CURRENCY AUTHENTICATION USING MATLAB

AMAN VELNAZAROV

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2005

CURRENCY AUTHENTICATION USING MATLAB

by

Aman Velnazarov

Dissertation submitted to the Electrical & Electronics Engineering Programme
In partial Fullfilment of the Requirements for the
Degree Bachelor of Engineering (Hons)
ELECTRICAL&ELECTRONICS ENGINEERING

JUNE 2005

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Currency Authentication using MATLAB

by

Aman Velnazarov

A project dissertation submitted to the
Electrical and Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (HONS)
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by

(Mr. Patrick Sebastian)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JUNE 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

AMAN VELNAZAROV

Acknowledgments

First of all, I would like to thank my advisor, Mr. Patrick Sebastian, who has been very helpful through out the year with his very helpful guidance, endless support and encouragement. His support was very crucial during the most demanding phases of this research.

A great deal of thanks goes to Mr. Santiago Serrano from Department of Electrical Engineering, Drexel University for his excellent code on Eigenfaces. His tutorials helped me to complete the project.

I sincerely wish to express my appreciation to my friends, without whose help and support at critical times in my research could not have gone so far.

ABSTRACT

Today the ubiquitous distribution of high technology scanning and printing equipment enables the "home" user to make counterfeits of high value documents like checks, tickets, licenses, identification cards and other secure documents. High value documents have been and will continue to be forged as long as the value realized from the counterfeiting is higher than the cost of duplicating the original. There are no perfect counterfeits, and there are no perfect designs fully immune to counterfeiting. In the past, the hands of craftsman and a perfect eye were required to make a high quality counterfeit. Today, highly sophisticated, state-of-the-art reprographic systems do not require skilled professionals to operate them. They are widely available to the general public. These devices are generally simple to use and create an "opportunity" for the "home counterfeiter". It is becoming increasingly difficult to spot alterations or counterfeits using only human sensory evaluation. There is an ever-increasing demand for new technologies and methods of counterfeit detection and forensic analysis to safeguard the integrity of high value documents. This is where process of authentication comes in. The aim of the project is to produce a reliable system that offers an easy and effective authentication technique of a currency. The objectives of *Currency Authentication* are to design a system that will be able to evaluate the integrity of currency contents relative to the original and of being able to detect, in an automatic way, malevolent currency modifications. For these purposes system software must be developed based on current available techniques and mechanisms.

Table of Contents

Acknowledgment....................i
Abstract....................ii

CHAPTER 1. INTRODUCTION.................... 6
1.1 Background Study.................... 6
1.1.1 Image recognition systems.................... 6
1.1.2 The difficulty of computer vision.................... 7
1.2 Currency Recognition/Authentication.................... 9
1.2.1 Existing Application for Currency Authentication.................... 10
1.3 Problem Statement.................... 10
1.3.1 Problem Identification.................... 10
1.3.2 Significance of the Project..................... 10
1.3 Objectives and Scope of Study.................... 11
1.3.1 Objectives.................... 11
1.3.2 Scope of study.................... 12
CHAPTER 2. LITERATURE REVIEW/THEORY.................... 13
2.1 Basic Concepts.................... 13
2.1.1 Overview.................... 13
2.1.2 Pattern Classes and Patterns.................... 14
2.1.3 Fundamental Problems in Pattern Recognition System Design.................... 14
2.1.4 Outline of a Typical Pattern Recognition System.................... 16
2.1.2 Currency Recognition.................... 16
2.2 Systems Overview.................... 17
2.2.1 Eigenfaces.................... 18
2.2.2 Neural Networks.................... 20
2.2.3 PCA-Based and Fisher Discriminant-Based Image Recognition.................... 20
2.3 Eigenfaces for Recognition.................... 21
2.3.1 How does it work?.................... 22
2.3.2 Overview over the algorithm.................... 23
2.3.3 Eigenvectors and eigenvalues.................... 26
2.3.4 Calculation of eigenfaces with PCA.................... 26
CHAPTER 3. METHODOLOGY/PROJECT WORK.................... 28
3.1 The Eigenface Technique.................... 28
3.1.1 Approaching Eigenface Technique.................... 28
3.1.2 Eigenface Recognition Procedure.................... 30
3.2 Images Database.................... 31
3.3 Tools Used.................... 31
3.3.1 MATLAB 7.0.................... 31
3.3.2 High Quality Scanner.................... 31
CHAPTER 4. RESULTS AND FINDINGS.................... 32
4.1 Results.................... 32
4.1.1 Output of Currency Authentication System.................... 32
4.2 Input image selection and results.................... 44
4.2.1 Euclidean Distance.................... 49

4.2.1.1 Introduction 49
4.2.1.2 Classifying the images 51
4.2.1.3 Euclidean Distance 51
CHAPTER 5. CONCLUSION AND RECOMMENDATION . 57
5.1 Conclusion 57
5.2 Recommendations and future work 59
REFERENCES 61
APPENDICES 62
Appendix A - Code Listing 62
Appendix B – Results 70
Appendix C – Malaysian Ringgit 79
RM1 Security Features 82
Appendix D - How To Detect Counterfeit Money (US dollar) 89
Appendix F - Suggested Milestones for Final Year Project 91

List of Figures

Figure 1.1 Image of *Polypedates eques* - frog.

Figure 2.1. Two disjoint pattern classes. Each pattern is characterized by two measurements: height and weight. The pattern vector therefore is in the form of $x = \{x_1, x_2\}^T$

Figure 2.2. Functional block diagram of an adaptive pattern recognition system

Figure 2.3 High-level functioning principle of the eigenface-based facial recognition algorithm

Figure 2.4: The basic algorithm for neural-networks, used in this case for face detection.

Figure 2.5 a) Points in a 2-dimensional space. b) Points mixed when projected onto a line. c) Points separated when projected onto a line

Figure 2.6 High-level functioning principle of the eigenface-based facial recognition Algorithm

Figure 4.1 Currency Authentication Menu Box

Figure 4.2 Currency Authentication Help

Figure 4.3 The "Preprocessing" Menu Box

Figure 4.4 Currency images displayed in the Training Set

Figure 4.5 Normalized Training Set

Figure 4.6 The mean image

Figure 4.7 The "Recognition" Menu Box

Figure 4.8 A 7x7 image transformed into a 49 dimension vector

Figure 4.9 Image space

Figure 4.10 Eigenfaces

Figure 4.11 The four possible results when projecting an image into faces space. The face space is formed by just two eigenfaces (u_1 and u_2) and contains three known images (Ω_1, Ω_2 and Ω_3)

Figure 4.12 Currency image in image space

Figure 4.13 Reconstructed currency image in image space

Figure 4.14 Example of a known currency image

Figure 4.15 Reconstructed currency image

Figure 4.16 Weight of known input image

Figure 4.17 Euclidian distance of known input image

Figure 4.18 Training set with 10 images

Figure 4.19 Eigenfaces, Training Set 1

Figure 4.20 Test input face image

Figure 4.21 The reconstructed image

Figure 4.22 Euclidean distance of face input image

Figure 6.1 Training Set 1 eigenfaces

Figure 6.1 Input image 1

Figure 6.3 Euclidean distance of input image 1

Figure 6.4 Input image 2

Figure 6.5 Euclidean distance of input image 2

Figure 6.6 Slightly tampered counterfeit image

Figure 6.7 Euclidean distance of counterfeit currency image

Figure 6.8 Plane as an input image

Figure 6.9 Euclidean distance of a plane image

Figure 6.10 All test images

Figure 6.11 Counterfeit image 1

Figure 6.12 Counterfeit image 2

Figure 6.13 Counterfeit image 3

Figure 6.14 Counterfeit image 4

Figure 6.15 Counterfeit image 5

Figure 6.16 50 Malaysian Ringgit

List of Tables

Table 4.1 Resulting Euclidean distances for various inputs including "counterfeit" image

Table 6.1 Euclidean distances for various input images including "counterfeit" image

List of Abbreviations

I	Face image
$N \times N$	Size of I
Γ	Training set

Γ_i	Face image i of the training set
Γ_{new}	New (unknown) image
Φ	Average face
$M = \mid \Gamma \mid$	Number of eigenfaces
M`	Number of eigenfaces used for face recognition
C	Covariance matrix
X^T	Transposed X (if X is a matrix)
u	Eigenvector (eigenface)
λ	Eigenvalue
ω_i	Weight i
θ	Threshold value

CHAPTER 1

INTRODUCTION

1.1 Background Study

1.1.1 Image recognition systems

Automated image recognition is an interesting computer vision problem with many commercial and law enforcement applications. Mug shot matching, user verification and user access control, crowd surveillance, enhanced human computer interaction and of course in our case currency authentication all become possible if an effective image recognition system can be implemented. While research into this area dates back a few decades, it is only very recently that acceptable results have been obtained. However, image recognition is still an area of active research since a completely successful approach or model has not been proposed to solve the problem. The inadequacy of automated image recognition systems is especially apparent when compared to our own innate image recognition ability. We perform recognition, an extremely complex visual task, almost instantaneously and our own recognition ability is far more robust than any computer's can hope to be. We can recognize a familiar individual under very adverse lighting conditions, from varying angles or view points. Scaling differences, different backgrounds do not affect our ability to recognize. Furthermore, we are able to recognize images or the objects that can number in several thousand which we saw during our lifetime.

1.1.2 The difficulty of computer vision

Unfortunately it is not possible now, nor will it be possible in the foreseeable future to make a computing machine that actually 'understands' what is sees. The level of vision and understanding which is instinctive to us is still far out of the reach of our silicon creations.

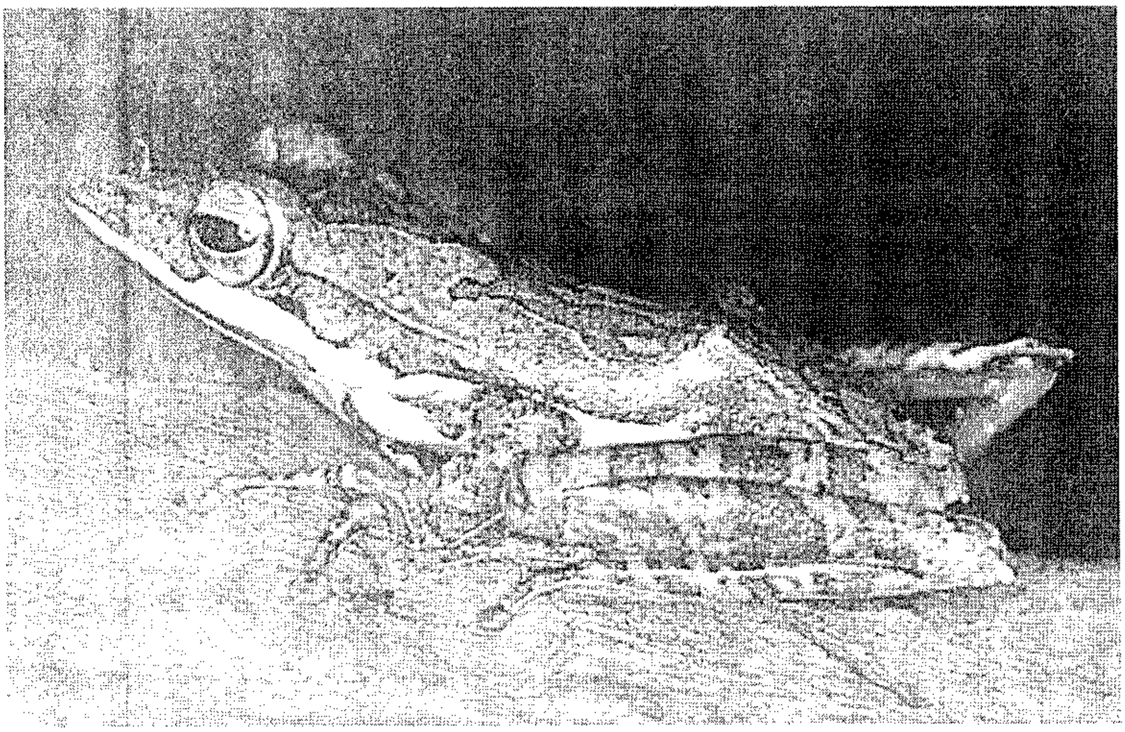


Figure 1.1 Image of *Polypedates eques* - frog.

The ability to understand that the above image is not just a collection of pixels but is of a camouflaged frog on a log and to be able to identify exactly where the frog ends and log begins on the image is truly incredible. The fact that half of a primate's cerebral cortex is dedicated to visual processing underlies the difficulty of this task (Zeki, 1993), and it would be naïve of not impossible to think that we can enable computers to perform similar tasks.

But technically, why are computer vision problems so hard to solve? After all, while laudable results have been obtained in other artificial intelligence areas such as natural language processing, game theory, forecasting, control and even speech processing, computer vision seems to have lagged behind.

The main difficulty in vision problems is that almost all of them are ill-defined. For example, while segmenting (dividing) our image of *Polypedates eques* into areas of "frog", "log" and "background" is intuitive and an innate ability to us, it does not seem possible to find a definite problem specification of this task that a computer would understand.

Another factor is that even well defined computer vision problems may be ill-posed. Hadamard (1923) defined a problem as well posed if

(1) a solution exists,

(2) the solution is unique,

(3) the solution depends continuously on the initial data (stability property).

Many computer vision problems are ill-posed because information is lost in the transformation from the 3D world to a 2D image. Therefore, we cannot uniquely reconstruct the 3D representation from the 2D image and multiple solutions are often 'correct'.

The complexity of computer vision problems is exacerbated by the fact that we are dealing with huge chunks of data. A typical gray-scale image has 640x480 pixels, each with 8-bit (256) intensity values (gray-levels). Therefore, the size of the whole image is 640x480x8 bits = 2,457,600 bits. Any algorithm with high complexity would be extremely slow in computer vision and we must therefore make an effort to solve these problems using very simple processing techniques.

However, even with all these constraints it is possible to get useful results in computer vision by reducing a problem's generality. The computer vision application's problem domain can be restricted to a well-defined structured environment and assumptions could be made about lighting, types of object, etc. Therefore, instead of trying to create a system that is suitable for all vision problems the computer vision and artificial intelligence communities have concentrated on obtaining useful results to real-world, limited applications in vision. Automated image recognition has thus become the holy grail of computer vision artificial intelligence. It is probably the most challenging and ambitious of the computer vision projects that are being studied and is not just a fascinating theoretical problem, but there is a real-world need for such a system.

1.2 Currency Recognition/Authentication

The recent proliferation of digital multimedia content has raised concerns about authentication mechanisms for multimedia data. For example, consider digital photography, which is fast replacing conventional analog techniques. In the analog world, an image had generally been accepted as a "proof of occurrence" of the depicted event. The proliferation of digital images and the relative ease, by which they can be manipulated, has changed this situation dramatically. Given an image, in digital or analog form, one can no longer be assured of its authenticity. This has led to the need for image authentication techniques.

Currency recognition is a pattern recognition task performed specifically on currencies. It can be described as classifying a currency either "authentic" or "unauthentic", after comparing it with stored known images. Computational models of currency recognition must address several difficult problems. This difficulty arises from the fact that currencies must be represented in a way that best utilizes the available image information to distinguish a particular currency from all other. Currencies pose a particularly difficult problem in this respect because all of them are similar to one another in that they contain the same set of features in roughly the same manner.

1.2.1 Existing Application for Currency Authentication

- **microDAST CURRENCY DETECTOR**

  San Diego Magnetics' Document Authentication Security Technology (DAST™) sensors use proprietary technology to provide very sensitive and accurate reading of magnetic security features in currency and documents.

1.3 Problem Statement

1.3.1 Problem Identification

Designing a system for automatic image content recognition is a non-trivial task that has been studied for a variety of applications. Computer recognition of specific objects in digital images has been put to use in manufacturing industries, intelligence and surveillance, and image database cataloging to name a few. But perhaps an area involving currency authentication is most important. Digital image application has become important in daily life with the arrival of the digital era. With an ever advancing technology the proliferation of a currency, is also becoming an easy task. A currency can be forged without leaving any traces, using some image processing software.

1.3.2 Significance of the Project

A counterfeit is an imitation that is made with the intent to deceptively represent its content or origins. The word counterfeit most frequently describes forged money or documents. Counterfeiting money is probably as old as money itself. However, the introduction of paper money has made it an easier thing to do.
Traditionally, anti-counterfeiting measures involved including fine detail with raised intaglio printing on bills which would allow non-experts to easily spot forgeries.

In the late twentieth century advances in computer and photocopy technology made it possible for people without sophisticated training to easily copy currency. In response, national engraving bureaus began to include new more sophisticated anti-counterfeiting systems such as holograms, multi-colored bills, embedded devices such as strips, micro printing and inks whose colors changed depending on the angle of the light, and the use of design features such as the "EURion constellation" which disables modern photocopiers. Software programs such as Adobe Photoshop have been modified by their manufacturers to obstruct manipulation of scanned images of banknotes.
But still the ubiquitous distribution of high technology scanning and printing equipment enables the "home" user to make counterfeits. These devices are generally simple to use and create an "opportunity" for the "home counterfeiter". There is an ever-increasing demand for new technologies and methods of counterfeit detection and forensic analysis to safeguard the integrity of high value documents in this case a currency.
This currency authentication project is conducted as an initiative in helping government to fight the counterfeiters. The project is focused on providing a currency authentication system that will benefit government as well as other private sectors.

1.3 Objectives and Scope of Study

1.3.1 Objectives

A human eye cannot directly detect the tampered regions of a forged currency. Because of that we need to develop an automated, reliable and computationally efficient authentication system. A system that can provide means of ensuring the originality of a currency by detecting any significant malicious manipulations.

A software using MATLAB must be developed to carry out tasks mentioned.

There are some subordinate objectives to successfully implement the tasks and achieve the focal objectives:

I. to study the image recognition basics
II. to research different techniques applied to image recognition
III. evaluate the techniques based on their adaptability, stability and reliability
IV. to develop the currency authentication algorithm

1.3.2 Scope of study

The following problem scope for this project was arrived at after reviewing the literature on image recognition, and determining possible real-world situations where such systems would be of use. The following system requirements were identified:

- A system to recognize a given currency image.
- An implemented system must display a high degree of lighting invariance.
- A system must posses near real-time performance.

CHAPTER 2

LITERATURE REVIEW/THEORY

2.1 Basic Concepts

2.1.1 Overview

One of the major problems in the design of modern information systems is automatic pattern recognition.

Recognition is regarded as a basic attribute of human beings, as well as other living organisms. A pattern is the description of an object. According to the nature of the patterns to be recognized, recognition acts can be divided into two major types:

- Recognition of concrete items. This may be referred to as sensory recognition, which includes visual and aural pattern recognition. This recognition process involves the identification and classification of spatial and temporal patterns. Examples of spatial patterns are characters, fingerprints, physical objects, and images. Temporal patterns include speech waveforms, time series, electrocardiograms and target signatures.

- Recognition of abstract items. On the other hand, an old argument, or a solution to a problem can be recognized. This process involves the recognition of abstract items and can be termed conceptual recognition.

2.1.2 Pattern Classes and Patterns

Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail.

A pattern class is a category determined by some given common attributes or features. The features of a pattern class are the characterizing attributes common to all patterns belonging to that class. Such features are often referred to as intraset features. The features which represent the differences between pattern classes may be referred to as the interset features.

A pattern is the description of any member of a category representing a pattern class. For convenience, patterns are usually represented by a vector such as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots\cdot \\ \cdots\cdot \\ x_n \end{bmatrix}$$

where each element x_j, represents a feature of that pattern. It is often useful to think of a pattern vector as a point in an n-dimensional Euclidean space.

2.1.3 Fundamental Problems in Pattern Recognition System Design

The design of an automatic pattern recognition system generally involves several major problem areas:

- First of all, we have to deal with the representation of input data which can be measured from the objects to be recognized. This is the sensing problem. Each measured quantity describes a characteristic of the pattern or object. In other words, a pattern vector that describes the input data has to be formed. The pattern vectors contain all the measured information available about the patterns. The set of patterns belonging to the same class corresponds to an ensemble of points scattered within some region of the measurement space. A simple example of this is shown in Figure 2.1 for two pattern classes denoted by w 1 and w 2.

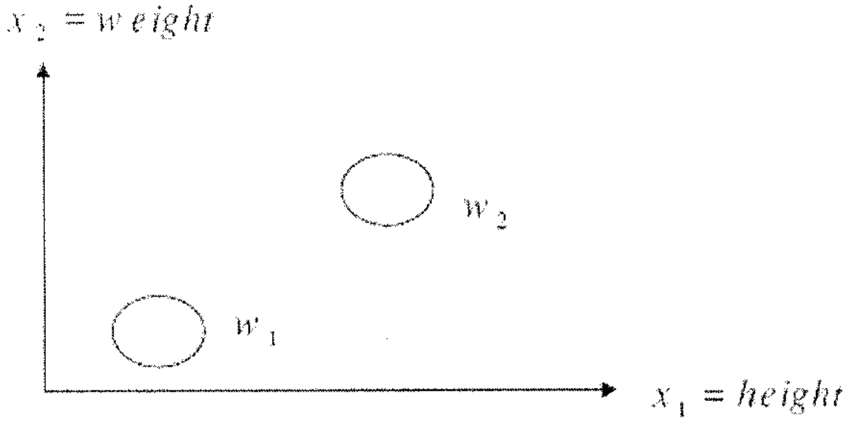


Figure 2.1. Two disjoint pattern classes. Each pattern is characterized by two measurements: height and weight. The pattern vector therefore is in the form of $x = \{x_1, x_2\}^T$

- The second problem in pattern recognition concerns the extraction of characteristic features or attributes from the received input data and the reduction of the dimensionality of pattern vectors. This is often referred to as the pre-processing and the feature extraction problem. The elements of intraset features which are common to all pattern classes under consideration carry no discriminatory information and can be ignored. If a complete set of discriminatory features for each pattern class can be determined from the measured data, the recognition and classification of patterns will present little difficulty. Automatic recognition may be reduced to a simple matching process or a table look-up scheme. However, in most pattern recognition problems which arise in practice, the determination of a complete set of discriminatory features is extremely difficult, if not impossible.

2.1.4 Outline of a Typical Pattern Recognition System

In Figure 2.2, functional block diagram of an adaptive pattern recognition system is shown. Although the distinction between optimum decision and pre-processing or feature extraction is not essential, the concept of functional breakdown provides a clear picture for the understanding of the pattern recognition problem.

