
        classif(j,:)=[french weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave french];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for French\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results french/cork=%4.0f\n',count_01);
fprintf('Classification results french/french=%4.0f\n',count_02);
fprintf('Classification results french/liz=%4.0f\n',count_03);
fprintf('Classification results french/net=%4.0f\n',count_04);
fprintf('Classification results french/repskin=%4.0f\n',count_05);
fprintf('Classification results french/scloth=%4.0f\n',count_06);
fprintf('Classification results french/straw=%4.0f\n',count_07);
fprintf('Classification results french/streloth=%4.0f\n',count_08);
fprintf('Classification results french/water=%4.0f\n',count_09);
fprintf('Classification results french/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(2,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,2)=count_02*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,2));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;

```

```

count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=81:120 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=["liz 0"];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
            count_01=count_01+1;
            classif(j,:)=["cork 0"];
        elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))

```

```

        &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
    count_04=count_04+1;
    classif(j,:)=['net 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
    count_05=count_05+1;
    classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
    count_06=count_06+1;
    classif(j,:)=['sclotch 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
    count_07=count_07+1;
    classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
    count_08=count_08+1;
    classif(j,:)=['strcloth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
    count_10=count_10+1;
    classif(j,:)=['weave 0'];
elseif (ml_x03(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand; % We use rand to let random selection when
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz french'];
    else
        count_02=count_02+1;

```

```

        classif(j,:)=['french liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz cork'];
    else
        count_01=count_01+1;
        classif(j,:)=['cork liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz scloth'];
    else
        count_06=count_06+1;
        classif(j,:)=['scloth liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz straw'];
    else
        count_07=count_07+1;
        classif(j,:)=['straw liz'];
    end
elseif (ml_x03(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=['liz strcloth'];
    end

```

```

else
    count_08=count_08+1;
    classif(j,:)=[strcloth liz];
end
elseif (ml_x03(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=[liz water];
    else
        count_09=count_09+1;
        classif(j,:)=[water liz];
    end
elseif (ml_x03(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_03=count_03+1;
        classif(j,:)=[liz weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave liz];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Liz\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results liz/cork=%4.0f\n',count_01);
fprintf('Classification results liz/french=%4.0f\n',count_02);
fprintf('Classification results liz/liz=%4.0f\n',count_03);
fprintf('Classification results liz/net=%4.0f\n',count_04);
fprintf('Classification results liz/repskin=%4.0f\n',count_05);
fprintf('Classification results liz/scloth=%4.0f\n',count_06);
fprintf('Classification results liz/straw=%4.0f\n',count_07);
fprintf('Classification results liz/strcloth=%4.0f\n',count_08);
fprintf('Classification results liz/water=%4.0f\n',count_09);
fprintf('Classification results liz/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(3,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,3)=count_03*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,3));

%%%%%%%%%%
%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;

```

```

count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=121:160 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
            count_04=count_04+1;
            classif(j,:)=["net 0"];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=["french 0"];
        elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=["liz 0"];
        end
    end
end

```

```

elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['sclotch 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strelotch 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (ml_x04(i*5-4,j)==ml_x02(i*5-4,j))
random_pick=rand; % We use rand to let random selection when
if random_pick>.5

```

```

        count_04=count_04+1;
        classif(j,:)=[net french];
    else
        count_02=count_02+1;
        classif(j,:)=[french net];
    end
elseif (ml_x04(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net liz];
    else
        count_03=count_03+1;
        classif(j,:)=[liz net];
    end
elseif (ml_x04(i*5-4,j)==ml_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork net];
    end
elseif (ml_x04(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net repskin];
    else
        count_05=count_05+1;
        classif(j,:)=[repskin net];
    end
elseif (ml_x04(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net scloth];
    else
        count_06=count_06+1;
        classif(j,:)=[scloth net];
    end
elseif (ml_x04(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw net];
    end
elseif (ml_x04(i*5-4,j)==ml_x08(i*5-4,j))

```

```

    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net strcloth];
    else
        count_08=count_08+1;
        classif(j,:)=[strcloth net];
    end
elseif (ml_x04(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net water];
    else
        count_09=count_09+1;
        classif(j,:)=[water net];
    end
elseif (ml_x04(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_04=count_04+1;
        classif(j,:)=[net weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave net];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Net\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results net/cork=%4.0f\n',count_01);
fprintf('Classification results net/french=%4.0f\n',count_02);
fprintf('Classification results net/liz=%4.0f\n',count_03);
fprintf('Classification results net/net=%4.0f\n',count_04);
fprintf('Classification results net/repskin=%4.0f\n',count_05);
fprintf('Classification results net/scloth=%4.0f\n',count_06);
fprintf('Classification results net/straw=%4.0f\n',count_07);
fprintf('Classification results net/strcloth=%4.0f\n',count_08);
fprintf('Classification results net/water=%4.0f\n',count_09);
fprintf('Classification results net/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(4,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,4)=count_04*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,4));

```

```

%%%%%%%%%
%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=161:200 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
            count_05=count_05+1;
            classif(j,:)=['repskin 0'];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=['french 0'];
        elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...

```

```

    &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['scloth 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strecloth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))

```

```

count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (eu_x05(i,j)==eu_x02(i,j))
random_pick=rand;          % We use rand to let random selection when
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin french];
else
count_02=count_02+1;
classif(i,j*2-1:j*2)=[french repskin];
end
elseif (eu_x05(i,j)==eu_x03(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin liz];
else
count_03=count_03+1;
classif(i,j*2-1:j*2)=[liz repskin];
end
elseif (eu_x05(i,j)==eu_x04(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin net];
else
count_04=count_04+1;
classif(i,j*2-1:j*2)=[net repskin];
end
elseif (eu_x05(i,j)==eu_x01(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin cork];
else
count_01=count_01+1;
classif(i,j*2-1:j*2)=[cork repskin];
end
elseif (eu_x05(i,j)==eu_x06(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin scloth];
else
count_06=count_06+1;
classif(i,j*2-1:j*2)=[scloth repskin];
end
elseif (eu_x05(i,j)==eu_x07(i,j))
random_pick=rand;
if random_pick>.5
count_05=count_05+1;
classif(i,j*2-1:j*2)=[repskin straw];

```

```

else
    count_07=count_07+1;
    classif(i,j*2-1:j*2)=[straw repskin];
end
elseif (eu_x05(i,j)==eu_x08(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin strcloth];
    else
        count_08=count_08+1;
        classif(i,j*2-1:j*2)=[strcloth repskin];
    end
elseif (eu_x05(i,j)==eu_x09(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin water];
    else
        count_09=count_09+1;
        classif(i,j*2-1:j*2)=[water repskin];
    end
elseif (eu_x05(i,j)==eu_x10(i,j))
    random_pick=rand;
    if random_pick>.5
        count_05=count_05+1;
        classif(i,j*2-1:j*2)=[repskin weave];
    else
        count_10=count_10+1;
        classif(i,j*2-1:j*2)=[weave repskin];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Repskin\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results repskin/cork=%4.0f\n',count_01);
fprintf('Classification results repskin/french=%4.0f\n',count_02);
fprintf('Classification results repskin/liz=%4.0f\n',count_03);
fprintf('Classification results repskin/net=%4.0f\n',count_04);
fprintf('Classification results repskin/repskin=%4.0f\n',count_05);
fprintf('Classification results repskin/scloth=%4.0f\n',count_06);
fprintf('Classification results repskin/straw=%4.0f\n',count_07);
fprintf('Classification results repskin/strcloth=%4.0f\n',count_08);
fprintf('Classification results repskin/water=%4.0f\n',count_09);
fprintf('Classification results repskin/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
0;
fprintf('\nTotal samples=%4.0f\n',a);

```

```

rec(5,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,5)=count_05*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,5));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=201:240 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
            count_06=count_06+1;
            classif(j,:)=scloth 0];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...

```

```

    &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
    count_02=count_02+1;
    classif(j,:)=['french 0'];
elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
    count_03=count_03+1;
    classif(j,:)=['liz 0'];
elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
    count_04=count_04+1;
    classif(j,:)=['net 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
    count_05=count_05+1;
    classif(j,:)=['repskin 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
    count_01=count_01+1;
    classif(j,:)=['cork 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
    count_07=count_07+1;
    classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
    count_08=count_08+1;
    classif(j,:)=['strecloth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];

```