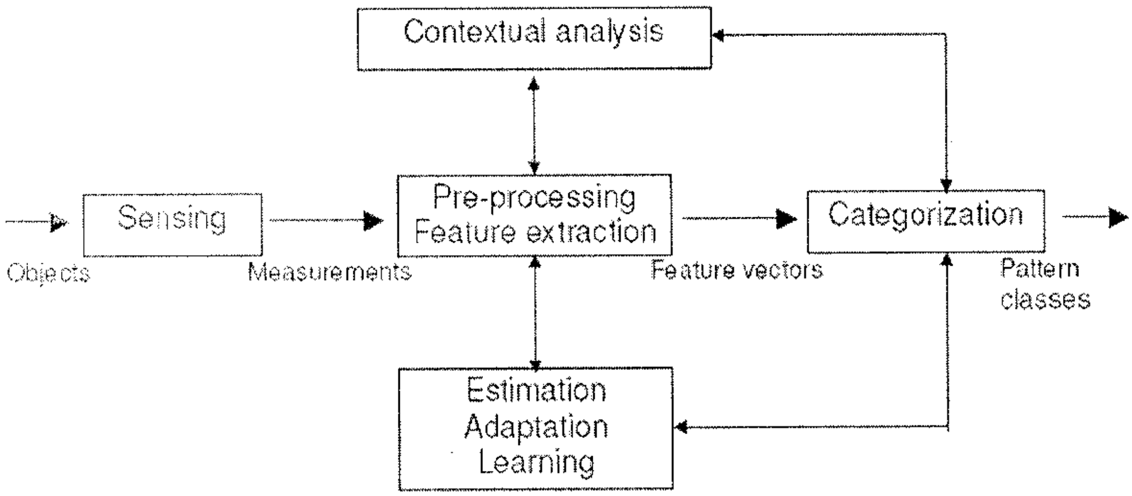


Figure 2.2. Functional block diagram of an adaptive pattern recognition system.

Correct recognition will depend on the amount of discriminating information contained in the measurements and the effective utilization of this information. In some applications, contextual information is indispensable in achieving accurate recognition. For instance, in the recognition of cursive handwritten characters and the classification of fingerprints, contextual information is extremely desirable. When we wish to design a pattern recognition system which is resistant to distortions, flexible under large pattern deviations, and capable of self-adjustment, we are confronted with the adaptation problem.

2.1.2 Currency Recognition

Currency recognition is a pattern recognition task performed specifically on currencies. It can be described as classifying a currency either "authentic" or "unauthentic", after comparing it with stored known images. Computational models of currency recognition must address several difficult problems. This difficulty arises from the fact that currencies must be represented in a way that best utilizes the available image information to distinguish a particular currency from all other. Currencies pose a particularly difficult problem in this respect because all of them are similar to one another in that they contain the same set of features in roughly the same manner.

2.2 Systems Overview

In order to successfully accomplish the project we have to have an efficient method for image recognition task.

Automated image recognition is a well-studied problem in computer vision. With the immense material on the web and also in scientific journals, it was hard to stick to one method of approach in this vast field. Many applications for image recognition are used today, especially with the current situation in the world. Reliability and adaptability can be considered to be the primary concern of an image recognition system. Automated recognition and authentication system will have a great use to fight the counterfeiters.

To this end, many image recognition techniques and methods have been proposed to solve this problem. Mainly they are such as Eigenfaces, Fisher Linear Discriminant, Neural Networks, and Support Vector Machines. Success has been achieved with each method to varying degrees and complexities.

Since most of them are heavily applied on biometrics, we will be basing our research on that.

2.2.1 Eigenfaces

The eigenface representation method for face recognition is based on the principal component analysis (PCA). The main idea is to decompose the face images into a small set of characteristic feature images called eigenfaces, which may be thought of as the principal component of the original images.

The algorithm for the facial recognition using eigenfaces is basically described in figure 2.3. First, the original images of the training set are transformed into a set of eigenfaces E. Afterwards, the weights are calculated for each image of the training set and stored in the set W. Upon observing an unknown image X, the weights are calculated for that particular image and stored in the vector W_X. Afterwards, W_X is compared with the weights of images, of which one knows for certain that they are faces (the weights of the training set W). One way to do it would be to regard each weight vector as a point in space and calculate an average distance D between the weight vectors from W_X and the weight vector of the unknown image W_X. If this average distance exceeds some threshold value θ then the weight vector of the unknown image WX lies too "far apart" from the weights of the faces. In this case, the unknown X is considered to not a face. Otherwise (if X is actually a face), its weight vector W_X is stored for later classification. The optimal threshold value θ has to be determined empirically.

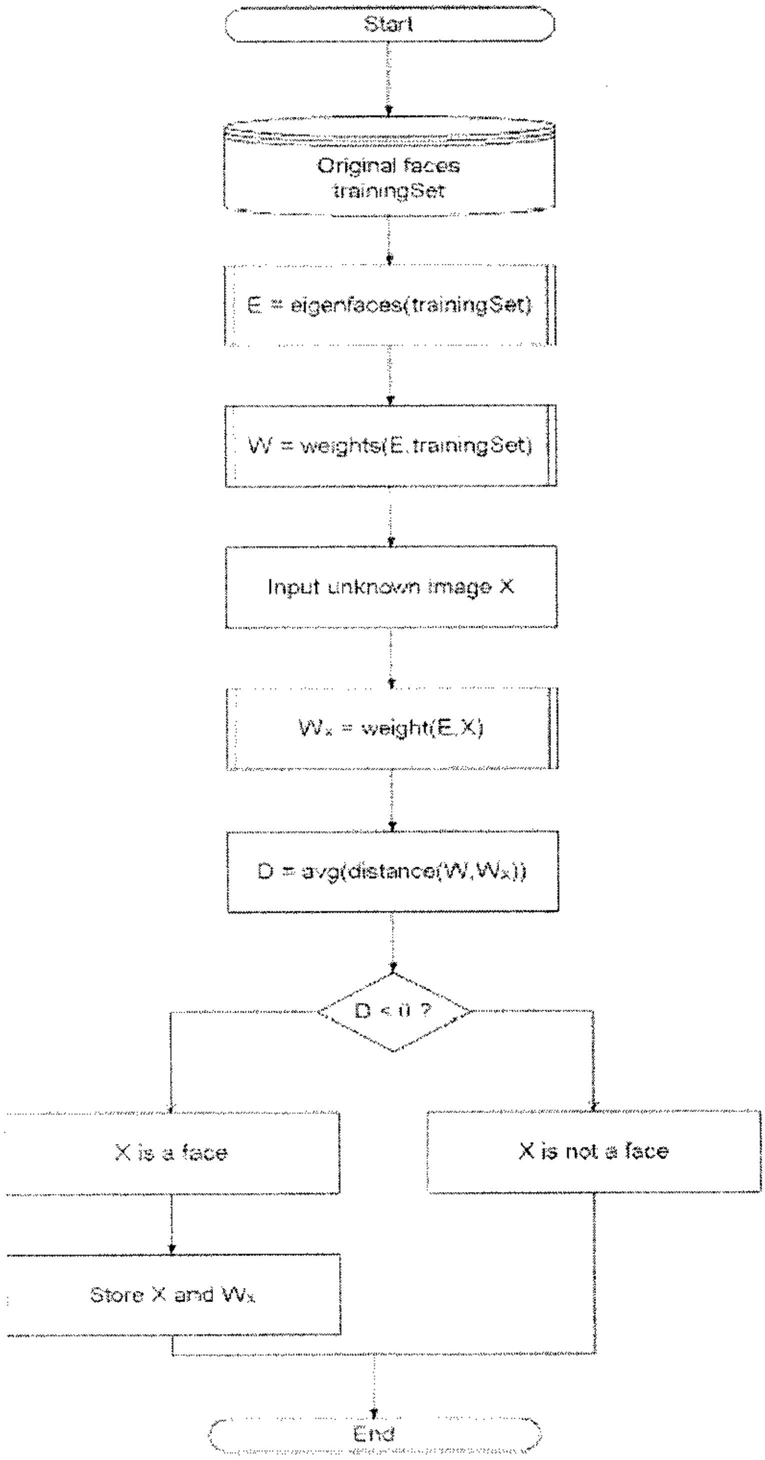


Figure 2.3 High-level functioning principle of the eigenface-based facial recognition algorithm

2.2.2 Neural Networks

A retinally connected neural network examines small windows of an image, and decides whether each window contains a feature image. The system arbitrates between multiple networks to improve performance over a single network. A bootstrap algorithm is used for training the networks, which adds false detections into the training set as training progresses. This eliminates the difficult task of manually selecting nonface training examples, which must be chosen to span the entire space of nonface images.

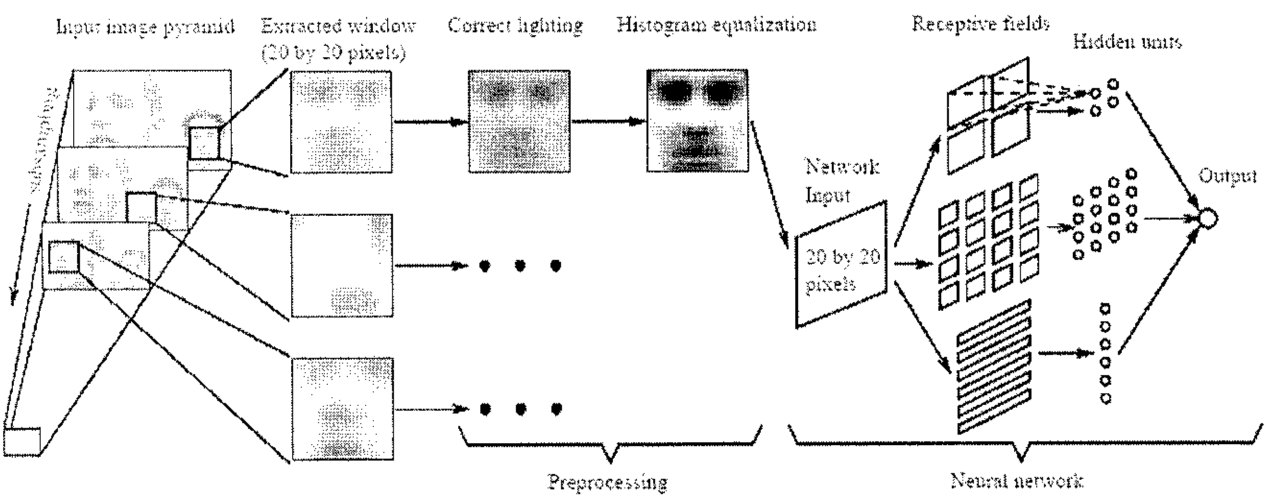


Figure 2.4: The basic algorithm for neural-networks, used in this case for face detection.

2.2.3 PCA-Based and Fisher Discriminant-Based Image Recognition

One method of identifying images is to measure the similarity between images. This is accomplished by using measures such as the L1 norm, L2 norm, covariance, Mahalanobis distance, and correlation. These similarity measures can be calculated on the images in their original space or on the images projected into a new space.
Fisher discriminants group images of the same class and separates images of different classes. Images are projected from N-dimensional space (where N is the number of pixels

in the image) to C-1 dimensional space (where C is the number of classes of images). For example, consider two sets of points in 2-dimensional space that are projected onto a single line (Figure 2a). Depending on the direction of the line, the points can either be mixed together (Figure 2b) or separated (Figure 2c). Fisher discriminants find the line that best separates the points. To identify a test image, the projected test image is compared to each projected training image, and the test image is identified as the closest training image.

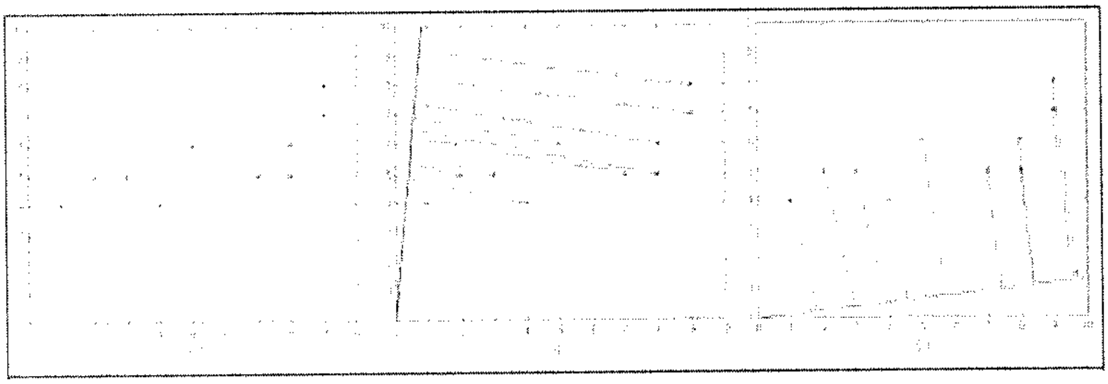


Figure 2.5 a) Points in a 2-dimensional space. b) Points mixed when projected onto a line. c) Points separated when projected onto a line

2.3 Eigenfaces for Recognition

Many techniques applied to the automated image recognition problem make arbitrary decisions on which image characteristics are actually important for recognition. For example, correlation-based techniques assume that all pixels of an image are equally important, when this is actually not the case at all. Another difficulty with other techniques is that they presume some significance of certain image characteristics over others with no grounds for this presumption.

Also there are some limitations on computing power for some algorithms for example, Neural Networks. Pattern recognition is a powerful technique for harnessing the information in the data and generalizing about it. Neural Networks learn to recognize the patterns which exist in the data set. Neural Networks teach themselves the patterns in the data. Major drawback for Neural Networks is that it can take time to train a model from a very complex data set. Neural techniques are computer intensive and will be slow on low end PCs or machines without math coprocessors.

We have focused our research toward developing a sort of pattern recognition scheme that does not depend on excessive geometry and computations. Eigenfaces approach seemed to be an adequate method to be used in currency authentication system due to its simplicity, speed and learning capability.

The scheme is based on an information theory approach that decomposes currency image into a small set of characteristic feature images called eigenfaces, which may be thought of as the principal components of the initial training set of main image. Recognition is performed by projecting a new image onto the subspace spanned by the eigenfaces and then classifying that feature by comparing its position in the face space with the positions of known individuals.

2.3.1 How does it work?

According to Dimitri Passarenko the task of image recognition is discriminating input signals (image data) into several classes (persons). The input signals are highly noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns which occur in any input signal. Such patterns, which can be observed in all signals, could be – in the domain of image recognition – the presence of some objects in any image as well as relative distances between these objects. These characteristic features are called eigenfaces in the image recognition domain (or principal components generally). They can be extracted out of original image data by means of a mathematical tool called Principal Component Analysis (PCA).

By means of PCA one can transform each original image of the training set into a corresponding eigenface. An important feature of PCA is that one can reconstruct any original image from the training set by combining the eigenfaces. Remember that eigenfaces are nothing less than characteristic features of the faces. Therefore one could say that the original image can be reconstructed from eigenfaces if one adds up all the eigenfaces (features) in the right proportion. Each eigenface represents only certain features of the image, which may or may not be present in the original image. If the feature is present in the original image to a higher degree, the share of the corresponding eigenface in the"sum" of the eigenfaces should be greater. If, contrary, the particular feature is not (or almost not) present in the original image, then the corresponding eigenface should contribute a smaller (or not at all) part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces. That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image.
If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces exactly. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces).

2.3.2 Overview over the algorithm

The algorithm for the image recognition using eigenfaces is basically described in figure 2.1. First, the original images of the training set are transformed into a set of eigenfaces E. Afterwards; the weights are calculated for each image of the training set and stored in the set W. Upon observing an unknown image X, the weights are calculated for that particular image and stored in the vector W_X. Afterwards, W_X is compared with the weights of images, of which one knows for certain that they are faces (the weights of the training set W). One way to do it would be to regard each weight vector as a point in space and calculate an average distance D between the weight vectors from W_X and the weight vector of the unknown image W_X. If this average distance exceeds some threshold value θ then the weight vector of the unknown image W_X lies too "far apart" from the weights of the faces. In this case, the unknown X is considered to not a face. Otherwise (if X is actually a face), its weight vector W_X is stored for later classification. The optimal threshold value θ has to be determined empirically.

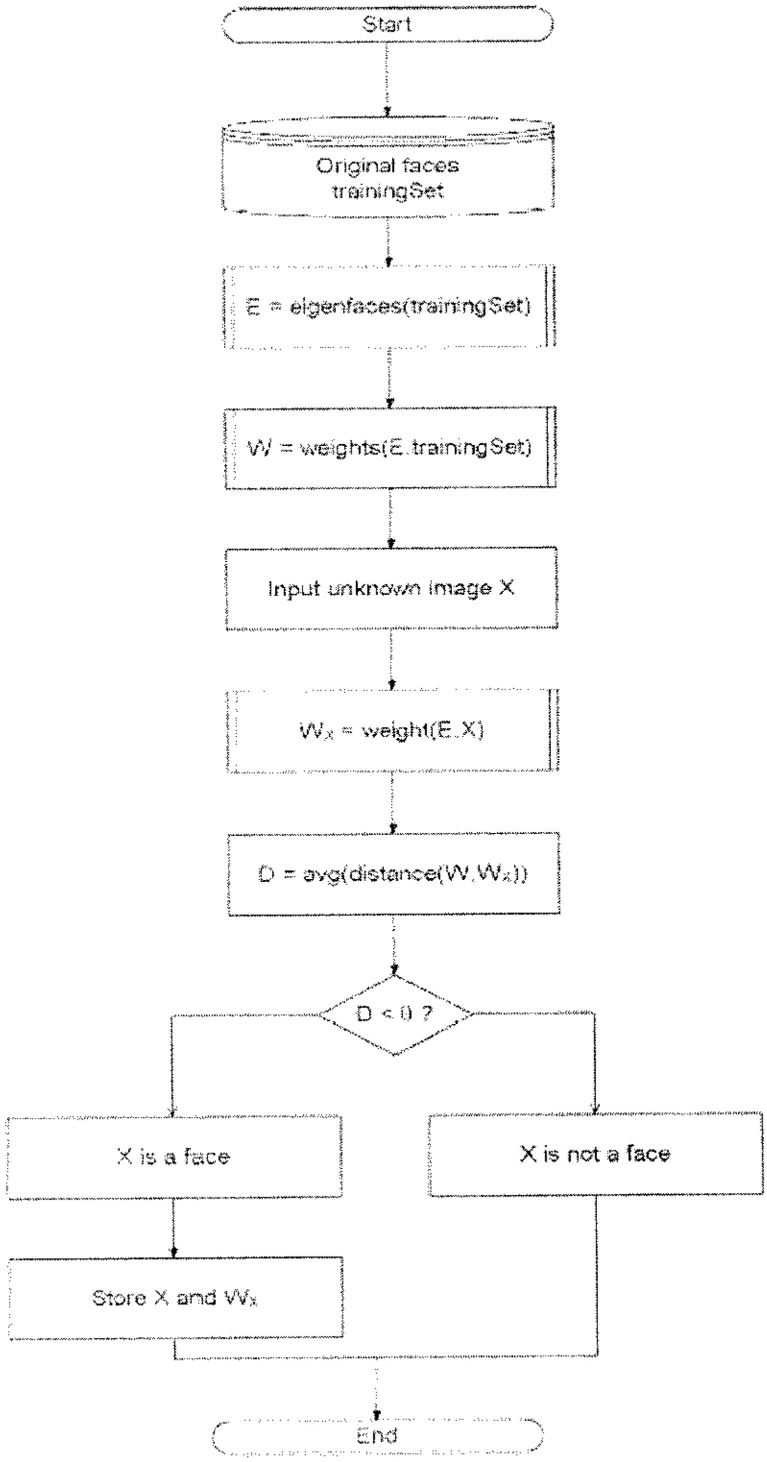


Figure 2.1 High-level functioning principle of the eigenface-based facial recognition algorithm

2.3.3 Eigenvectors and eigenvalues

An eigenvector of a matrix is a vector such that, if multiplied with the matrix, the result is always an integer multiple of that vector. This integer value is the corresponding eigenvalue of the eigenvector. This relationship can be described by the equation $M \times u = \lambda \times u$, where u is an eigenvector of the matrix M and λ is the corresponding eigenvalue. Eigenvectors possess following properties:

- They can be determined only for square matrices.
- There are n eigenvectors (and corresponding eigenvalues) in a $n \times n$ matrix.
- All eigenvectors are perpendicular, i.e. at right angle with each other.

2.3.4 Calculation of eigenfaces with PCA

In this section, the original scheme for determination of the eigenfaces using PCA will be presented.

Step 1: Prepare the data

In this step, the faces constituting the training set (Γi) should be prepared for processing.

Step 2: Subtract the mean

The average matrix Ψ has to be calculated, and then subtracted from the original faces (Γi) and the result stored in the variable Φi:

$$\psi = \frac{1}{M} \sum_{n-1}^{M} \Gamma_n$$

$$\Phi_i = \Gamma_i - \psi$$

Step 3: Calculate the covariance matrix

In the next step the covariance matrix C is calculated according to

$$C = \frac{1}{M} \sum_{n-1}^{M} \Phi_n \Phi_n^T$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

In this step, the eigenvectors (eigenfaces) u_i and the corresponding eigenvalues λi should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e. of length 1. The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here, as it belongs to the standard arsenal of most math programming libraries.

Step 5: Select the principal components

From M eigenvectors (eigenfaces) u_i, only M' should be chosen, which have the highest eigenvalues. The higher the eigenvalue, the more characteristic features of a face does the particular eigenvector describe. Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of characteristic features of the faces. After M' eigenfaces u_i are determined, the"training" phase of the algorithm is finished.

CHAPTER 3

METHODOLOGY/PROJECT WORK

3.1 The Eigenface Technique

The system being developed for Currency Authentication is mainly based on the eigenface technique.

We have focused our research toward developing a sort of pattern recognition scheme that does not depend on excessive geometry and computations. Eigenfaces approach seemed to be an adequate method to be used in currency authentication system.

A step-by-step eigenface approach is summarized from the methodology explained by Turk and Pentland in their paper.