```

elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
    count_10=count_10+1;
    classif(j,:)=['weave 0'];
elseif (ml_x06(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french scloth'];
    end
elseif (ml_x06(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz scloth'];
    end
elseif (ml_x06(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net scloth'];
    end
elseif (ml_x06(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin scloth'];
    end
elseif (ml_x06(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=['scloth cork'];
    else
        count_01=count_01+1;
        classif(j,:)=['cork scloth'];
    end
end

```

```

elseif (ml_x06(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw scloth];
    end
elseif (ml_x06(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth strcloth];
    else
        count_08=count_08+1;
        classif(j,:)=[strcloth scloth];
    end
elseif (ml_x06(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth water];
    else
        count_09=count_09+1;
        classif(j,:)=[water scloth];
    end
elseif (ml_x06(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_06=count_06+1;
        classif(j,:)=[scloth weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave scloth];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Scloth\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results scloth/cork=%4.0f\n',count_01);
fprintf('Classification results scloth/french=%4.0f\n',count_02);
fprintf('Classification results scloth/liz=%4.0f\n',count_03);
fprintf('Classification results scloth/net=%4.0f\n',count_04);
fprintf('Classification results scloth/repskin=%4.0f\n',count_05);
fprintf('Classification results scloth/scloth=%4.0f\n',count_06);
fprintf('Classification results scloth/straw=%4.0f\n',count_07);
fprintf('Classification results scloth/strcloth=%4.0f\n',count_08);
fprintf('Classification results scloth/water=%4.0f\n',count_09);
fprintf('Classification results scloth/weave=%4.0f\n',count_10);

```

```

% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(6,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,6)=count_06*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,6));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=241:280 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
            count_07=count_07+1;
            classif(j,:)=["straw 0"];
        end
    end
end

```

```

elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
count_02=count_02+1;
classif(j,:)=['french 0'];
elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
count_03=count_03+1;
classif(j,:)=['liz 0'];
elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
count_04=count_04+1;
classif(j,:)=['net 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
count_05=count_05+1;
classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
count_06=count_06+1;
classif(j,:)=['scloth 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
&&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strelth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
&&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...

```

```

        &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (ml_x07(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand;        % We use rand to let random selection when
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw scloth'];

```

```

else
    count_06=count_06+1;
    classif(j,:)=['scloth straw'];
end
elseif (ml_x07(i*5-4,j)==ml_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw cork'];
    else
        count_01=count_01+1;
        classif(j,:)=['cork straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw strcloth'];
    else
        count_08=count_08+1;
        classif(j,:)=['strcloth straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw water'];
    else
        count_09=count_09+1;
        classif(j,:)=['water straw'];
    end
elseif (ml_x07(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_07=count_07+1;
        classif(j,:)=['straw weave'];
    else
        count_10=count_10+1;
        classif(j,:)=['weave straw'];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Straw\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results straw/cork=%4.0f\n',count_01);
fprintf('Classification results straw/french=%4.0f\n',count_02);
fprintf('Classification results straw/liz=%4.0f\n',count_03);
fprintf('Classification results straw/net=%4.0f\n',count_04);
fprintf('Classification results straw/repskin=%4.0f\n',count_05);
fprintf('Classification results straw/scloth=%4.0f\n',count_06);

```

```

fprintf('Classification results straw/straw=%4.0f\n',count_07);
fprintf('Classification results straw/strcloth=%4.0f\n',count_08);
fprintf('Classification results straw/water=%4.0f\n',count_09);
fprintf('Classification results straw/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(7,:)=[count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,7)=count_07*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,7));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=281:320 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...

```

```

        &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
    count_08=count_08+1;
    classif(j,:)=['strecloth 0'];
elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
    count_02=count_02+1;
    classif(j,:)=['french 0'];
elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
    count_03=count_03+1;
    classif(j,:)=['liz 0'];
elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
    count_04=count_04+1;
    classif(j,:)=['net 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
    count_05=count_05+1;
    classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
    count_06=count_06+1;
    classif(j,:)=['scloth 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
    count_07=count_07+1;
    classif(j,:)=['straw 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
    count_01=count_01+1;

```

```

    classif(j,:)=['cork 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
    count_10=count_10+1;
    classif(j,:)=['weave 0'];
elseif (ml_x08(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand;      % We use rand to let random selection when
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french strcloth'];
    end
elseif (ml_x08(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz strcloth'];
    end
elseif (ml_x08(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net strcloth'];
    end
elseif (ml_x08(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=['strcloth repskin'];
    else
        count_05=count_05+1;
        classif(j,:)=['repskin strcloth'];
    end
elseif (ml_x08(i*5-4,j)==ml_x06(i*5-4,j))

```

```

random_pick=rand;
if random_pick>.5
    count_08=count_08+1;
    classif(j,:)=[strcloth scloth];
else
    count_06=count_06+1;
    classif(j,:)=[scloth strcloth];
end
elseif (ml_x08(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw strcloth];
    end
elseif (ml_x08(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork strcloth];
    end
elseif (ml_x08(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth water];
    else
        count_09=count_09+1;
        classif(j,:)=[water strcloth];
    end
elseif (ml_x08(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_08=count_08+1;
        classif(j,:)=[strcloth weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave strcloth];
    end
end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion
end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Strcloth\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results strcloth/cork=%4.0f\n',count_01);
fprintf('Classification results strcloth/french=%4.0f\n',count_02);

```

```

fprintf('Classification results strcloth/liz=%4.0f\n',count_03);
fprintf('Classification results strcloth/net=%4.0f\n',count_04);
fprintf('Classification results strcloth/repskin=%4.0f\n',count_05);
fprintf('Classification results strcloth/scloth=%4.0f\n',count_06);
fprintf('Classification results strcloth/straw=%4.0f\n',count_07);
fprintf('Classification results strcloth/strcloth=%4.0f\n',count_08);
fprintf('Classification results strcloth/water=%4.0f\n',count_09);
fprintf('Classification results strcloth/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(8,:)=count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09
count_10];
perc(1,8)=count_08*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,8));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.

```

```

for i=321:360 % number of training samples
for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
if ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
    count_09=count_09+1;
    classif(j,:)=['water 0'];
elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
    count_02=count_02+1;
    classif(j,:)=['french 0'];
elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
    count_03=count_03+1;
    classif(j,:)=['liz 0'];
elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
    count_04=count_04+1;
    classif(j,:)=['net 0'];
elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
    count_05=count_05+1;
    classif(j,:)=['repskin 0'];
elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
    count_06=count_06+1;
    classif(j,:)=['sclotch 0'];
elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
    &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
    count_07=count_07+1;
    classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...

```

```

        &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strecth 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
count_10=count_10+1;
classif(j,:)=['weave 0'];
elseif (ml_x09(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand;        % We use rand to let random selection when
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water french'];
    else
        count_02=count_02+1;
        classif(j,:)=['french water'];
    end
elseif (ml_x09(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water liz'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz water'];
    end
elseif (ml_x09(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water net'];
    else
        count_04=count_04+1;
        classif(j,:)=['net water'];
    end
elseif (ml_x09(i*5-4,j)==ml_x05(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=['water repskin'];
    end

```

```

else
    count_05=count_05+1;
    classif(j,:)=[repskin water];
end
elseif (ml_x09(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water scloth];
    else
        count_06=count_06+1;
        classif(j,:)=[scloth water];
    end
elseif (ml_x09(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water straw];
    else
        count_07=count_07+1;
        classif(j,:)=[straw water];
    end
elseif (ml_x09(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water strcloth];
    else
        count_08=count_08+1;
        classif(j,:)=[strcloth water];
    end
elseif (ml_x09(i*5-4,j)==ml_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water cork];
    else
        count_01=count_01+1;
        classif(j,:)=[cork water];
    end
elseif (ml_x09(i*5-4,j)==ml_x10(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_09=count_09+1;
        classif(j,:)=[water weave];
    else
        count_10=count_10+1;
        classif(j,:)=[weave water];
    end
end
end
results(i*(N-5)-(N-6):i*(N-5),:)=classif; % this part is going to be used for fusion

```