3.1.1 Approaching Eigenface Technique

1. The first step is to obtain a set S of M images. Each image is transformed into a vector of size N and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \ldots\ldots\ldots, \Gamma_M\}$$

2. Next, the images in the training set are normalized.

3. The average image is obtained from the image set.

$$\psi = \frac{1}{M}\sum_{n-1}^{M}\Gamma_n$$

4. Then we find the difference between the input image and the mean image.

$$\Phi_i = \Gamma_i - \psi$$

5. Next we find a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The kth vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T\Phi_n)^2$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & if\ l = k \\ 0 & otherwise \end{cases}$$

6. After that, we obtain the covariance matrix C in the following manner
7.

$$\begin{aligned} C &= \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T \\ &= AA^T \end{aligned}$$

8. From step 6, we construct

$$L = A^T A \text{ where}$$

$$L_{mn} = \Phi_m^T \Phi_n$$

9. Now we find the eigenvectors, v_l and u_l.

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \qquad l = 1,......., \quad M$$

3.1.2 Eigenface Recognition Procedure

1. A new image is transformed into its eigenface components. First, we compare the input image with our mean image and multiply their difference with each eigenvector of the L matrix. Each value would represent a weight and would be saved on a vector.

$$\omega_k = u_k^T (\Gamma - \Psi) \qquad \Omega^T = [\omega_1, \omega_2, ...\omega_M]$$

2. Now we determine which image class provides the best description for the input image. This is done by minimizing the Euclidian distance

$$\varepsilon_k = \left\| \Omega - \Omega_k \right\|^2$$

3. The input image is considered to belong to a class if k is below an established threshold. Then the face image is considered to be a known. If the difference is above the given threshold, but below a second threshold it can be considered as an unknown image. If the input image is above these two thresholds, the image is determined not to belong to class image.

3.2 Images Database

We need to have a database of pre-scanned currency images to perform the authentication through eigenfaces technique. The database consists of images of currencies in different conditions. They vary from new ones to really used, worn out types. Because it will be useful to see the results if we use currency images that differ from each other.
It is conventional and because the computational power will be less, we convert the color images to gray, 8-bit format. This task can be performed with an easy-to-use Wavel Pic2Pic software.

3.3 Tools Used

3.3.1 MATLAB 7.0

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, and data analysis.

We can use MATLAB in a wide range of applications including image processing which applies to our project. And since the project requires us to develop the code based on MATLAB, it was used to perform the currency authentication.

3.3.2 High Quality Scanner

High quality digital scanner was used to get the currency images. It is crucial to capture the details of those images because they vary greatly from each other.

CHAPTER 4

RESULTS AND FINDINGS

4.1 Results

As it was stated in the objectives, a software using MATLAB must be developed to carry out the currency authentication process.

There are some subordinate objectives to successfully implement the tasks and achieve the focal objective:

1. to study the image recognition basics
2. to research different techniques applied to image recognition
3. evaluate the techniques based on their adaptability, stability and reliability
4. to develop the currency authentication algorithm

After reviewing the current techniques and algorithms we have focused our research toward developing a sort of recognition scheme that does not depend on excessive geometry and computations. Eigenfaces approach seemed to be an adequate method to be used in currency authentication system due to its simplicity, speed and learning capability. This approach was originally developed for facial recognition and biometrics. But we are also using eigenfaces to see and test whether it really works for currency authentication too.

4.1.1 Output of Currency Authentication System

The system was implemented using MATLAB 7.0 and tested on an Intel Pentium 4 with 256MB of RAM running Windows XP. This platform should be considered as the optimum hardware requirement since the authentication algorithms could have been modified for increased accuracy on a more powerful testing platform.

10 different images of RM1s were obtained to test the systems. The images can be categorized as "in good condition" and "in bad condition". They would be used as the known images in the authentication system.

1. **The Menu Box**

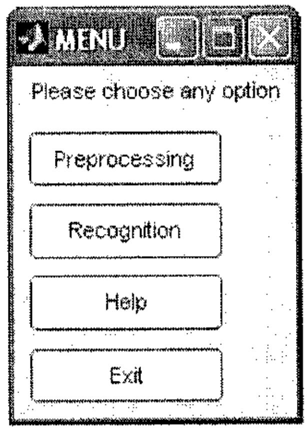


Figure 4.1 Currency Authentication Menu Box

It will appear when the program is executed and will display four options for a user to select.

1) Preprocessing – to read pre-scanned images and select input image, normalize images and calculate the mean image.
2) Recognition – to show eigenfaces, after the input image is selected to calculate its weight, find Euclidian distance and then calculate Min/Max values
3) Help – when pressed will display the information about the program, the following will appear

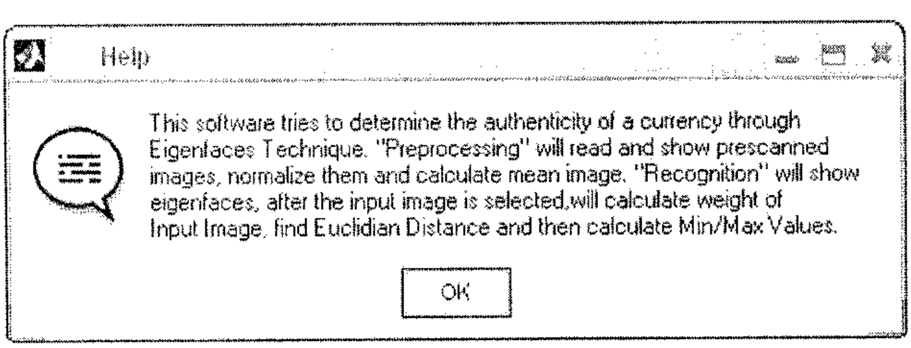


Figure 4.2 Currency Authentication Help

4) Exit – to exit from currency authentication system

2. **Preprocessing**

When "Preprocessing" option is selected a new window will be displayed.

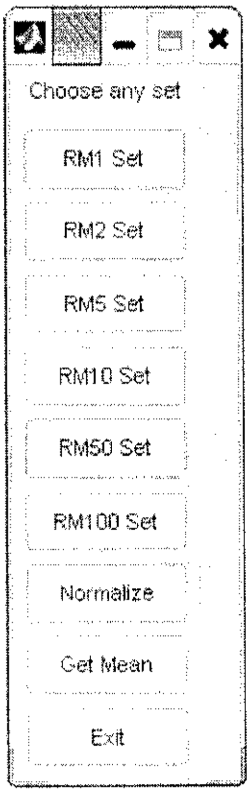


Figure 4.3 The "Preprocessing" Menu Box

This menu has options for each ringgit value. If RM1 Set is selected it will display the images of RM1s in the database. The same goes for each set.

10 different images with variation in conditions were chosen to form the database of known or authentic currencies. This would be our training set. The program codes that would produce and display the training set are shown below. We are using 10 images as samples; hence n in the code represents the number of currency images.

% read and show pre-scanned images ;

```
S=[];  %img matrix
for i=1:n
   str=strcat('C:\MATLAB7\work\test\set\rm1\',...
   int2str(i),'.jpg');  %concatenates two strings that form the name of the image
   eval('img=imread(str);');
   [irow icol]=size(img);   % get the number of rows (N1) and columns (N2)
   temp=reshape(img',irow*icol,1);   %creates a (N1*N2)x1 matrix
   S=[S temp];         %X is a N1*N2xM matrix after finishing the sequence
                       %this is our S
end

if F1 = = 1
   close;
for i=1:n
str = strcat('set\rm1\',...
   num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
   if i==3
      title('Training set','fontsize',18)
   end
   drawnow;
end
```

We are using the RM1 images for the testing of system developed. And referring to the source code in Appendices we can see that F1 represents menu for selection of RM1 images. This portion of code will produce the following figure.

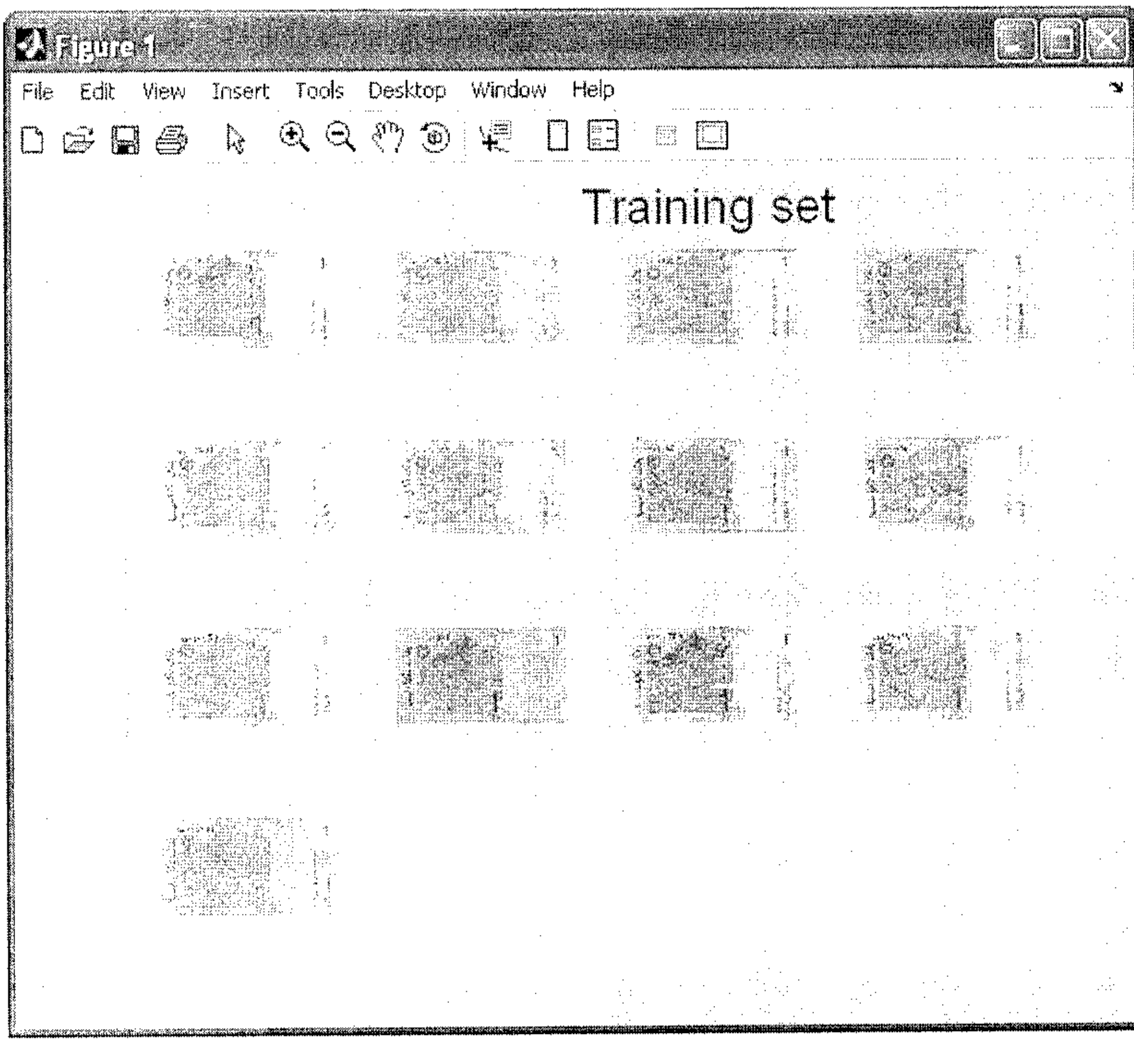


Figure 4.4 Currency images displayed in the Training Set

3. **Normalize**

   Very rarely will an image recognition system be presented with perfect lighting conditions with ambient lighting not causing distinct irregular shadows on a subject. It can be face, license plate recognition or in this case currency recognition and authentication. Even though we are have scanned the images there still could be some degree of lighting invariance. And if a currency authentication system were to be used in real-world environments, under varying light conditions, it must be able to overcome irregular lighting. Therefore normalizing technique was used during the research to provide a degree of lighting invariance. Lightening conditions would adversely affect the performance of the whole system and therefore must be adjusted.

Thus the following lines of code are used to perform normalization.

```
figure(2);
for i=1:n
 str = strcat('set\rm1\',...
 int2str(i), '.bmp');
 img=reshape(S(:,i),icol,irow);
 norm_img=img';
 eval('imwrite(norm_img,str)');
 subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
 imshow(norm_img)
drawnow;
        if i==3
         title('Normalized Training Set','fontsize',18)
                end
        end
```

Output of the above normalization code is shown below.

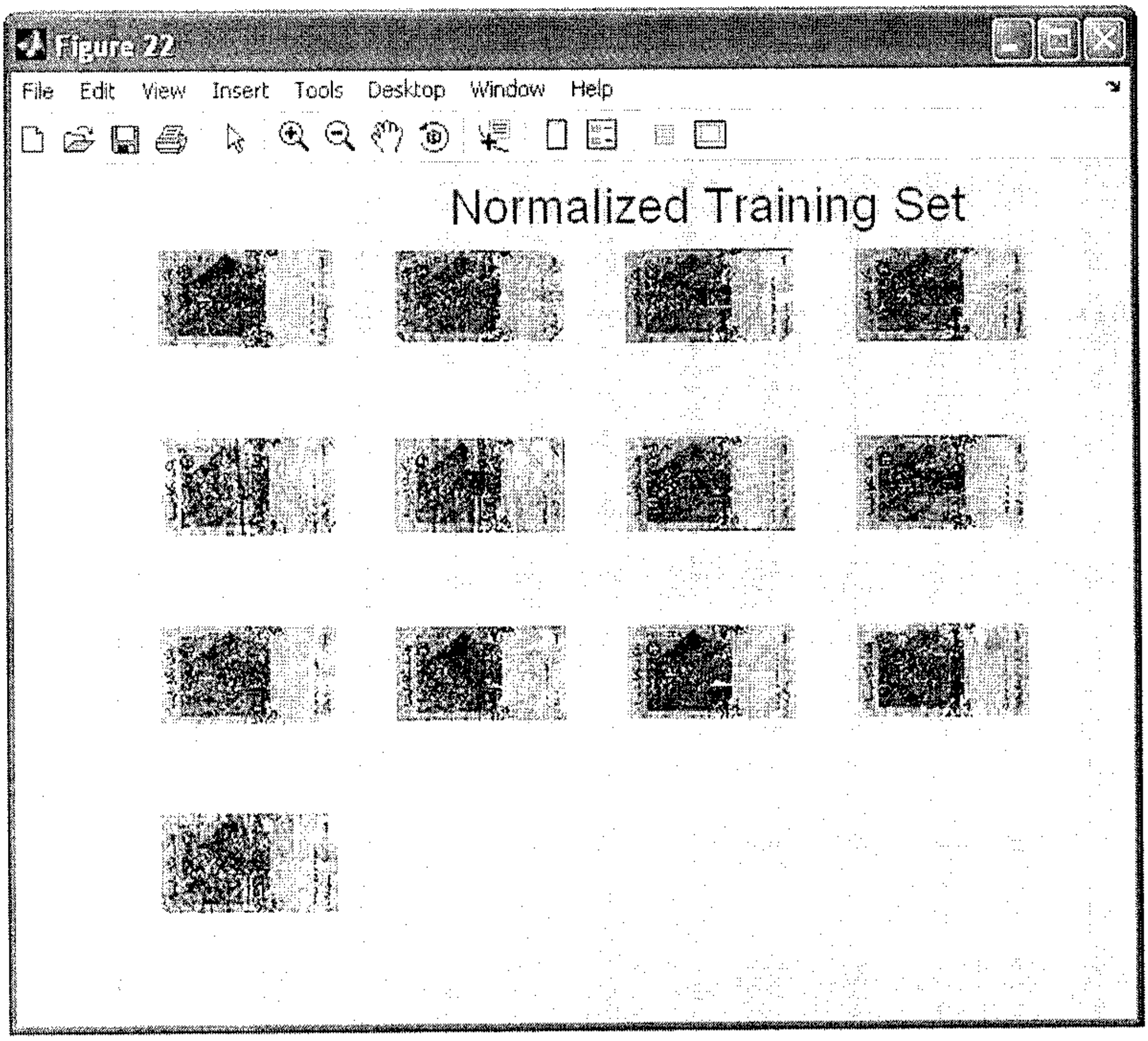


Figure 4.5 Normalized Training Set

4. **Mean Image**

   We need to find the mean image of all currency images in the training set as it will later be used to compare with input image and multiply their difference with each eigenvector.

```
%mean image;
m=mean(S,2);  %obtains the mean of each row instead of each column
tmimg=uint8(m);  %converts to unsigned 8-bit integer. Values range from 0 to
255
img=reshape(tmimg,icol,irow);  %takes the N1*N2x1 vector and creates a
N2xN1 matrix
mean_img=img';  %creates a N1xN2 matrix by transposing the image.
```

```
for i=1:n
    figure(3);
    imshow(mean_img);
    title('Mean Image','fontsize',18)
end
```

The produced mean image is shown below.

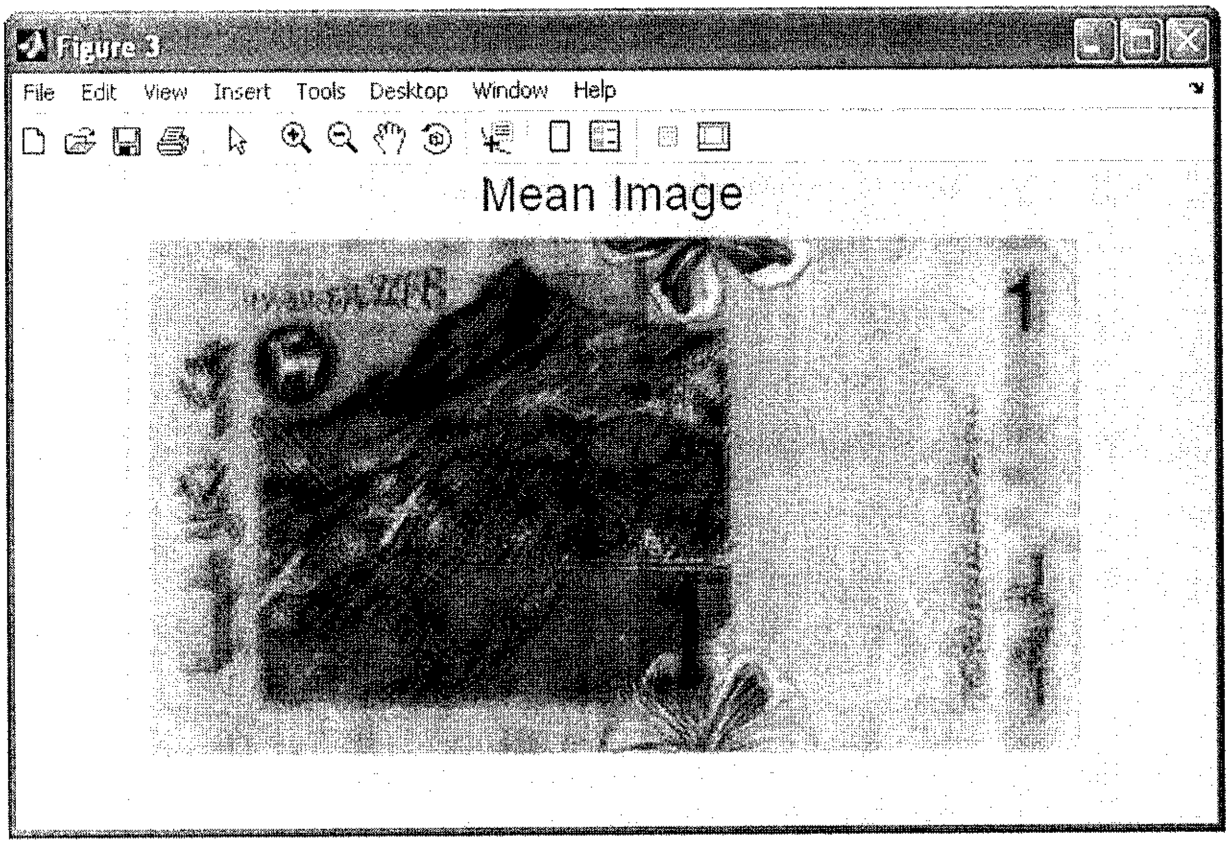


Figure 4.6 The mean image.

After preprocessing is finished user can press "Exit" and can return back to main menu to proceed with rest of the authentication process.

5. **Recognition**

When this option is selected the following window will appear.

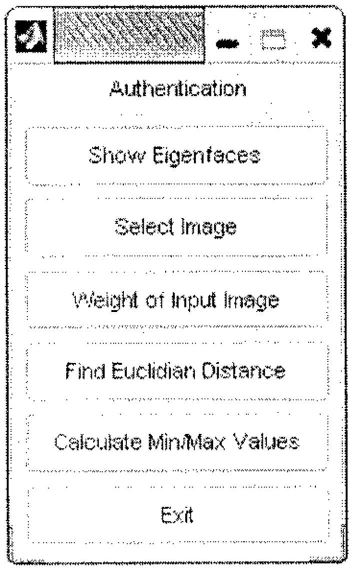


Figure 4.7 The "Recognition" Menu Box

6. **Show Eigenfaces**

Any grey scale face image I(x,y) consisting of a NxN array of intensity values may also be consider as a vector of N2. For example, a typical 100x100 image will have to be transformed into a 10000 dimension vector.

Figure 4.8 A 7x7 image transformed into a 49 dimension vector.

This vector can also be regarded as a point in 10000 dimension space. Therefore, all the images to be recognized can be regarded as points in 10000 dimension space. Recognition using these images is doomed to failure because all currency images are quite similar to one another so all associated vectors are very close to each other in the 10000-dimension space.

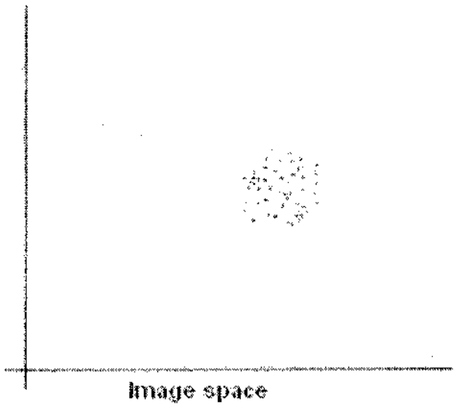


Figure 4.9 Image space.

Therefore classification of a new vector (image) would be a very sensitive process since even a slight change in the image would cause it to be nearer another image than the subject image in the database.

The original variables or vectors which described the subject image are highly correlated. With PCA, we tried to find a better representation of images by finding the specific vectors that account for the distribution of face images. These vectors will define the subspace of face images (sometimes called 'face space').