```

end
fprintf('\n\nResults for Approximation Coefficients of Testing Samples for Water\n\n');
fprintf('Results will be expressed as: ("original class"/"class classified as")\n');
fprintf('Classification results water/cork=%4.0f\n',count_01);
fprintf('Classification results water/french=%4.0f\n',count_02);
fprintf('Classification results water/liz=%4.0f\n',count_03);
fprintf('Classification results water/net=%4.0f\n',count_04);
fprintf('Classification results water/repskin=%4.0f\n',count_05);
fprintf('Classification results water/scloth=%4.0f\n',count_06);
fprintf('Classification results water/straw=%4.0f\n',count_07);
fprintf('Classification results water/strcloth=%4.0f\n',count_08);
fprintf('Classification results water/water=%4.0f\n',count_09);
fprintf('Classification results water/weave=%4.0f\n',count_10);
% Percentaje of classification
a=count_01+count_02+count_03+count_04+count_05+count_06+count_07+count_08+count_09+count_10;
fprintf('\nTotal samples=%4.0f\n',a);
rec(9,:)= [count_01 count_02 count_03 count_04 count_05 count_06 count_07 count_08 count_09 count_10];
perc(1,9)=count_09*100/a;
fprintf('\nPercentaje of classification=%4.4f\n',perc(1,9));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classification process %
count_01=0;
count_02=0;
count_03=0;
count_04=0;
count_05=0;
count_06=0;
count_07=0;
count_08=0;
count_09=0;
count_10=0;

% Data classified
cork=1;
french=2;
liz=3;
net=4;
repskin=5;
scloth=6;
straw=7;
strcloth=8;
water=9;
weave=10;
random_selection=11;
% the only thing that is going to change according to what we are
% classifying is the first element of the array:
% i.e. classif(j,:)=["cork" cork]; only change the part between " "

```

```

% Depending on which class is going to be classified we must change the
% first "if" below, instead of ml_x01 be compared with every single ml_x we
% must change it for example if is french class it should be ml_02 instead
% of ml_01.
for i=361:400 % number of training samples
    for j=1:(N-5) % number of vectors for each coefficient matrix to be classified
        if ((ml_x10(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x10(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x10(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x10(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x10(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x10(i*5-4,j)>ml_x01(i*5-4,j)))
            count_10=count_10+1;
            classif(j,:)=['weave 0'];
        elseif ((ml_x02(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x02(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x02(i*5-4,j)>ml_x10(i*5-4,j)))
            count_02=count_02+1;
            classif(j,:)=['french 0'];
        elseif ((ml_x03(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x03(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x03(i*5-4,j)>ml_x10(i*5-4,j)))
            count_03=count_03+1;
            classif(j,:)=['liz 0'];
        elseif ((ml_x04(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x04(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x04(i*5-4,j)>ml_x10(i*5-4,j)))
            count_04=count_04+1;
            classif(j,:)=['net 0'];
        elseif ((ml_x05(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x05(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x05(i*5-4,j)>ml_x10(i*5-4,j)))
            count_05=count_05+1;
            classif(j,:)=['repskin 0'];
        elseif ((ml_x06(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x07(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x06(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x06(i*5-4,j)>ml_x10(i*5-4,j)))
            count_06=count_06+1;
            classif(j,:)=['sclotch 0'];
        elseif ((ml_x07(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x03(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x05(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x01(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x07(i*5-4,j)>ml_x09(i*5-4,j))...
            &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))

```

```

        &&(ml_x07(i*5-4,j)>ml_x10(i*5-4,j)))
count_07=count_07+1;
classif(j,:)=['straw 0'];
elseif ((ml_x08(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x01(i*5-4,j))&&(ml_x08(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x08(i*5-4,j)>ml_x10(i*5-4,j)))
count_08=count_08+1;
classif(j,:)=['strcloth 0'];
elseif ((ml_x09(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x09(i*5-4,j)>ml_x01(i*5-4,j))...
        &&(ml_x09(i*5-4,j)>ml_x10(i*5-4,j)))
count_09=count_09+1;
classif(j,:)=['water 0'];
elseif ((ml_x01(i*5-4,j)>ml_x02(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x03(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x04(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x05(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x06(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x07(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x08(i*5-4,j))&&(ml_x01(i*5-4,j)>ml_x09(i*5-4,j))...
        &&(ml_x01(i*5-4,j)>ml_x10(i*5-4,j)))
count_01=count_01+1;
classif(j,:)=['cork 0'];
elseif (ml_x10(i*5-4,j)==ml_x02(i*5-4,j))
    random_pick=rand;          % We use rand to let random selection when
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_02=count_02+1;
        classif(j,:)=['french weave'];
    end
elseif (ml_x10(i*5-4,j)==ml_x03(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_03=count_03+1;
        classif(j,:)=['liz weave'];
    end
elseif (ml_x10(i*5-4,j)==ml_x04(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=['weave random_selection'];
    else
        count_04=count_04+1;
        classif(j,:)=['net weave'];
    end
elseif (ml_x10(i*5-4,j)==ml_x05(i*5-4,j))

```

```

random_pick=rand;
if random_pick>.5
    count_10=count_10+1;
    classif(j,:)=[weave random_selection];
else
    count_05=count_05+1;
    classif(j,:)=[repskin weave];
end
elseif (ml_x10(i*5-4,j)==ml_x06(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_06=count_06+1;
        classif(j,:)=[scloth weave];
    end
elseif (ml_x10(i*5-4,j)==ml_x07(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_07=count_07+1;
        classif(j,:)=[straw weave];
    end
elseif (ml_x10(i*5-4,j)==ml_x08(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_08=count_08+1;
        classif(j,:)=[strecloth weave];
    end
elseif (ml_x10(i*5-4,j)==ml_x09(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_09=count_09+1;
        classif(j,:)=[water weave];
    end
elseif (ml_x10(i*5-4,j)==ml_x01(i*5-4,j))
    random_pick=rand;
    if random_pick>.5
        count_10=count_10+1;
        classif(j,:)=[weave random_selection];
    else
        count_01=count_01+1;
        classif(j,:)=[cork weave];
    end

```



```

% Using Pseudo Inverse for the inversion of the covariance matrix
% maximum likelihood classification
% Example of how this works
% But we will use the general case multivariate and multiple classes
%  $h_i(X) = -(1/2) \ln |K_i| - (1/2)(X - \mu_i)'(K_i^{-1})(X - \mu_i)$ 
%  $h_i(X) >= h_j(X)$  (for all j ----> X element of class i)

% % difference of discriminant functions using Pseudo Inverse
%  $h_x = -0.5 \log(\det(\text{cov\_data0})) - 0.5(x\_data0 - \text{mean\_data0})'((\text{cov\_data0}' * \text{cov\_data0})^{-1} * \text{cov\_data0})(x\_data0 - \text{mean\_data0});$ 

% % SVD for matrix inversion
%  $[U, S, V] = \text{svd}(\text{cov\_data0});$ 
%  $\text{cov\_inv} = V * S^{-1} * U';$ 
rank_cov = rank(cov_data0);
if rank_cov == 3
    cov_data0 = eye(4);
end
% difference of discriminant functions using SVD
 $h_x = -0.5 \log(\det(\text{cov\_data0})) - 0.5(x\_data0 - \text{mean\_data0})'(\text{cov\_data0}^{-1})(x\_data0 - \text{mean\_data0});$ 

% % QR for matrix inversion
%  $[Q, R] = \text{qr}(\text{cov\_data0});$ 
%  $\text{cov\_inv} = R^{-1} * Q;$ 
%
% % difference of discriminant functions using QR
%  $h_x = -0.5 \log(\det(\text{cov\_data0})) - 0.5(x\_data0 - \text{mean\_data0})' \text{cov\_inv} (x\_data0 - \text{mean\_data0});$ 

```

APPENDIX C6 MAJORITY VOTING FUSION RULE