The vectors that describe faces in face space are eigenfaces. These are in fact the eigenvectors of the covariance matrix of a set of mean subtracted face images. If an image used is 100x100 (therefore associated vectors 10000x1), and if there are 30 sample images in the training set for PCA, the covariance matrix C would be:

$$C = XX^T$$

Here X is a 10000x30 matrix containing the mean subtracted face images. Therefore, the covariance matrices dimensions would be 10000x10000. Calculating this matrix would be an impossible task for most modern computers. This is one of the problems of using PCA in pattern recognition since high dimension vectors are used. A computationally feasible method must be found to calculate eigenfaces. When calculating eigenfaces, the number of faces that are used will always be less than the dimension of the images. Therefore instead of calculating $C=X X^T$, calculate

$$c=X^T X,$$

find c's eigenvectors, and thereby deduce the eigenvectors or eigenfaces of C. In a PCA with 30 images, c will be 30x30 and can easily be calculated. Then the matrix of eigenvectors (v) and matrix of eigenvalues (λ) for c are a vectors and scalars that satisfy,

$$c\,v = \lambda\,v$$

All the eigenvectors, which were calculated, need not be used. Further, dimensionality reduction can be done by sorting the eigenvectors according to their associated eigenvalues and just taking eigenvectors with the largest eigenvalues. These describe the greatest variation. Now that the eigenvectors of c have been found, the eigenvectors of C (eigenfaces) are in the matrix U where,

$$U = X\,v$$

Any face can be described using these eigenfaces. For a detailed description of Principal Component Analysis for face images the reader is encouraged to refer to Turk and Pentland (1991a).

The following figure contains 10 eigenfaces calculated.

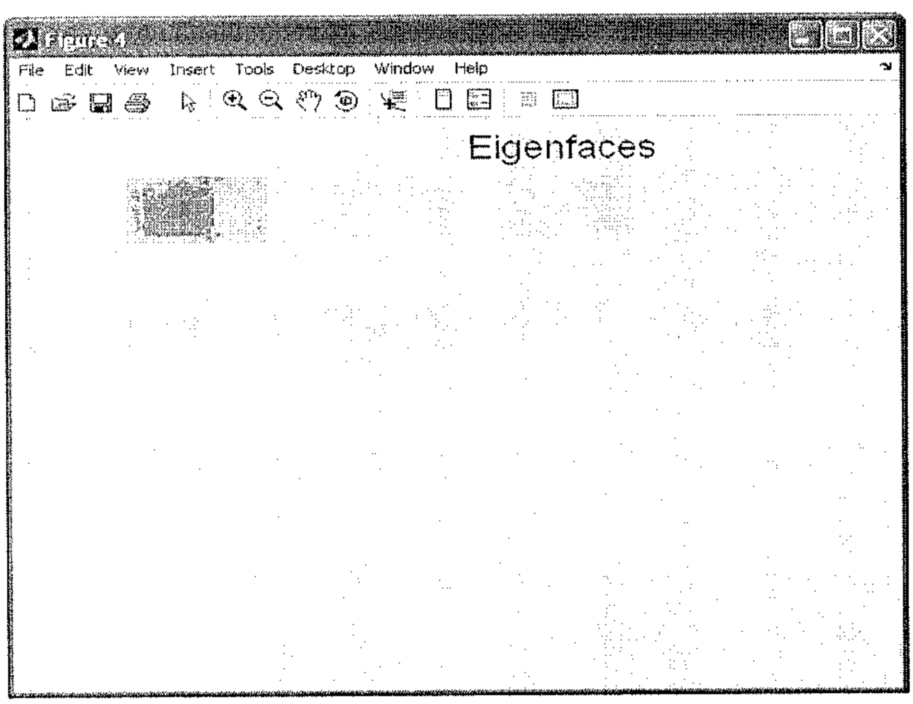


Figure 4.10 Eigenfaces

Increasing the number of eigenvectors (or eigenfaces) that are used to describe an image, as well as normalizing the face space vector, will improve the accuracy and classification performance of the currency authentication system.

7. **Select Image**

Select Image will enable user to select an input image to be compared to stored, known images in the database. User is presented with a choice to select either bitmap (.bmp) or jpg images. Selected images can be known or unknown.

8. **Rest of the options**

As it was explained in the sections 3.1.1 and 3.1.2 we calculate the weight of an input image. After that we determine which image class provides the best description for the input image. This is done by minimizing the Euclidian distance

$$\varepsilon_k = \left\| \Omega - \Omega_k \right\|^2$$

"Find Euclidian Distance" option will calculate it and then print the results on the screen. The input image is considered to belong to a class if k is below an established threshold. Then the image is considered to be a known. If the difference is above the given threshold, but below a second threshold it can be considered as an unknown image. If the input image is above these two thresholds, the image is determined not to belong to class image. These actions are performed when "Calculate Min/Max" values option is selected. It will display the values and results box appears showing values and telling whether input image is "authentic" or "unauthentic".

4.2 Input image selection and results

The transformation of a face from image space (I) to face space (f) involves a simple matrix multiplication. If the average face image is A and U contains the previously calculated eigenfaces,

$$f = U * (I - A)$$

This is done to all the images in the database with known images and to the input image which must be recognized. The possible results when projecting an image into face space are given in the following figure.

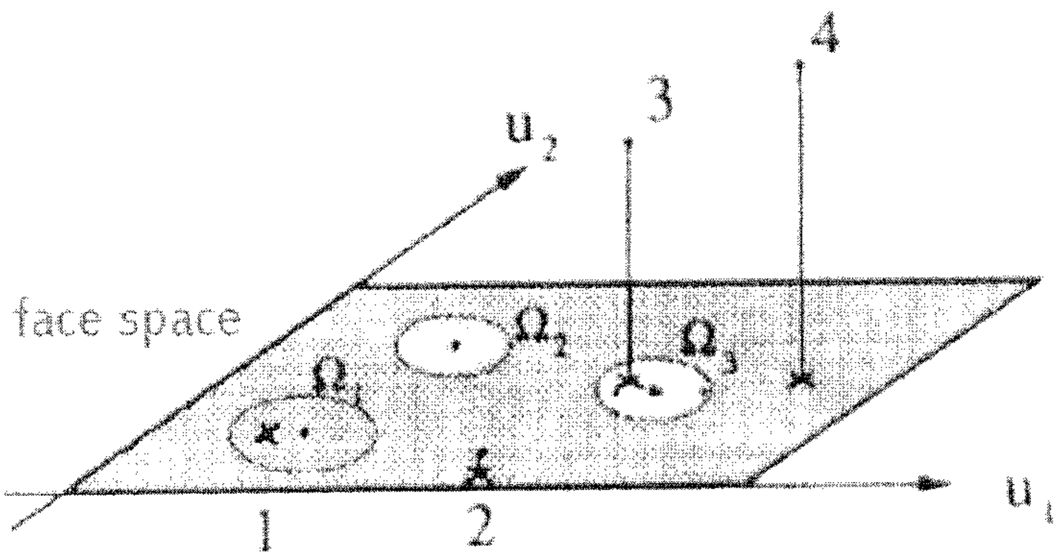


Figure 4.11 The four possible results when projecting an image into faces space. The face space is formed by just two eigenfaces (u_1 and u_2) and contains three known images (Ω_1, Ω_2 and Ω_3)

There are four possibilities:

1. Projected image is "known" and is transformed near a "known" image in the database
2. Projected image is "known" and is not transformed near a "known" image in the database

3. Projected image is not "known" and is transformed near a "known" image in the database

4. Projected image is not a "known" and is not transformed near a "known" image in the database

While it is possible to find the closest known image to the transformed image face by calculating the Euclidean distance to the other vectors, how does one know whether the image that is being transformed actually contains a currency image? Since PCA is a many-to-one transform, several vectors in the image space will map to a point in face space. The problem is that even non-currency images may transform near a known image's faces space vector.

Turk and Pentland (1991a), described a simple way of checking whether an image is actually belongs to a class. This is by transforming an image into face space and then transforming it back or reconstructing into image space. Using the previous notation,

$$I' = UT * U * (I - A)$$

Where I' is the reconstructed image. Then the Euclidean distance between (I'+A) and I can be calculated to find out if I actually is of a currency image. The following figures describe this well.

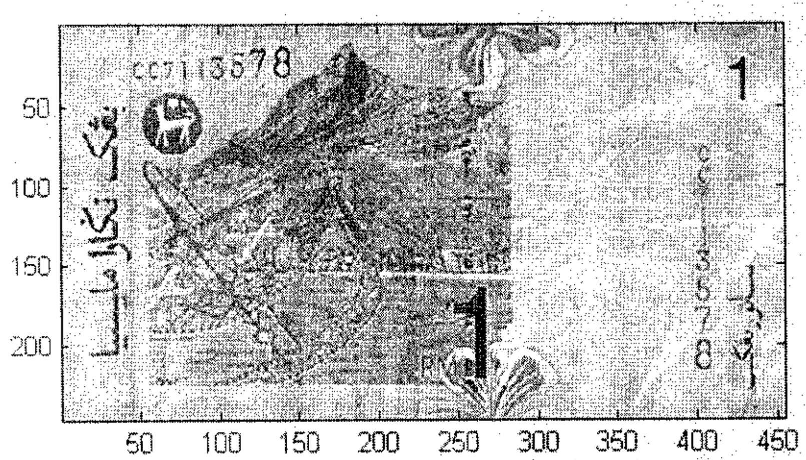


Figure 4.12 Currency image in image space.

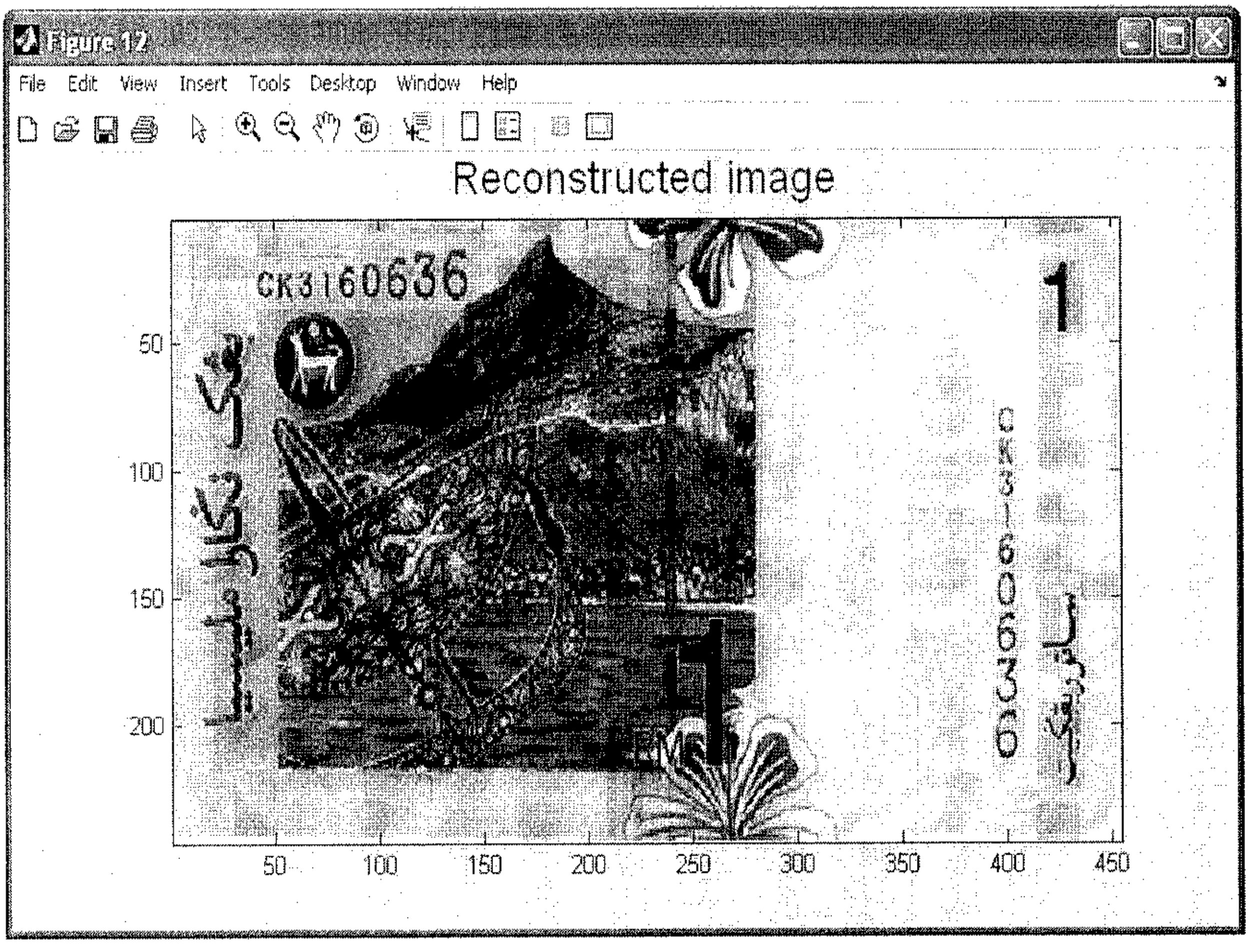


Figure 4.13 Reconstructed currency image in image space

We will select a known image, unknown image and a non-currency image as inputs and check whether the desired results are produced.

First, a known image in the database is selected.

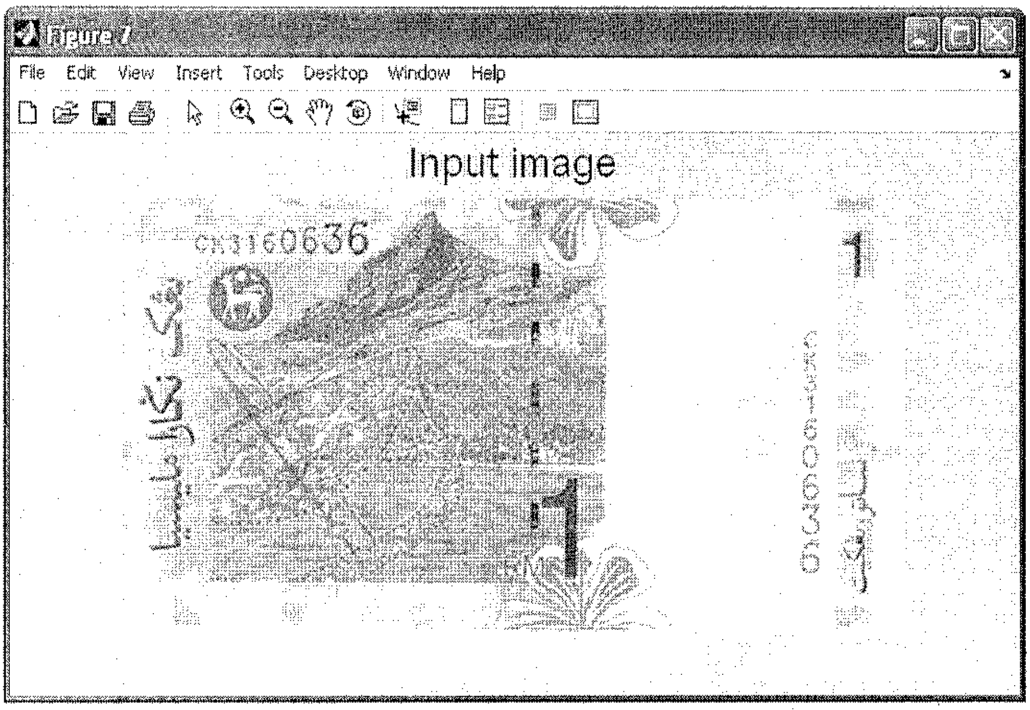


Figure 4.14 Example of a known currency image

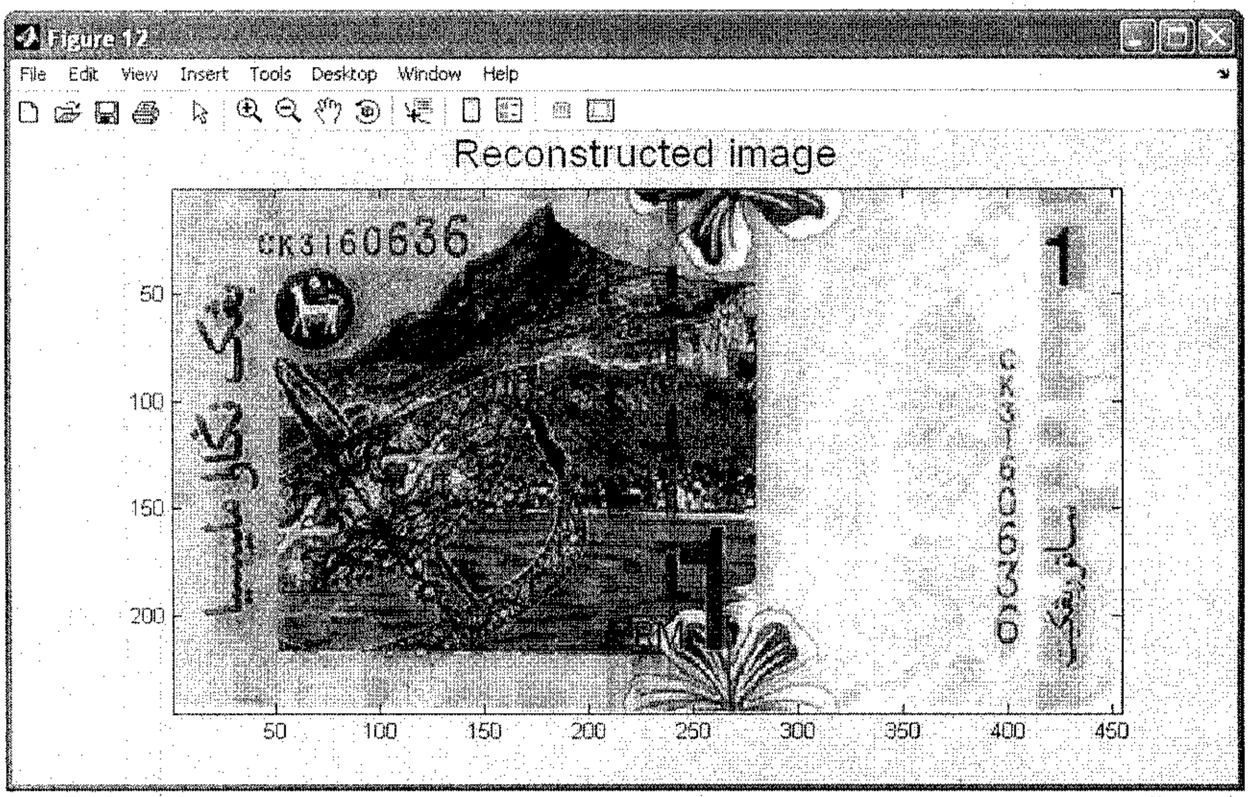


Figure 4.15 Reconstructed currency image

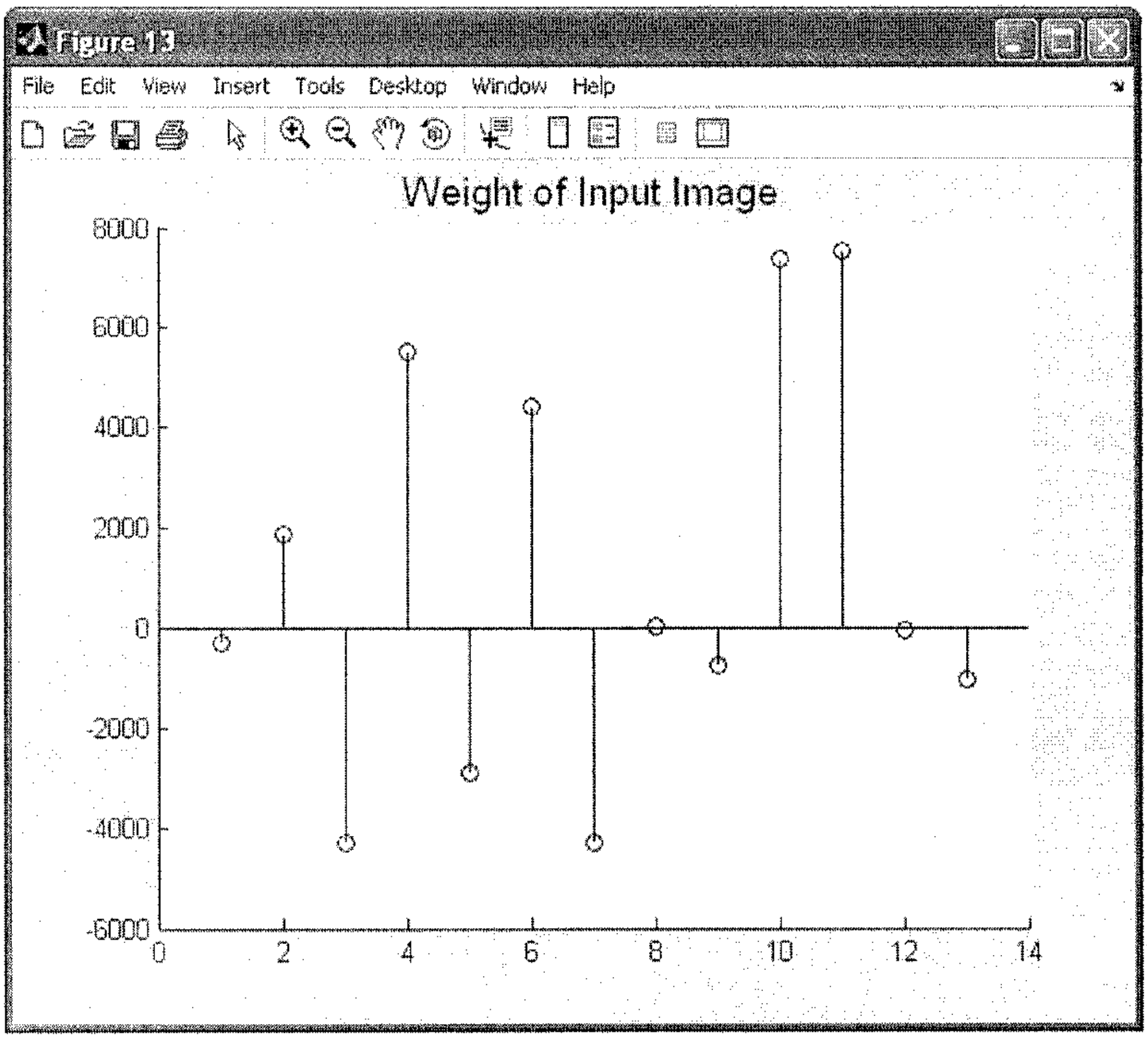


Figure 4.16 Weight of known input image

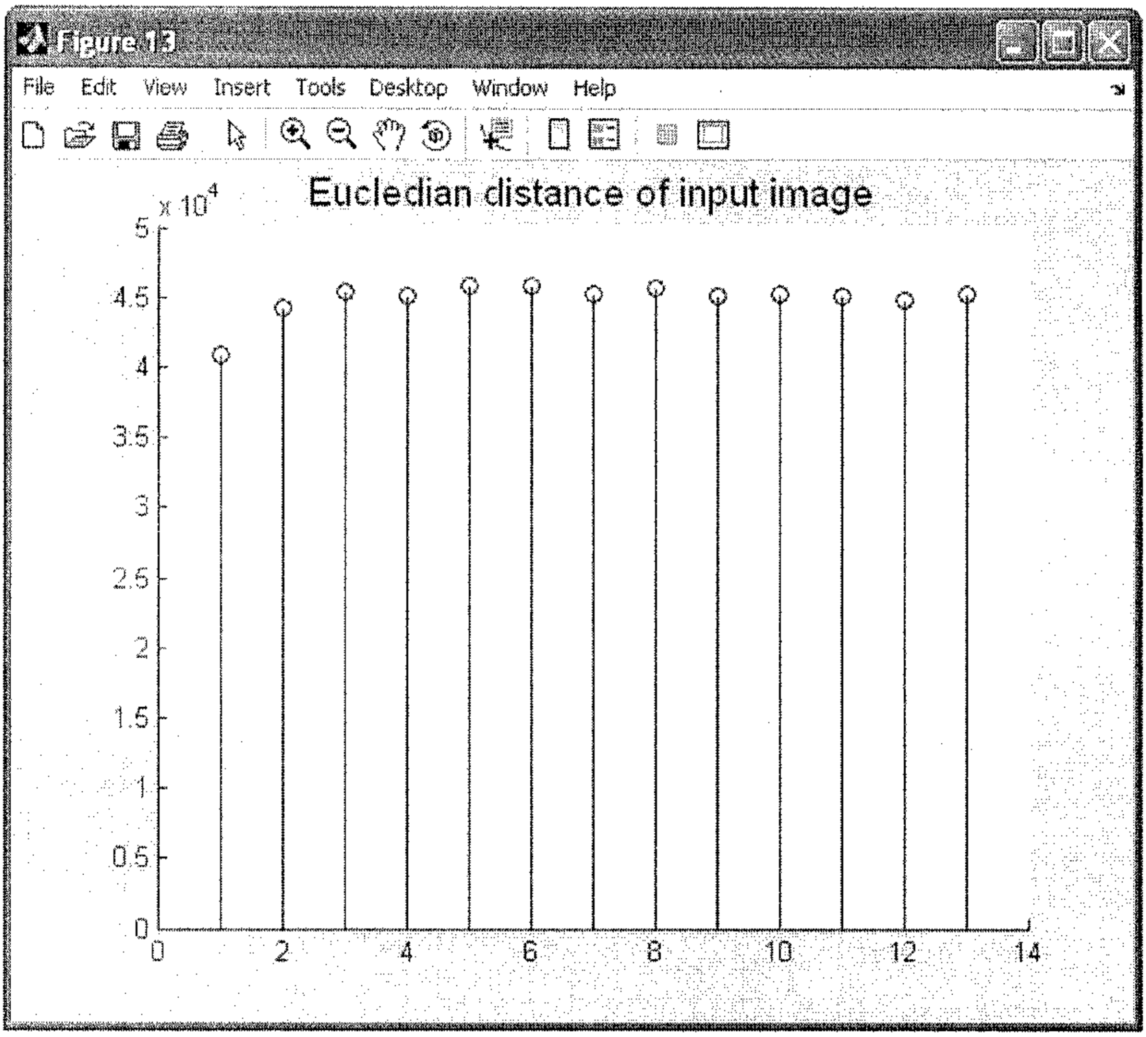


Figure 4.17 Euclidian distance of known input image

4.2.1 Euclidean Distance

4.2.1.1 Introduction

If we want to characterize an image according to the objects it contains (i.e. important feature points in currency image), it is necessary to define a mode of representation which is capable of describing it completely in terms of objects and the relations between them. Significant example can be found in the field of image recognition, pattern matching etc.

Whatever matching, recognizing or comparison techniques are used, if they consist of object-matching the crucial point is still definition of a criterion for comparison. With this aim in mind, it is necessary to identify a set of attributes, or features, such as to:

1) describe the object
2) accept an appropriate measure of similarity, i.e., a distance between the objects
3) act as a base for correct representation of the image

The features used over the years by researchers in this field are often used together in order to obtain a better characterization of an object starting from an approximate description. The simultaneous use of several features allows queries of the type "look for all images with an object having a shape/feature similar to this one and with this composition of color and texture". Information about the shape of an object is very often accompanied by information about its color or texture because finding a definition for the similarity between shapes which corresponds to the human concept of similarity is an extremely difficult problem.

Once a representation for an object has been found by identifying an appropriate set of features, a multidimensional feature space can be identified in which each object corresponds to a point. To be able to search for an object in this space it is therefore necessary to define a distance in it: this corresponds to defining a relationship of equivalence for the set of objects, by means of which it will be possible to establish whether two elements of the set are identical or not (exact matching), and a relationship of order, by which to establish an order of similarity (nearest neighbor). This function has to correspond, in the values supplied, to the criterion of similarity chosen, in the sense that the distance between two objects has to be null if and only if they coincide, and the distance between two different objects has to be proportional to the difference between them. In addition, the measure has to avoid "false dismissals", i.e., it must not allow two objects satisfying the similarity conditions to be considered different on the basis of the distance calculated. Literature provides a variety of techniques one of which is Euclidean distance.

4.2.1.2 Classifying the images

The process of classification of a new (unknown) image Γ_{new} to one of the classes (known) proceeds in two steps.

First, the new image is transformed into its eigenface components. The resulting weights form the weight vector Ω^T_{new}

$$\omega_k = u_k^T(\Gamma_{new} - \Psi) \qquad k = 1....M'$$

$$\Omega^T_{new} = [\omega_1 \omega_2 ....\omega_{M'}]$$

The Euclidean distance between two weight vectors $d(\Omega_i, \Omega_j)$ provides a measure of similarity between the corresponding images *i* and *j*. If the Euclidean distance between Γ_{new} and other images exceeds - on average - some threshold value θ, one can assume that Γ_{new} is no currency image all. $d(\Omega_i, \Omega_j)$ also allows one to construct "clusters" of images such that similar images are assigned to one cluster.

4.2.1.3 Euclidean Distance

Let an arbitrary instance *x* be described by the feature vector

$$x = [a_1(x), a_2(x),......\ a_n(x)]$$

where $a_r(x)$ denotes the value of the *r*th attribute of instance *x*. Then the distance between two instances *xi* and x_j is defined to be $d(x_i, x_j)$

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

Euclidian distance tells us how close the input image is to the images in the training set. Using Euclidean distance we can determine the class of an input image and whether it is a known or unknown image or not even a currency image at all. The determination depends on the maximum and minimum Euclidian distance value. An example Euclidean distance is shown in Figure 4.16.

For experimental purposes two training sets were used in this project. One training set consists of 10 RM1 images.

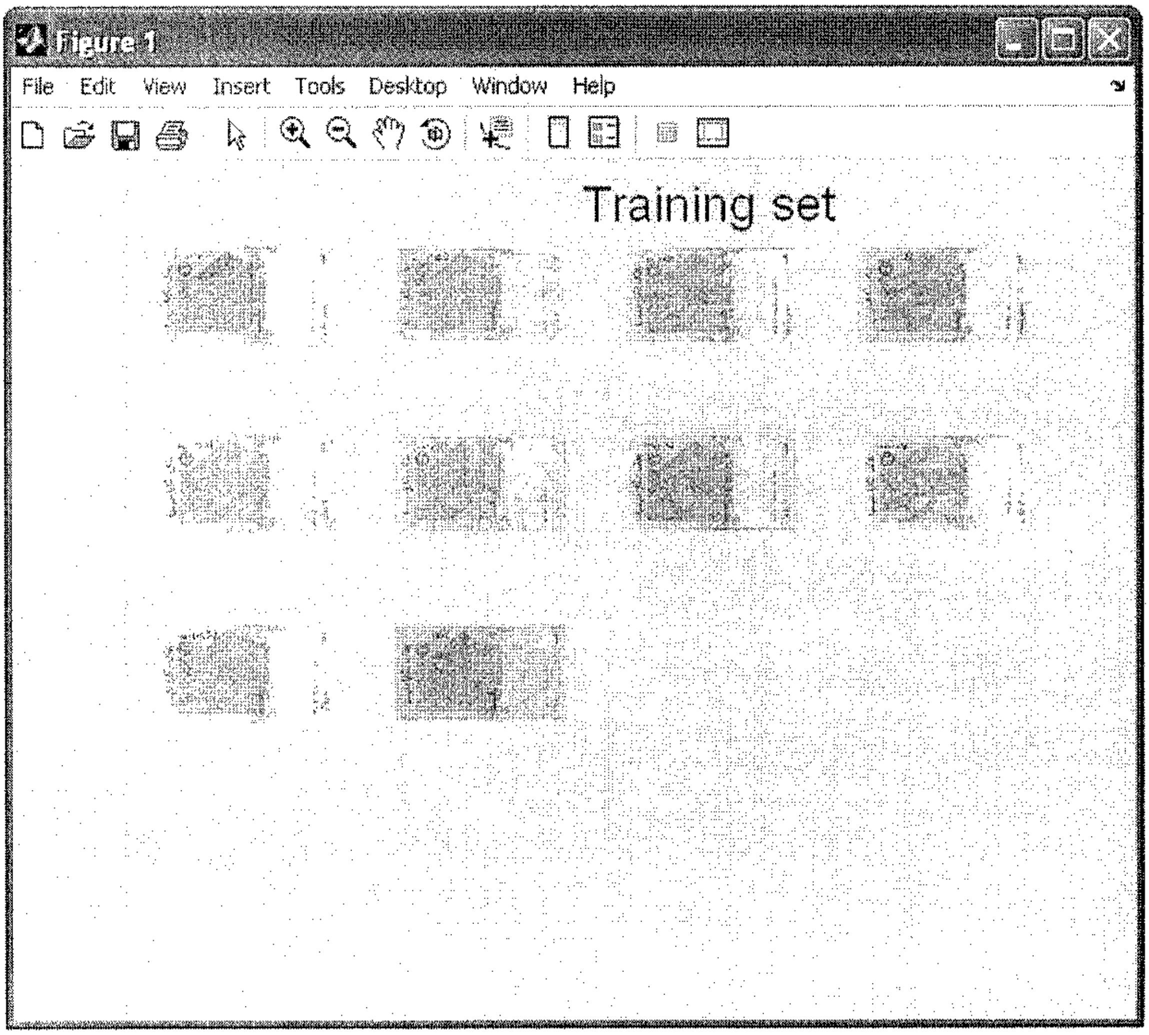


Figure 4.18 Training set with 10 images.

Resulting eigenface images are shown below

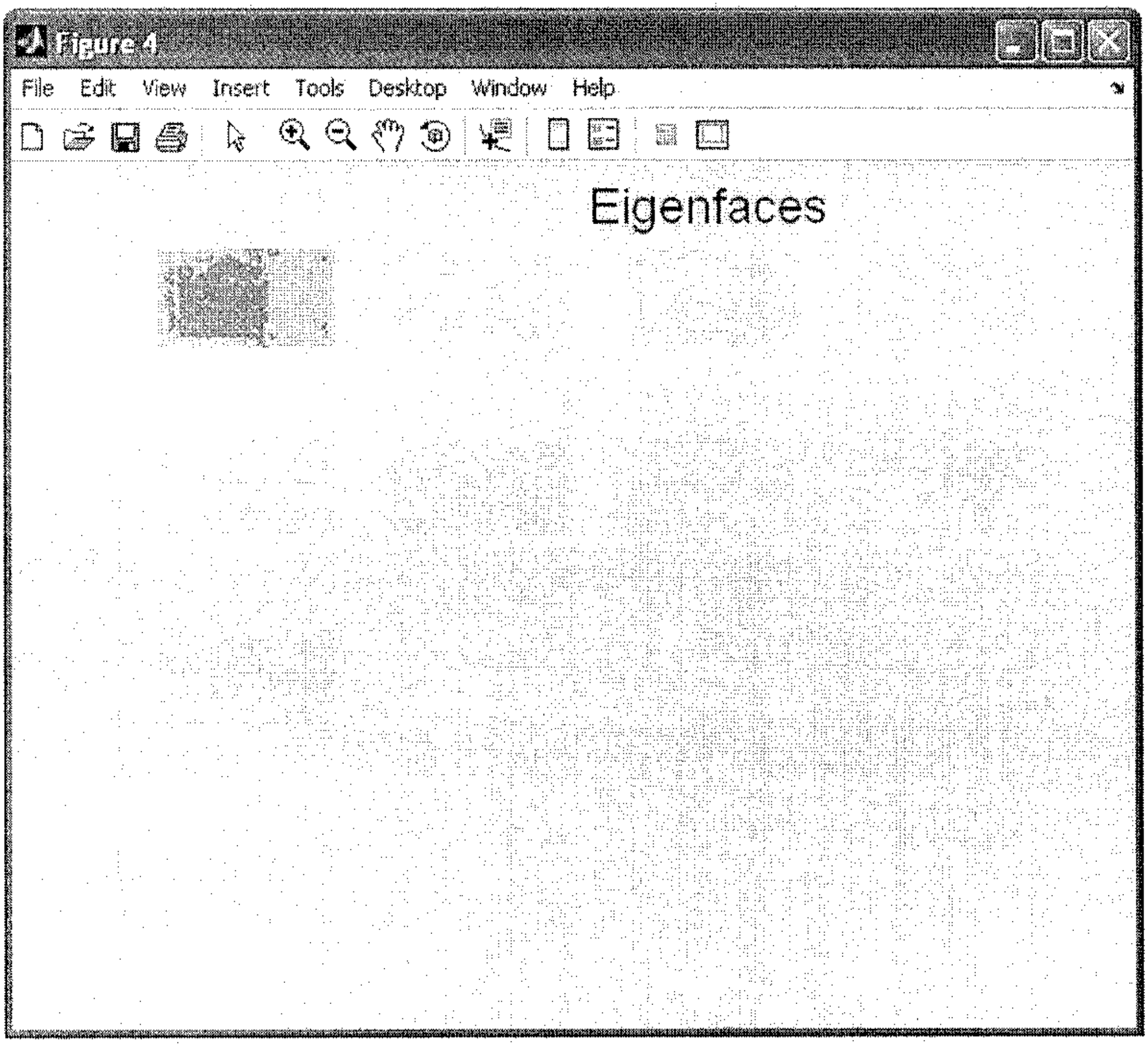


Figure 4.19 Eigenfaces, Training Set 1

Then test input image is selected to proceed with recognition. The input image is shown in Figure 4.13. And resulting Eclidean distance is shown in Figure 4.16. Maximum Value was calculated to be 46032 whereas the Minimum Value was found to be 40867.

To test the program face image is selected next, to see what results it produces.

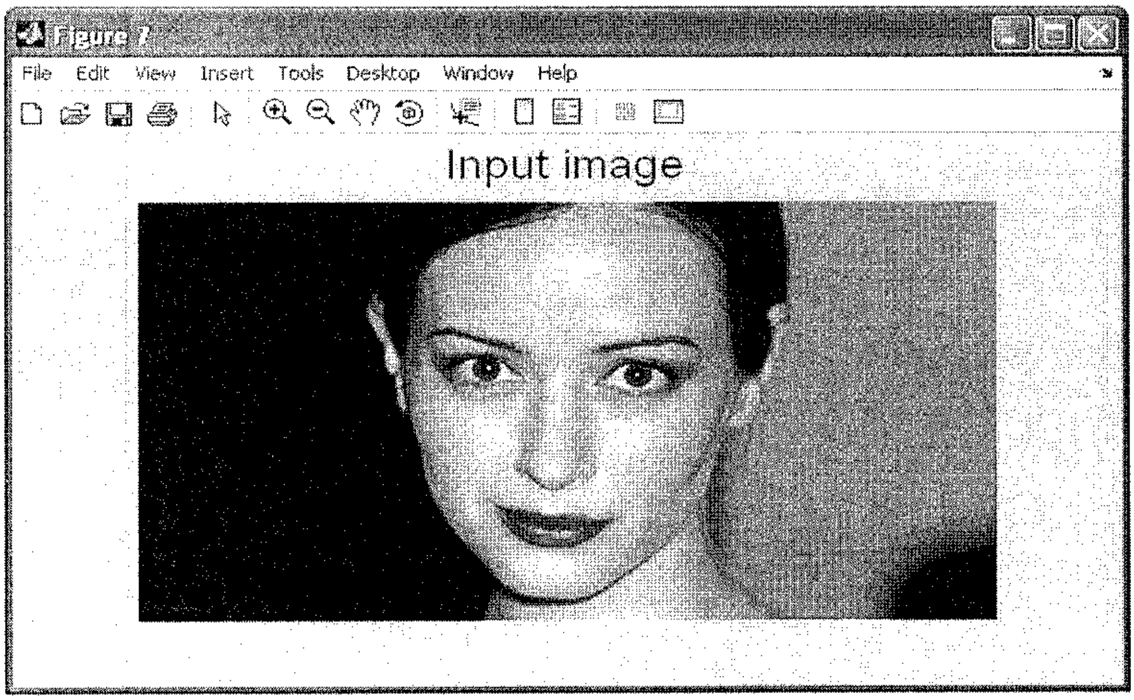


Figure 4.20 Test input face image

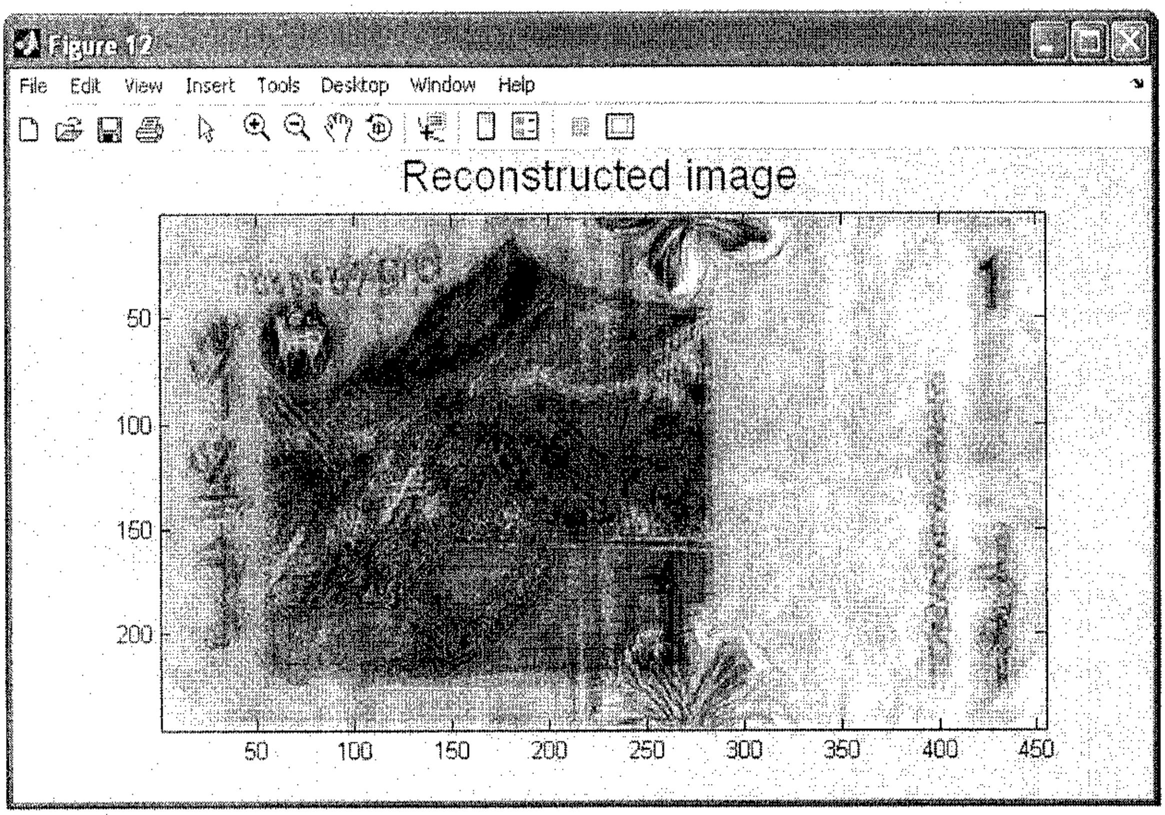


Figure 4.21 The reconstructed image

The input image produced the resulting Euclidean distance with Maximum Value of 53925 and Minimum Value of 51977.