```

% load results_db1.mat
% load results_bior31.mat
% load results_sym2.mat
% load results_coif1.mat
% load results_db2.mat
load results_db1.mat
[M N] = size(results_eu_db1);
count_01 = 0;
count_02 = 0;
count_03 = 0;

count_k01 = 0;
count_k02 = 0;
count_k03 = 0;

veggie = 1;

```

```

urban=2;
water=3;

random_selection=11;
count_times=0;
i_count=0;

for j=1:3
    for i=1:M
        count_01=0;
        count_02=0;
        count_03=0;

        eu=results_eu_db1(i,2*j-1);
        if (eu==1)
            count_01=count_01+1;
            count_times=count_times+1;
        elseif (eu==2)
            count_02=count_02+1;
            count_times=count_times+1;
        elseif (eu==3)
            count_03=count_03+1;
            count_times=count_times+1;

        end
        knn3=results_knn3_db1(i,2*j-1);
        if (knn3==1)
            count_01=count_01+1;
            count_times=count_times+1;
        elseif (knn3==2)
            count_02=count_02+1;
            count_times=count_times+1;
        elseif (knn3==3)
            count_03=count_03+1;
            count_times=count_times+1;

        end
        knn5=results_knn5_db1(i,2*j-1);
        if (knn5==1)
            count_01=count_01+1;
            count_times=count_times+1;
        elseif (knn5==2)
            count_02=count_02+1;
            count_times=count_times+1;
        elseif (knn5==3)
            count_03=count_03+1;
            count_times=count_times+1;

        end
        ml=results_ml_db1(i,2*j-1);
        if (ml==1)
            count_01=count_01+1;

```

```

    count_times=count_times+1;
elseif (m1==2)
    count_02=count_02+1;
    count_times=count_times+1;
elseif (m1==3)
    count_03=count_03+1;
    count_times=count_times+1;

end
if (count_times==4) % Classification of K Nearest Neighborhood
    count_times=0;
    % i_count=i_count+1;
    if ((count_01>count_02)&&(count_01>count_03))
        count_k01=count_k01+1;
        results(i,2*j-1:2*j)=[veggie 0];
    elseif ((count_02>count_01)&&(count_02>count_03))
        count_k02=count_k02+1;
        results(i,2*j-1:2*j)=[urban 0];
    elseif ((count_03>count_02)&&(count_03>count_01))
        count_k03=count_k03+1;
        results(i,2*j-1:2*j)=[water 0];

    elseif (count_01==count_02)
        random_pick=rand;
        if random_pick>.5
            count_k01=count_k01+1;
            results(i,2*j-1:2*j)=[veggie urban];
        else
            count_k02=count_k02+1;
            results(i,2*j-1:2*j)=[urban veggie];
        end
    elseif (count_01==count_03)
        random_pick=rand;
        if random_pick>.5
            count_k01=count_k01+1;
            results(i,2*j-1:2*j)=[veggie water];
        else
            count_k03=count_k03+1;
            results(i,2*j-1:2*j)=[water veggie];
        end
    elseif (count_02==count_03)
        random_pick=rand;
        if random_pick>.5
            count_k02=count_k02+1;
            results(i,2*j-1:2*j)=[urban water];
        else
            count_k03=count_k03+1;
            results(i,2*j-1:2*j)=[water urban];
        end
    end
end

```

```

        end
    end
end
end
for j=1:3
    count_01=0;
    count_02=0;
    count_03=0;

    for i=1:M
        x=results(i,2*j-1);
        if (x==1)
            count_01=count_01+1;
            % count_times=count_times+1;
        elseif (x==2)
            count_02=count_02+1;
            % count_times=count_times+1;
        elseif (x==3)
            count_03=count_03+1;
            % count_times=count_times+1;

        end
    end
    if (i==M)
        if (j==1)
            perc(j)=count_01*100/M;
            resul(1,:)=[count_01 count_02 count_03];
        elseif (j==2)
            perc(j)=count_02*100/M;
            resul(2,:)=[count_01 count_02 count_03];
        elseif (j==3)
            perc(j)=count_03*100/M;
            resul(3,:)=[count_01 count_02 count_03];

        end
    end
end
end
end
tabla=[resul perc'];

```