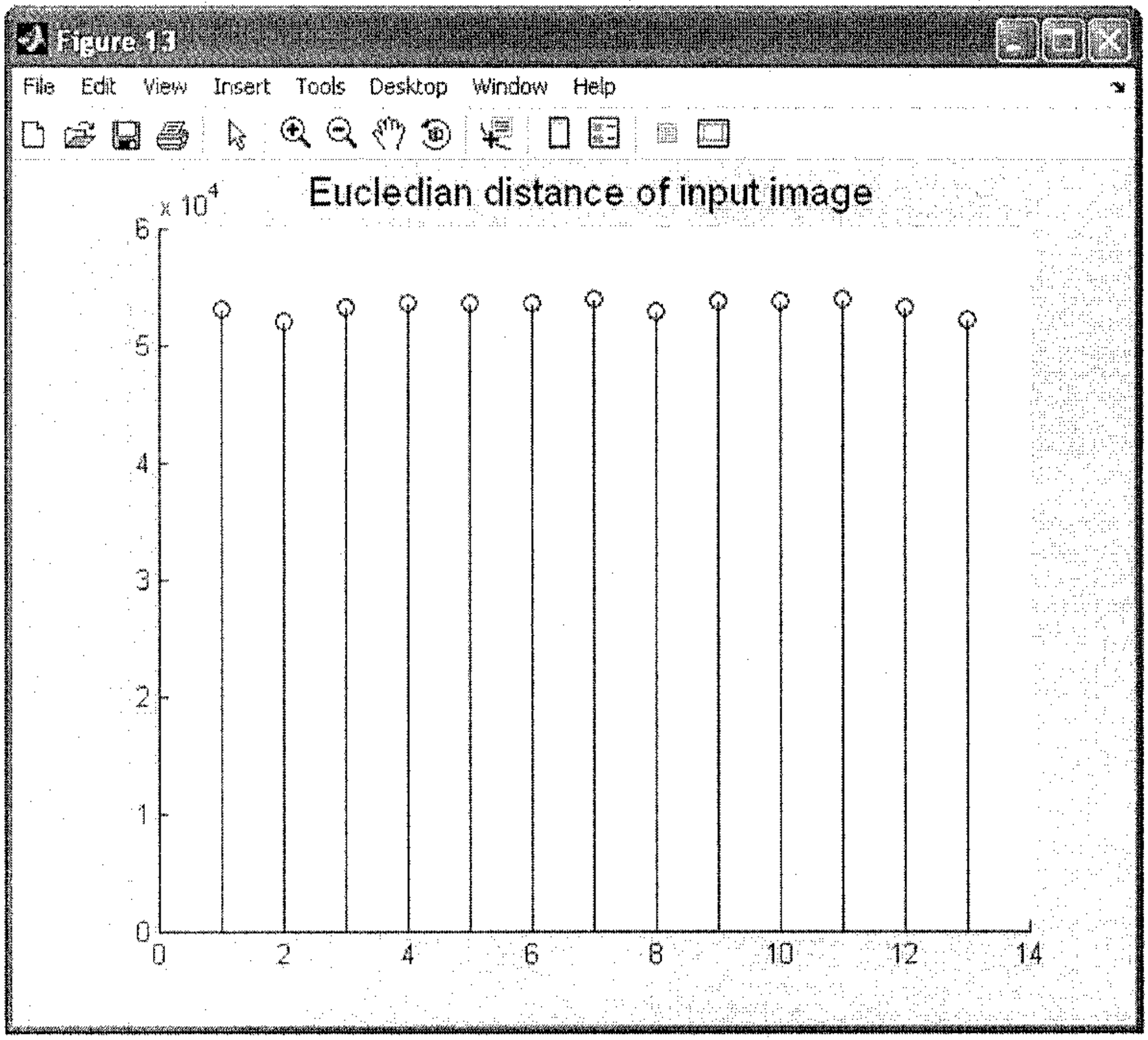


Figure 4.22 Euclidean distance of face input image.

Overall, all the images were selected as input images and the resulting Euclidean distances were recorded. As mentioned before there are two training sets. They are shown in Table 4.1 below.

Table 4.1 Resulting Euclidean distances for various inputs including "counterfeit" image.

Euclidean Distances of Images (13 RM1 Images in Training Set)			Euclidean Distances of Images (10 RM1 Images in Training Set)		Average	
Image	Max Value	Min Value	Max Value	Min Value	Max Value	Min Value
1	45809	40794	46032	40867	45952	41246
2	48126	40620	48024	40634		
3	45973	40727	45578	40750		
4	46502	40877	46252	40891		
5	46789	40921	46567	40943		
6	46907	40840	46699	40861		
7	45088	40699	45271	40756		
8	47514	41119	47398	41149		
9	45854	40981	46228	41088		
10	45615	40602	45777	40663		
11	45360	40740	44008	41387		
12	46898	40979	44619	42977		
13	47554	40798	44923	43239		
counterfeit 1	45721	41073				
counterfeit 2	45619	41060				
counterfeit 3	45784	41139				
counterfeit 4	45606	41055				
counterfeit 5	45575	41001				
counterfeit 6	*45706*	*40821*	*45931*	*40895*		
plane	50853	48946	50647	49071		
frontside	60837	58293	60472	58156		
face	54005	51934	53925	51977		

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Designing a system for automatic image content recognition is a non-trivial task that has been studied for a variety of applications. Computer recognition of specific objects in digital images has been put to use in manufacturing industries, intelligence and surveillance, and image database cataloging to name a few. But perhaps an area involving currency authentication is most important. Digital image application has become important in daily life with the arrival of the digital era. With an ever advancing technology the proliferation of a currency, is also becoming an easy task. A currency can be forged without leaving any traces, using some image processing software.

To detect such malicious tampering a system must be developed using MATLAB. A system that can provide means of ensuring the originality of a currency by detecting any significant malicious manipulations.

There are some subordinate objectives to successfully implement the tasks and achieve the focal objectives:

1. To study the image recognition basics
   Author believes that extensive research was conducted on this area. Basic concepts of pattern recognition were presented in Chapter 2.

2. To research different techniques applied to image recognition
   Since there are many different techniques and algorithms to do image recognition, we needed to have a look at them. As a result of thorough research and studies conducted on images and pattern recognition so far, an effective and fairly reliable techniques and algorithms were evaluated.

3. Evaluate the techniques based on their adaptability, stability and reliability
   To this end, many image recognition techniques and methods have been proposed to solve this problem. Mainly they are such as Eigenfaces, Fisher Linear Discriminant, Neural Networks, and Support Vector Machines. Success has been achieved with each method to varying degrees and complexities.
   Since all of them have their advantages at some point evaluating them based on their flexibility and reliability takes most of the time. Narrowing them down and selecting most important issues has been very challenging indeed.
   As it was stated before in Chapter 2 we have focused our research toward developing a sort of pattern recognition scheme that does not depend on excessive geometry and computations. Eigenfaces approach seemed to be an adequate method to be used in currency authentication system due to its simplicity, speed and learning capability.

4. To develop the currency authentication system based on chosen algorithm
   The following system requirements were identified for the Currency Authentication System:

   - A system to recognize a given currency image.
   - An implemented system must display a high degree of lighting invariance.
   - A system must posses near real-time performance.

Currency Authentication System was developed which can be used to verify a currency image when compared to images in the database. Author believes that the main objective was indeed achieved. The system was developed using the Eigenfaces Technique which was chosen after evaluating the current methods. The process for this technique can be described like this. First, we get the pre-scanned currency images to be our training set. In this project two training sets were used to see how Euclidean distance changes as number of samples increases. One set consists of 10 different RM1 images while the second set has 13 images.

Lightening conditions would adversely affect the performance of the whole system and therefore must be adjusted. Therefore normalization technique is used to normalize images and to provide a degree of lighting invariance. After that the mean image is obtained. An image is transformed into its eigenface components afterwards. Different input images are selected to see what Euclidean distance is produced by the program. As seen in Table 4.1 all the known images as well as image of a face, plane and most importantly counterfeit currency image are introduced as inputs. All of them produced different results. Range of Minimum Value for a set with 10 images is from 40634 to 41149 while Maximum Value ranges from 45271 to 48024. Minimum Value ranges from 40602 to 41119 and Maximum Value ranges between 45088 and 48126 for set having 13 images.

When counterfeit image is selected as input image the resulting Euclidean distance varies very slightly. In fact the Euclidean distance of counterfeit image shows almost no difference. Although Eigenfaces approach was proved to be very useful in face recognition, it produces mixed results for currency image recognition and authentication. Because the range of Euclidean distance can not be determined without strong results. Counterfeit currency should be really poor or else highly skillfully forged and tampered currency will pass as authentic and it would undermine the effectiveness of the program.

5.2 Recommendations and future work

To improve the accuracy of the results more samples of currency images in varying degrees of quality should be obtained. Because when using Eigenfaces technique, increased numbers of eigenvectors or eigenfaces that are used to describe an image will improve the accuracy and classification performance of the Currency Authentication System. Having different training sets could also produce dramatic change in the threshold values hence affecting the results. But perhaps most important factor is number of samples provided in training set. Eigenfaces technique was originally developed to recognize faces which would require a set with hundreds if not thousands of face images. The results would dramatically change when higher number of sample images is used.

Currency authentication system will work best when specific feature points are focused more on a currency image. Those feature points can be determined through effective feature selection process which was also discussed earlier in the project.

For the time being, the currency is categorized as "known" or "unknown". System would be more effective if not only it determines whether a currency if forged but also the points that were tampered. A new function that determines such points should be added in the future for this project to be more efficient.

The project could be expanded more to handle the recommended situations

REFERENCES

Ilker Atalay, "Face Recognition using Eigenfaces, " M. Sc. THESIS, January, 1996.

Turk, M., and Pentland, A., "Eigenfaces for image recognition," Journal of Cognitive Neuroscience, Vol. 3, pp. 71-86, 1991.

Dimitri Pissarenko, "Eigenface-based facial recognition", December 1, 2002

M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," 1996 International Conf. on Image Processing, Lausanne, Switzerland, 1996.

Lalendra Sumitha Balasuriya, Department of Statistics and Computer Science,

University of Colombo, "Frontal View Human Face Detection and Recognition", May 2000

Giancarlo Iannizzotto, Dipartimento di Matematica, University of Messina, C.da Papardo, Salita Sperone, 98166 Messina, Italy, Antonio Puliafito, Istituto di Informatica, University of Catania, Viale A. Doria 6, 95025 Catania, Italy, Lorenzo Vita, Dipartimento di Matematica, University of Messina, C.da Papardo, Salita Sperone, 98166 Messina, Italy, Using the Median Distance to Compare Object Shapes in Content-Based Image Retrieval

Nasir Memon , Poorvi Vora, Boon-Lock Yeo and Minerva Yeung, "Distortion Bounded Authentication Techniques", Jan 2000, In Proceedings of the SPIE, Security and Watermarking of Multimedia Contents II, vol.3971, pp. 164-74, 24-26

http://www.springerlink.com

http://www.pages.drexel.edu

APPENDICES

Appendix A - Code Listing

```
% Currency Authentication using MATLAB
clear all
close all
clc
% number of images on your training set.
global n
n=13; %change this if number of samples is more

%Chosen std and mean.
%It can be any number that it is close to the std and mean of most of the images.
um=100;
ustd=80;

%read and show images(bmp);
S=[]; %img matrix
for i=1:n
   str=strcat('C:\MATLAB7\work\test\set\rm1\',...
   int2str(i),'.jpg'); %concatenates two strings that form the name of the image
   eval('img=imread(str);');
   [irow icol]=size(img); % get the number of rows (N1) and columns (N2)
   temp=reshape(img',irow*icol,1); %creates a (N1*N2)x1 matrix
   S=[S temp]; %X is a N1*N2xM matrix after finishing the sequence
            %this is our S
end


%Here we change the mean and std of all images. We normalize all images.
%This is done to reduce the error due to lighting conditions.
for i=1:size(S,2)
   temp=double(S(:,i));
   m=mean(temp);
   st=std(temp);
   S(:,i)=(temp-m)*ustd/st+um;
end

%mean image;
m=mean(S,2); %obtains the mean of each row instead of each column
tmimg=uint8(m); %converts to unsigned 8-bit integer. Values range from 0 to 255
img=reshape(tmimg,icol,irow); %takes the N1*N2x1 vector and creates a N2xN1
matrix
mean_img=img'; %creates a N1xN2 matrix by transposing the image.
```

```
% Change image for manipulation
dbx=[];  % A matrix
for i=1:n
   temp=double(S(:,i));
   dbx=[dbx temp];
end

%Covariance matrix C=A'A, L=AA'
A=dbx';
L=A*A';
% vv are the eigenvector for L
% dd are the eigenvalue for both L=dbx'*dbx and C=dbx*dbx';
[vv dd]=eig(L);
% Sort and eliminate those whose eigenvalue is zero
v=[];
d=[];
for i=1:size(vv,2)
   if(dd(i,i)>1e-4)
      v=[v vv(:,i)];
      d=[d dd(i,i)];
   end
end

%sort,  will return an ascending sequence
[B index]=sort(d);
ind=zeros(size(index));
dtemp=zeros(size(index));
vtemp=zeros(size(v));
len=length(index);
for i=1:len
   dtemp(i)=B(len+1-i);
   ind(i)=len+1-index(i);
   vtemp(:,ind(i))=v(:,i);
end
d=dtemp;
v=vtemp;


%Normalization of eigenvectors
for i=1:size(v,2)       %access each column
  kk=v(:,i);
  temp=sqrt(sum(kk.^2));
  v(:,i)=v(:,i)./temp;
end

%Eigenvectors of C matrix
u=[];
```

```
for i=1:size(v,2)
   temp=sqrt(d(i));
   u=[u (dbx*v(:,i))./temp];
end

%Normalization of eigenvectors
for i=1:size(u,2)
  kk=u(:,i);
  temp=sqrt(sum(kk.^2));
        u(:,i)=u(:,i)./temp;
end


% Select from menu Preprocessing, Recognition or Help
F=0;
possibility=4;
while F~=possibility,
F = MENU('Please choose any option','Preprocessing','Recognition','Help','Exit');
if F==1
   %close all
   F1=0;
   possibility1=9;
   while F1~=possibility1;
%% Select Test Set
F1 = MENU('Choose any set','RM1 Set','RM2 Set','RM5 Set','RM10 Set','RM50
Set','RM100 Set','Normalize','Get Mean','Exit');
if F1 == 1
   close;
for i=1:n
str = strcat('set\rm1\',...
   num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
   if i==3
      title('Training set','fontsize',18)
   end
   drawnow;
end
elseif F1 == 2
for i=1:n
str = strcat('set\mio\',...
   num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
   if i==3
```

```
        title('Training set','fontsize',18)
    end
    drawnow;
end
elseif F1 == 3
for i=1:n
str = strcat('set\rm5\',...
    num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
    if i==3
        title('Training set','fontsize',18)
    end
    drawnow;
end
elseif F1 == 4
for i=1:n
str = strcat('set\rm10\',...
    num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
    if i==3
        title('Training set','fontsize',18)
    end
    drawnow;
end
elseif F1 == 5
for i=1:M
str = strcat('set\rm50\',...
    num2str(i), '.bmp');
figure(1);
eval('img=imread(str);');
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
    if i==3
        title('Training set','fontsize',18)
    end
    drawnow;
end
elseif F1 == 6
for i=1:n
str = strcat('set\rm100\',...
    num2str(i), '.jpg');
figure(1);
eval('img=imread(str);');
```

```
subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
imshow(img)
  if i==3
    title('Training set','fontsize',18)
  end
  drawnow;
end
elseif F1==7
  figure(22);
  for i=1:n
  str = strcat('set\rm1\',...
  int2str(i), '.bmp');
  img=reshape(S(:,i),icol,irow);
  norm_img=img';
  eval('imwrite(norm_img,str)');
  subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
  imshow(norm_img)
  drawnow;
   if i==3
    title('Normalized Training Set','fontsize',18)
  end
end

elseif F1==8
  for i=1:n
    figure(3);
    imshow(mean_img);
    title('Mean Image','fontsize',18)
  end
elseif F1==9
  close all;
end
  end

elseif F==2
  %close all
  F2=0;
  possibility2=6;
  while F2~=possibility2;
%% Select Test Set
F2 = MENU('Authentication','Show Eigenfaces','Select Image','Weight of Input
Image','Find Euclidian Distance','Calculate Min/Max Values','Exit');
if F2==1
% show eigenfaces;
figure(4);
for i=1:size(u,2)
   img=reshape(u(:,i),icol,irow);
   eigen_img=img';
   eigen_img=histeq(eigen_img,255);
```

```
   subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
   imshow(eigen_img)
   drawnow;
   if i==3
      title('Eigenfaces','fontsize',18)
   end
end

% Find the weight of each face in the training set.
omega = [];
for h=1:size(dbx,2)
   WW=[];
   for i=1:size(u,2)
      t = u(:,i)';
      WeightOfImage = dot(t,dbx(:,h)');
      WW = [WW; WeightOfImage];
   end
   omega = [omega WW];
end

elseif F2==2
   figure(7);
[filename, pathname] = uigetfile('*.bmp;*.jpeg;*.jpg','Please select currency image');
if filename~=0
inputimage=imread(strcat(pathname,filename));
imshow(inputimage);
title('Input image','fontsize',18)
end

elseif F2==3
inimage=reshape(double(inputimage)',irow*icol,1);
temp=inimage;
me=mean(temp);
st=std(temp);
temp=(temp-me)*ustd/st+um;
NormImage = temp;
Difference = temp-m;

p = [];
aa=size(u,2);
for i = 1:aa
   pare = dot(NormImage,u(:,i));
   p = [p; pare];
end
ReshapedImage = m + u(:,1:aa)*p;    %m is the mean image, u is the eigenvector
ReshapedImage = reshape(ReshapedImage,icol,irow);
ReshapedImage = ReshapedImage';
%show the reconstructed image.
figure(12)
```

```
imagesc(ReshapedImage); colormap('gray');
title('Reconstructed image','fontsize',18)

InImWeight = [];
for i=1:size(u,2)
   t = u(:,i)';
   WeightOfInputImage = dot(t,Difference');
   InImWeight = [InImWeight; WeightOfInputImage];
end

ll = 1:n;
figure(13)
stem(ll,InImWeight)
title('Weight of Input Image','fontsize',14)
end


if F2==4
% Find Euclidean distance
e=[];
for i=1:size(omega,2)
   q = omega(:,i);
   DiffWeight = InImWeight-q;
   mag = norm(DiffWeight);
   e = [e mag];
end

kk = 1:size(e,2);
figure(13)
stem(kk,e)
title('Eucledian distance of input image','fontsize',14)
end

if F2==5
MaximumValue=max(e)
MinimumValue=min(e)

%Set threshold values
if(MaximumValue>47700)&&(MinimumValue>41000)&&(MinimumValue<42000)
   errordlg(strcat('No Match! The input image is unknown.'));

end
if (MaximumValue>48000)&& (MinimumValue>42000)
   errordlg('No Match! The input image is not currency image.');

end
if (MaximumValue>46000) && (MaximumValue<47700) &&
(MinimumValue<42000)&&(MinimumValue>41000)
   helpdlg(strcat('The input image is a match.'));
```

```
end
end

if F2==6
   close all;
end
   end

elseif F==3
   message1='This software tries to determine the authenticity of a currency through
Eigenfaces Technique. ';
   message2=' "Preprocessing" will read and show prescanned images, normalize them
and calculate mean image. ';
   message3=' "Recognition" will show eigenfaces, after the input image is selected,will
calculate weight of Input Image, ';
   message4='find Euclidian Distance and then calculate Min/Max Values.';
   helpdlg(strcat(message1,message2,message3,message4),'Help');
end

if F==4
   clc;
   close all;
end
end
```

Appendix B – Results

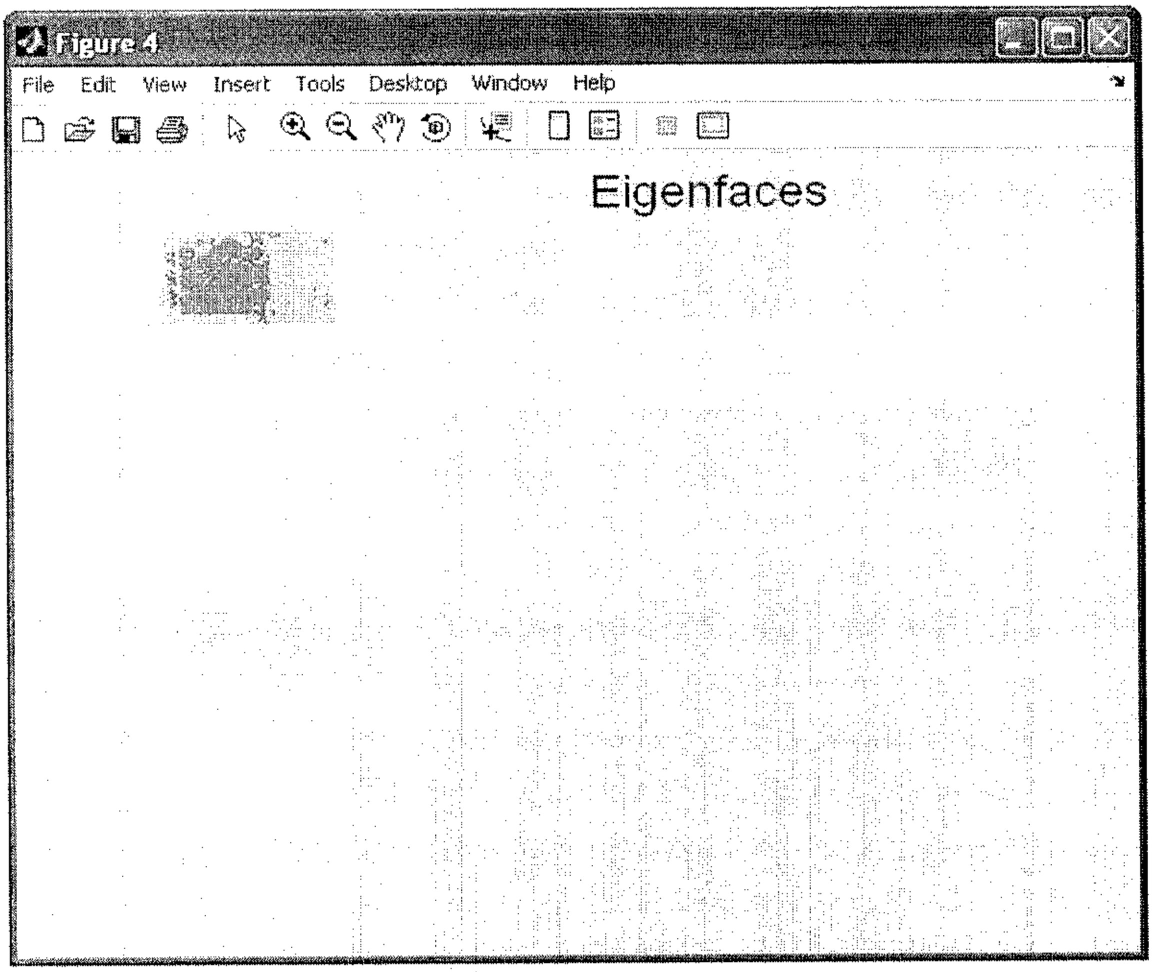


Figure 6.1 Training Set 1 eigenfaces

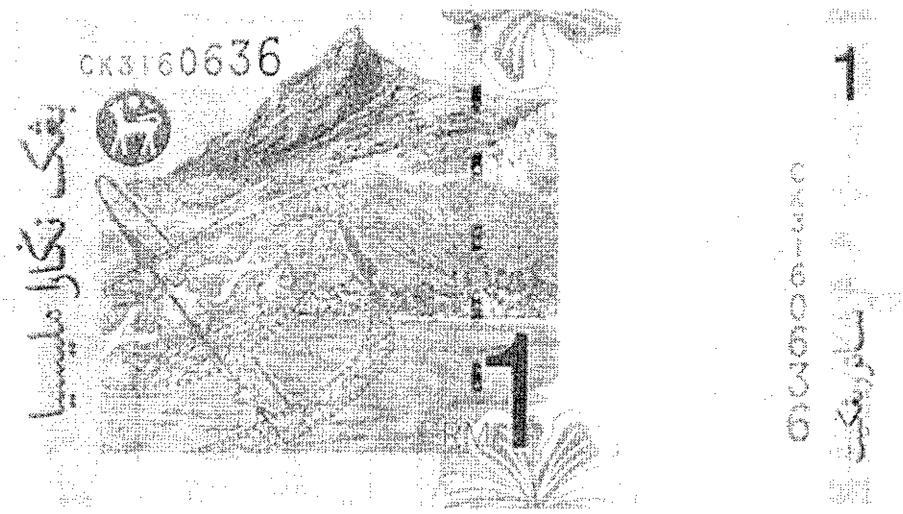


Figure 6.1 Input image 1

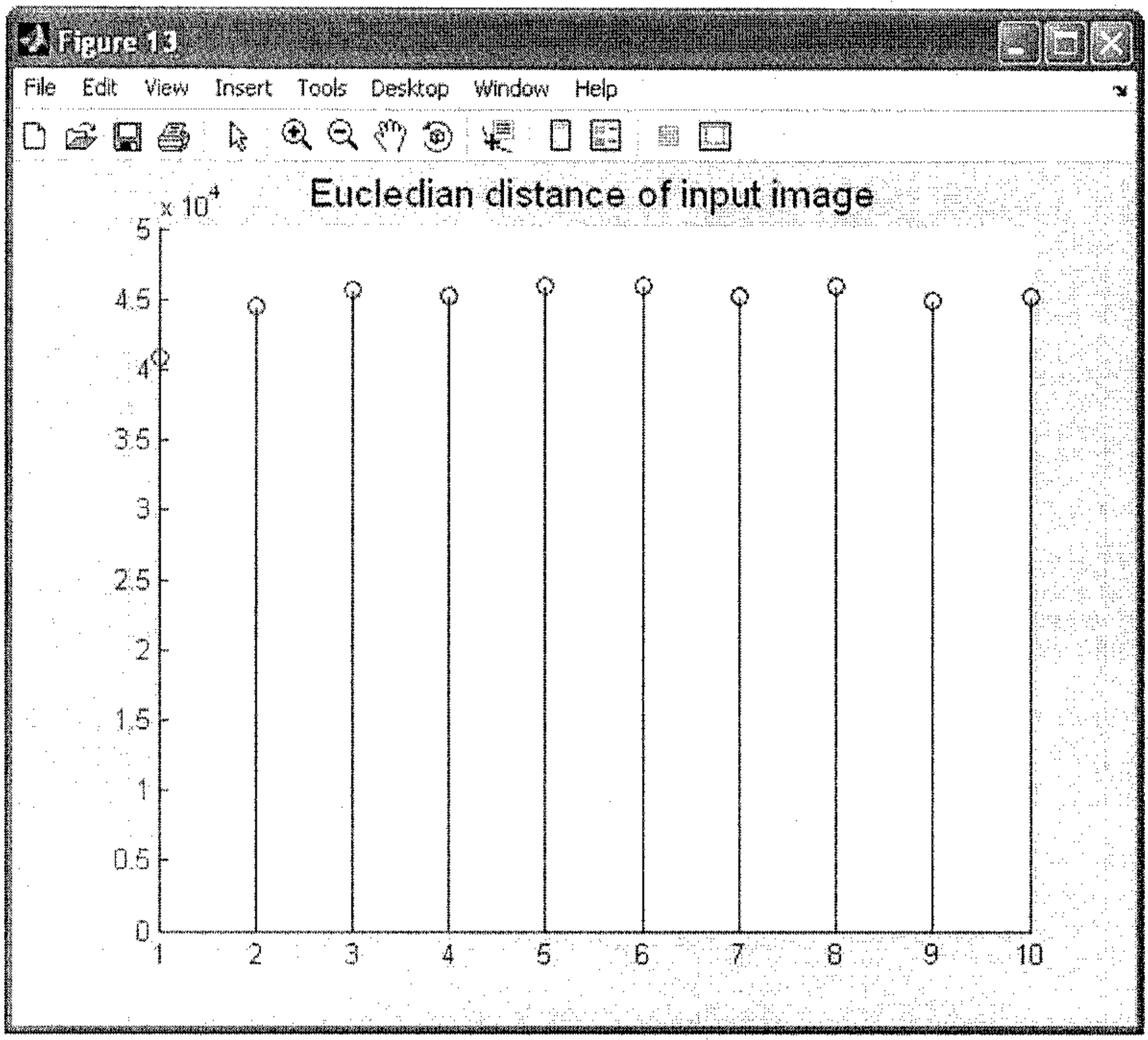


Figure 6.3 Euclidean distance of input image 1

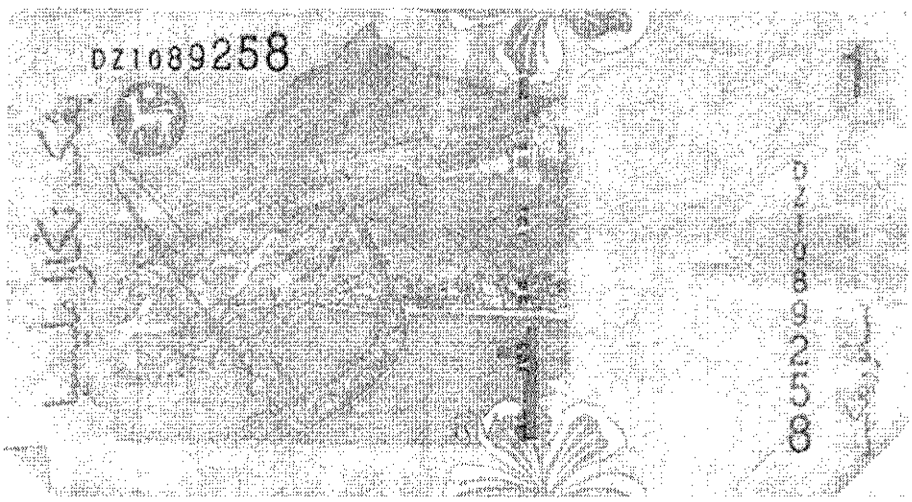


Figure 6.4 Input image 2

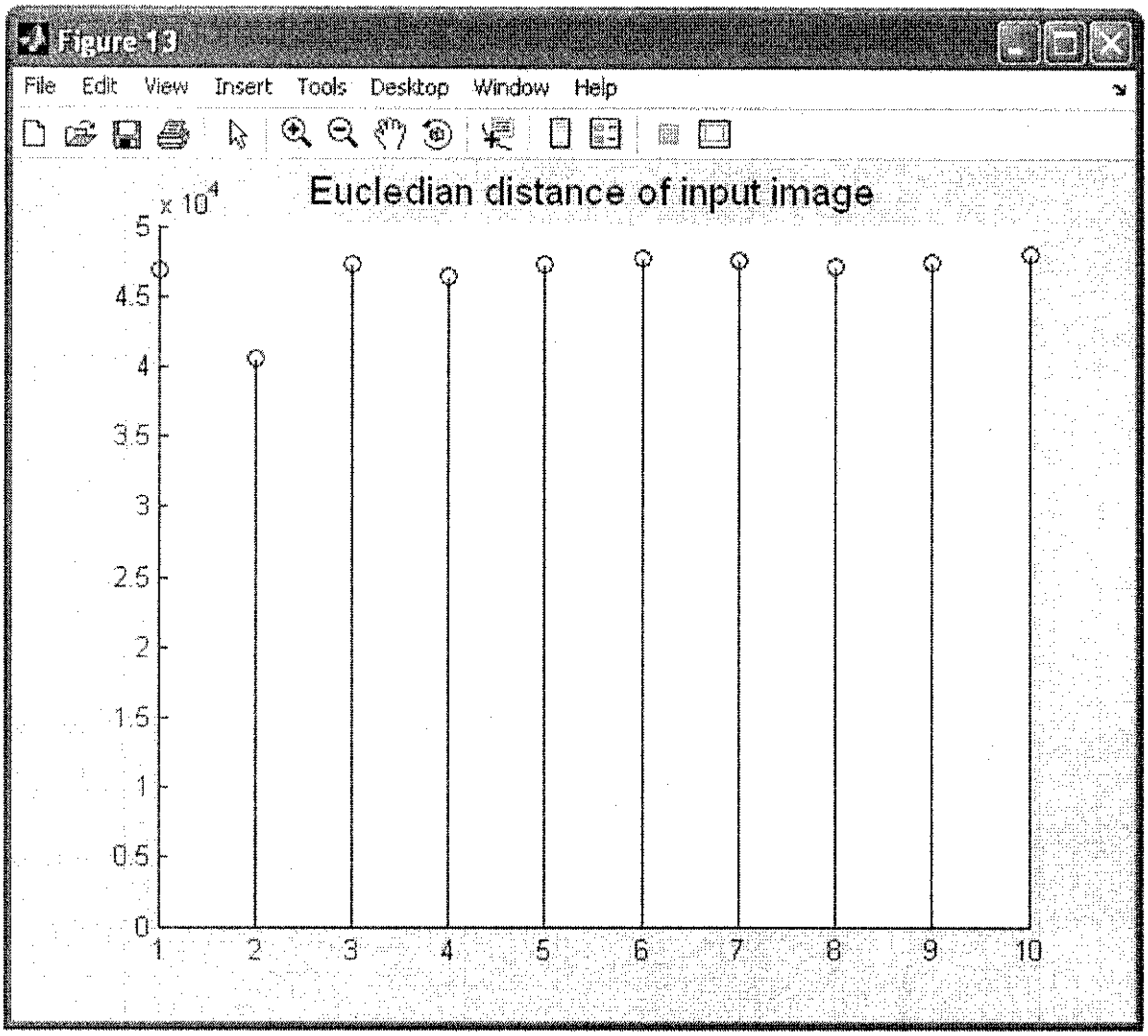


Figure 6.5 Euclidean distance of input image 2

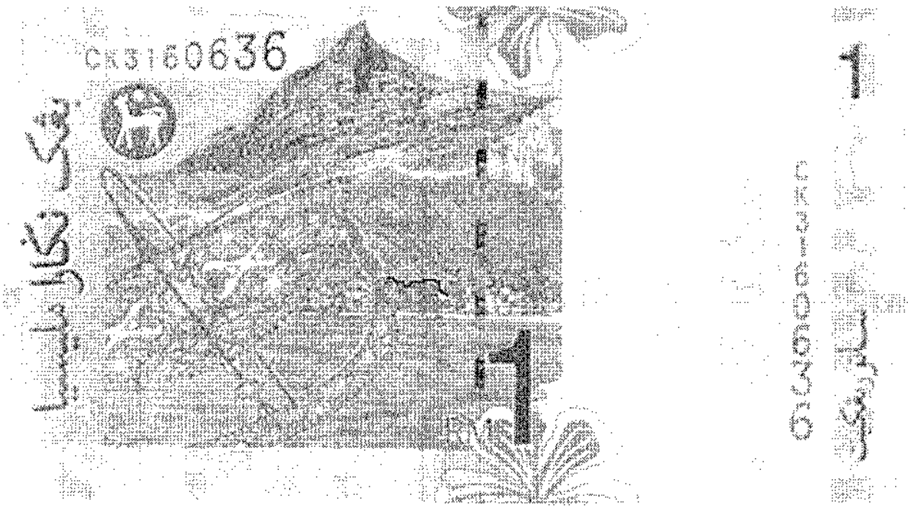


Figure 6.6 Slightly tampered counterfeit image

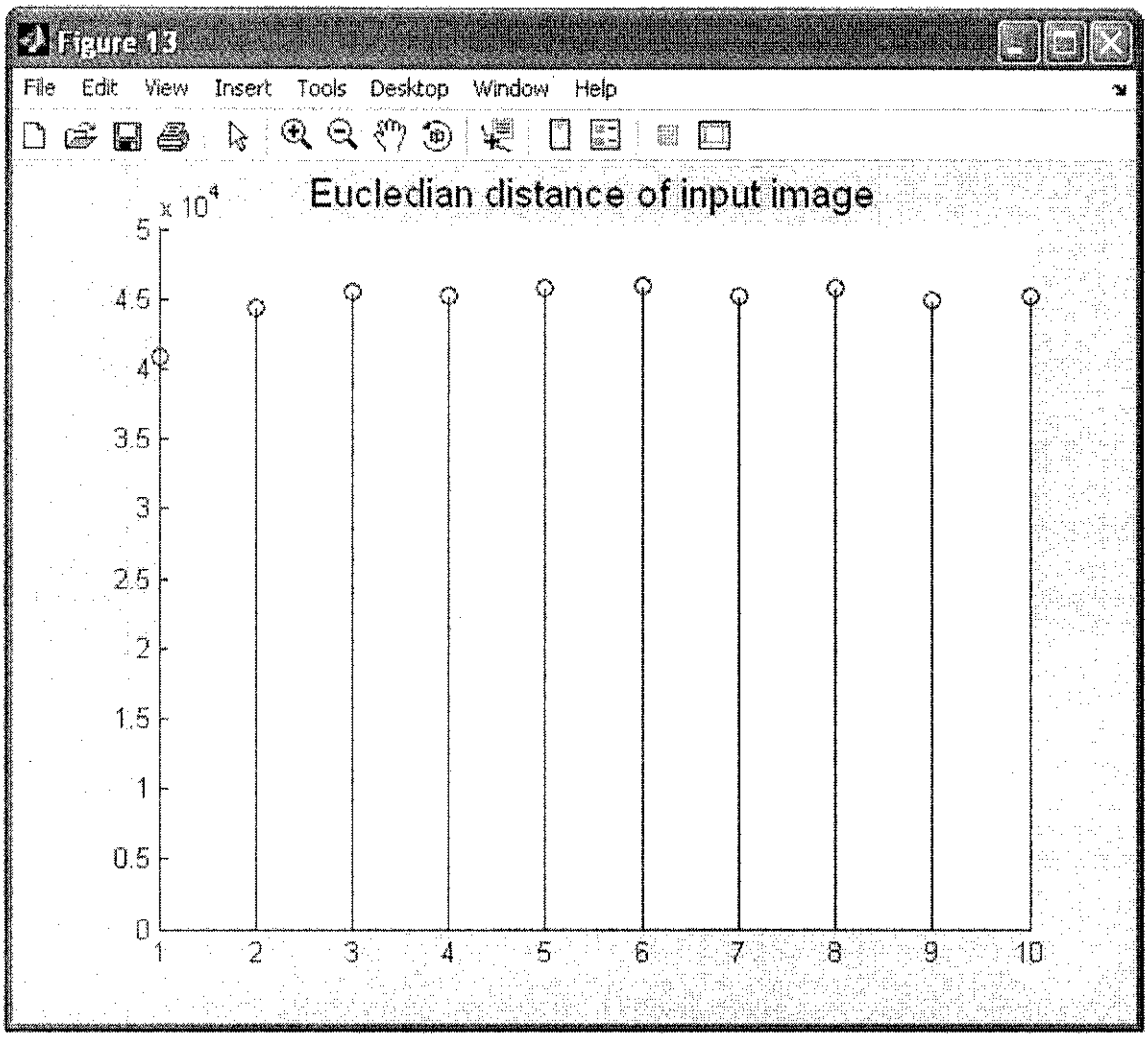


Figure 6.7 Euclidean distance of counterfeit currency image

Figure 6.8 Plane as an input image

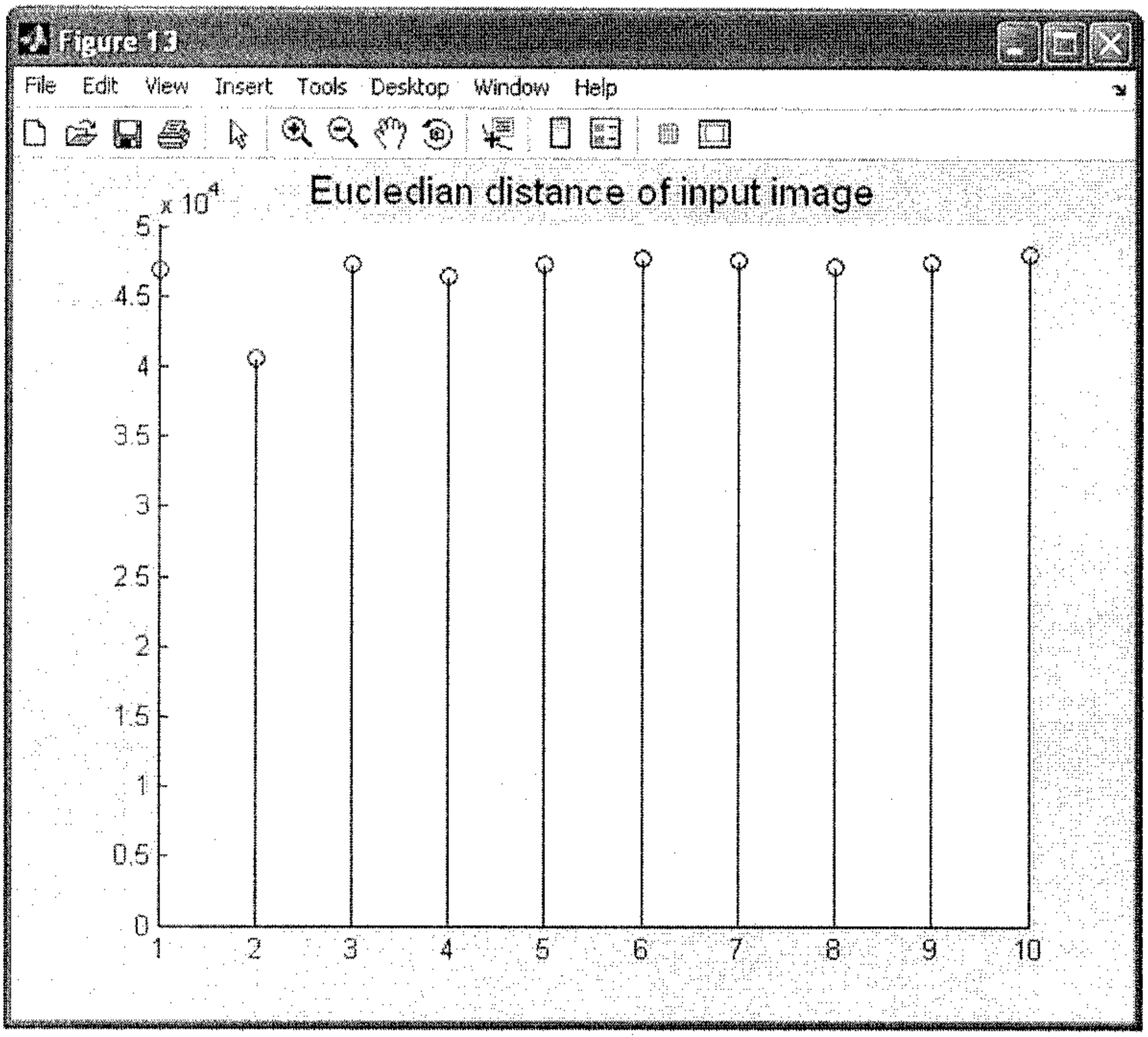
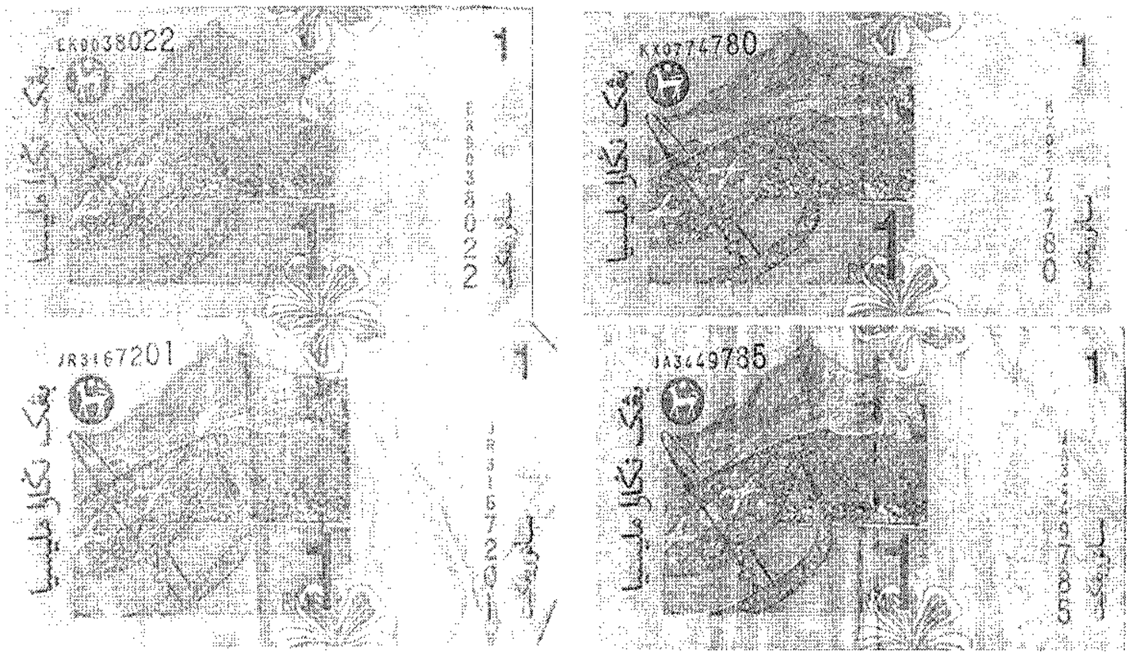
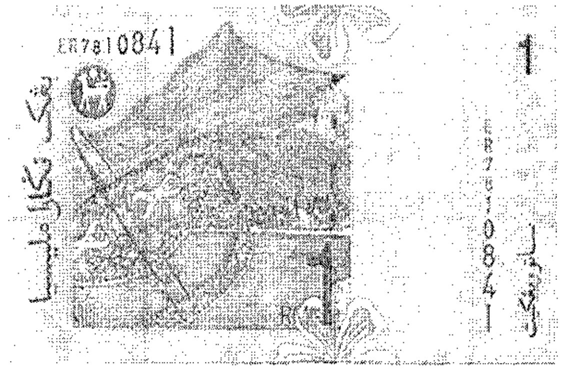
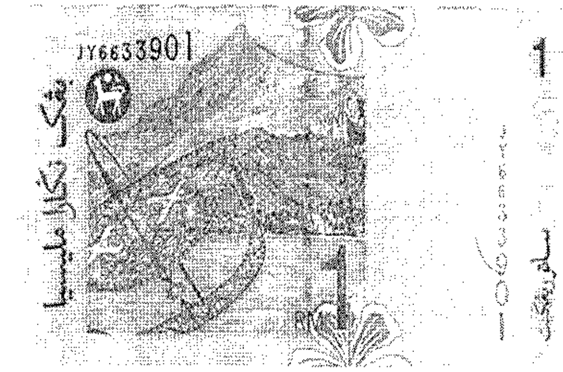


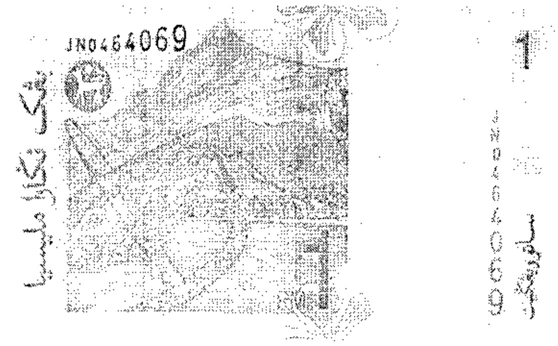
Figure 6.9 Euclidean distance of a plane image

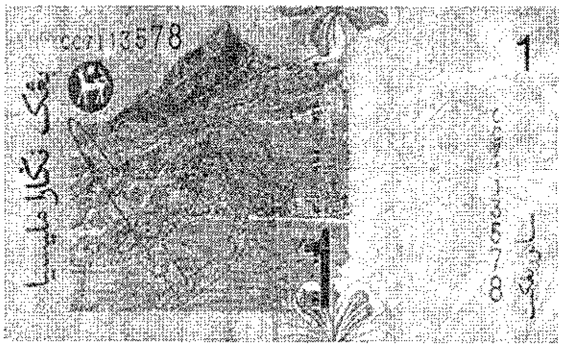
The rest of test images are shown below.

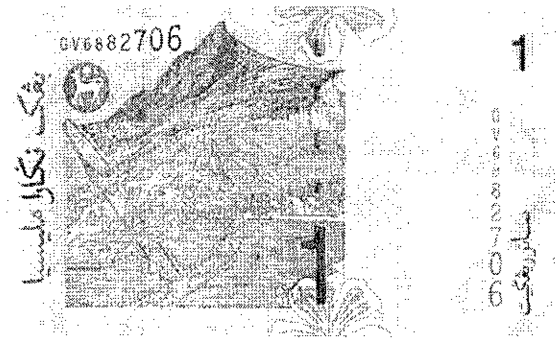


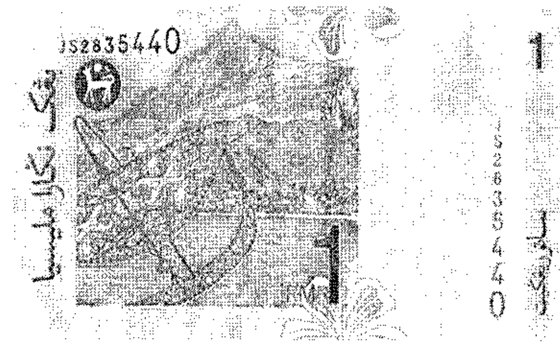


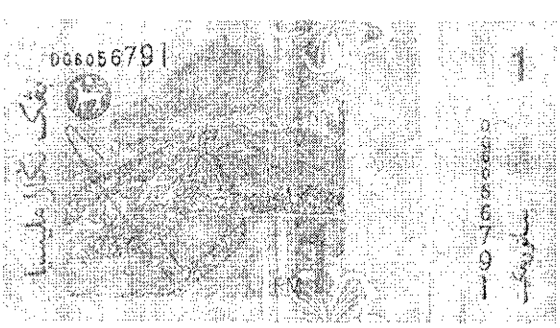












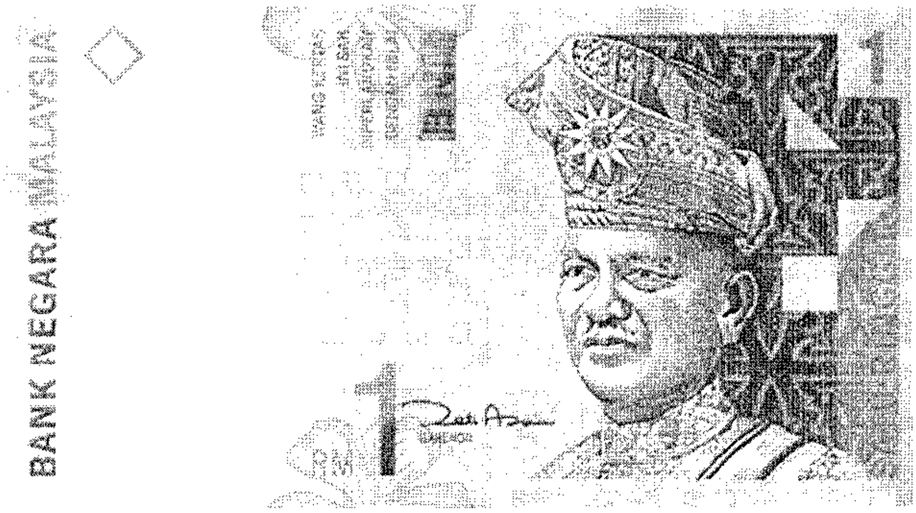


Figure 6.10 All test images

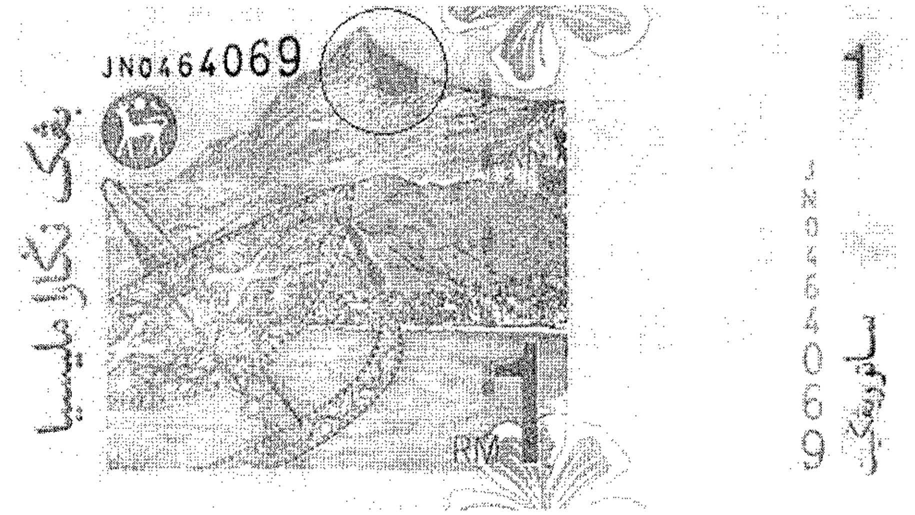


Figure 6.11 Counterfeit image 1

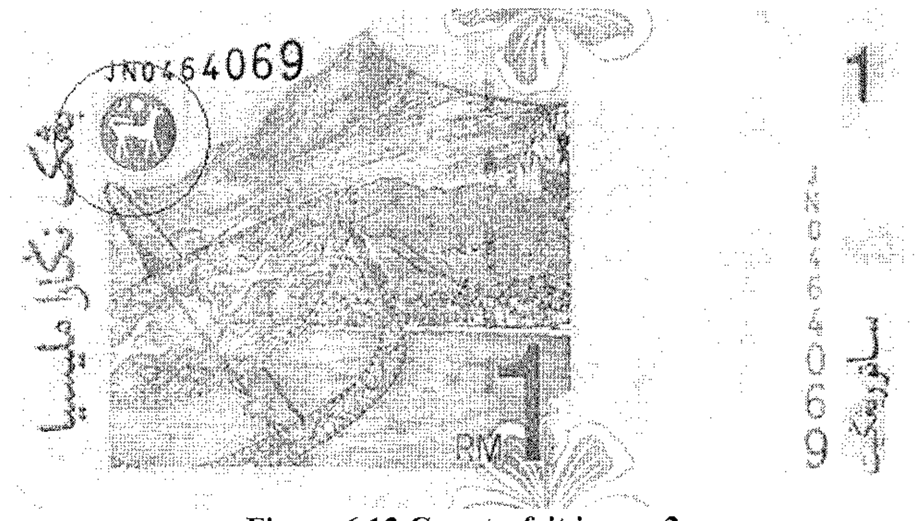


Figure 6.12 Counterfeit image 2

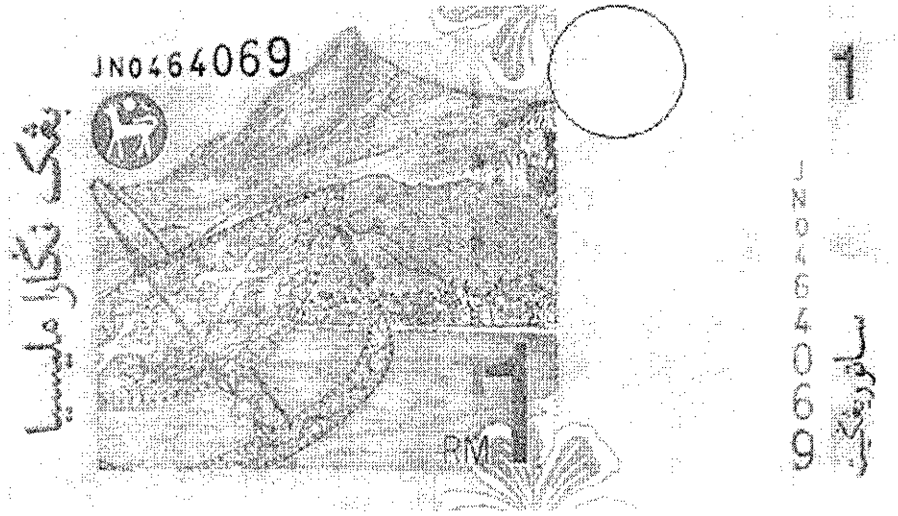


Figure 6.13 Counterfeit image 3

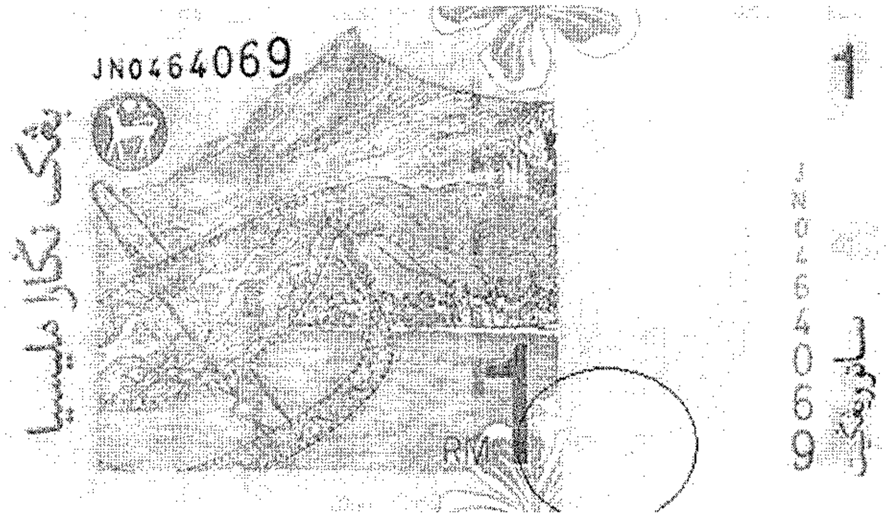


Figure 6.14 Counterfeit image 4

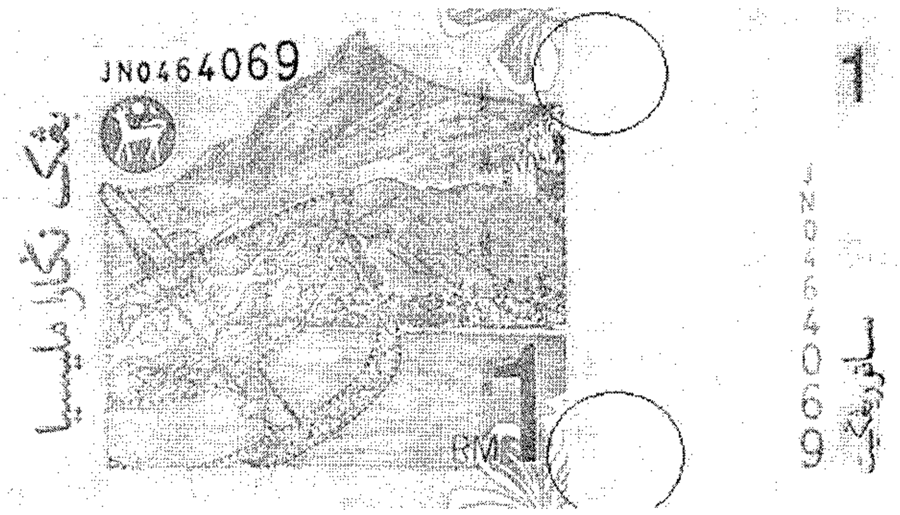


Figure 6.15 Counterfeit image 5

Euclidean Distances of Images (13 RM1 Images in Training Set)			Euclidean Distances of Images (10 RM1 Images in Training Set)		Average	
Image	Max Value	Min Value	Max Value	Min Value	Max Value	Min Value
1	45809	40794	46032	40867	45952	41246
2	48126	40620	48024	40634		
3	45973	40727	45578	40750		
4	46502	40877	46252	40891		
5	46789	40921	46567	40943		
6	46907	40840	46699	40861		
7	45088	40699	45271	40756		
8	47514	41119	47398	41149		
9	45854	40981	46228	41088		
10	45615	40602	45777	40663		
11	45360	40740	44008	41387		
12	46898	40979	44619	42977		
13	47554	40798	44923	43239		
counterfeit 1	45721	41073				
counterfeit 2	45619	41060				
counterfeit 3	45784	41139				
counterfeit 4	45606	41055				
counterfeit 5	45575	41001				
counterfeit 6	*45706*	*40821*	*45931*	*40895*		
plane	50853	48946	50647	49071		
frontside	60837	58293	60472	58156		
face	54005	51934	53925	51977		

Table 6.1 Euclidean distances for various input images including "counterfeit" image.

Appendix C – Malaysian Ringgit

Features are the basic elements for object recognition. Therefore, to identify a currency, we need to know what features are used effectively in the currency recognition process. Because the variance of each feature associated with the recognition process is relatively large, the features are classified into three major types as First-order features, Second-order features, and Higher-order feature values.

Since the first step of identification is to extract the features from currency images and also features being the basic elements for object recognition we list full features and details found on the Malaysian currency.

Figure 6.16 50 Malaysian Ringgit.

1. Watermark Portrait: The shaded watermark can be recognized by tints that are lighter or darker than the surrounding paper. This watermark portrait which has a three-dimensional effect appears soft and shady without sharp outlines. At the base of the watermark, the numeral 50 is clearly visible.

2. Security Fibres: When viewed under ultra-violet light, the security fibres in the paper become visible in three colors: red, yellow and blue.

3. Security Thread: The thread is embedded in the paper and appears on the reverse side of the note as a silver colored dotted line [a]. When the note is held against the light, it is seen as a continuous dark colored line and the repeated text BNM RM50 can be read [b]. When viewed under ultra-violet light, the thread is seen in various changing colors known as the "rainbow effect".

4. PEAK® (Printed and Embossed Anticopy Key): When changing the angle of view by shifting the note, the numeral 50 will be revealed in the centre of the PEAK® square. The whole square will glow under ultra-violet light. When held against the light, three open spaces on the obverse side will register perfectly with equal printed markings on the reverse side.

5. LEAD® (Long-lasting Economical Anticopy Device): Its holographic design represents the same motifs as used in the purple area with which the LEAD® strip is partially overprinted as well as the words "RM50" and "BNM". The colors of these elements change when the view angle is shifted. On both sides of the strip, a dedicated print pattern becomes visible under ultra-violet light.

6. Intaglio Print: The intaglio print is a raised printing effect produced by applying layers of tactile inks on various parts of the obverse and reverse sides of the notes, such as the portrait of the First Seri Paduka Baginda Yang di-Pertuan Agong, denomination figures, ornamental elements and the wordings "BANK NEGARA MALAYSIA".

7. Phosphorescence Square: When the note is held against the light, the hibiscus flower on the obverse will register perfectly with the same flower on the reverse of the note. This flower will also glow under ultra-violet light.

8. Perfect See-Through Register: When the note is held against the light, the hibiscus flower on the obverse will register perfectly with the same flower on the reverse of the note. This flower will also glow under ultra-violet light.

9. Invisible Fluorescent Elements: Various elements of the background on the obverse and reverse will fluoresce in different colors when viewed under ultra-violet light.

10. Novel Numbering: The serial numbers increase in size to make it more difficult to counterfeit. The numbers fluoresce under ultra-violet light.

11. Braille Feature: The diamond shape braille markings feature a layer of tactile ink printed in intaglio that can be felt by touching.

12. Anti-Scanner/Copier Features: The note features certain areas, designed such that these will change appearance when copied/scanned.

13. Modulated Micro-Letterings: In this tactile rectangle, the micro-letterings with the text RM50 are all legible under a magnifying glass and collectively form the word "BNM" if viewed from a distance.

14. Background Micro-Letterings: The pattern of the pink and bluish rectangles contains legible micro-letterings of "BNM" when viewed under a magnifying glass while some of the bluish rectangles will fluoresce under ultra-violet light.

15. Micro-Letterings: Circles around the oil valve hand wheel contain legible micro-letterings of the word "BANKNEGARAMALAYSIA" in light green when viewed under a magnifying glass.

RM1 Security Features

(Actual size 120 x 65 mm)

Watermark Portrait (1)

The shaded watermark can be recognised by tints that are lighter or darker than the surrounding paper. This watermark portrait which has a three-dimensional effect appears soft and shady without sharp outlines. At the base of the watermark, the numeral 1 is clearly visible.

Section 1 of the obverse of the note:

1

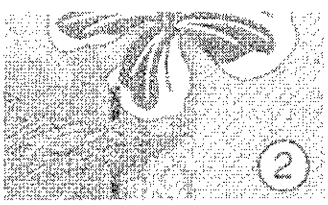
Paper Colour

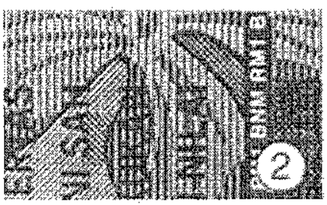
The paper is in a light shade of blue.

Security Thread (2)

The thread is embedded in the paper and appears on the reverse side of the note as a silver coloured dotted line. When the note is held against the light, it is seen as a continuous dark coloured line and the repeated text "BNM RM1" can be read. When viewed under ultra-violet light, the thread is seen in bluish colour.

Section 2 of the reverse of the note:

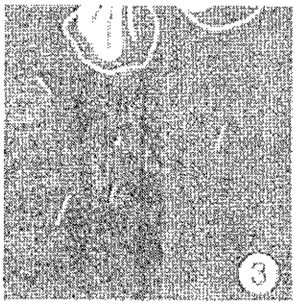




Security Fibre (3)

When viewed under ultra-violet light, the security fibres in the paper become visible in three colours: red, yellow and blue.

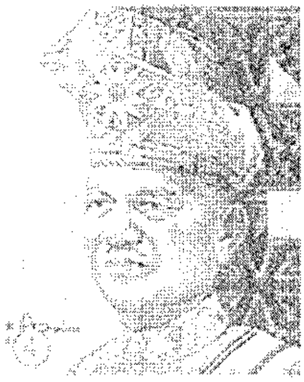
Section 3 of the obverse of the note:



Intaglio Print (4)

The intaglio print is a raised printing effect produced by applying layers of tactile inks on various parts of the obverse and reverse sides of the notes, such as the portrait of the First Seri Paduka Baginda Yang di-Pertuan Agong, denomination figures, ornamental elements and the wordings "BANK NEGARA MALAYSIA".

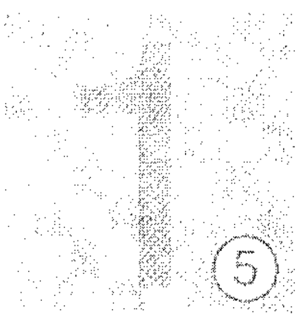
Section 4 of the obverse of the note:

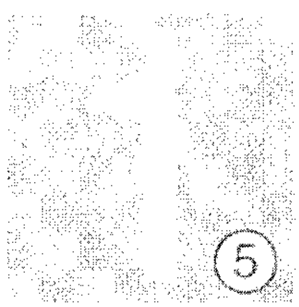


PEAK*Â*® (Printed and Embossed Anticopy Key) (5)

PEAKÂ® is a Printed and Embossed Anticopy Key security feature. When changing the angle of view by shifting the note, the numeral 1 will be revealed in the centre of the *PEAKÂ*® square. The whole square will glow under ultra-violet light.

Section 5 of the obverse of the note:



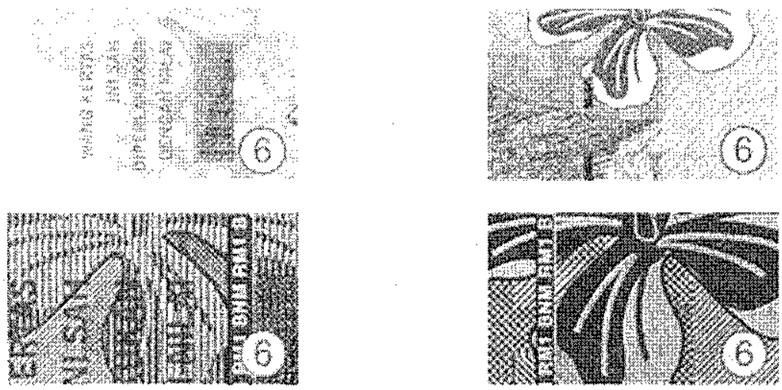


[a] Normal appearance [b] When held at an angle

Perfect See-Through Register (6)

When the note is held against the light, the hibiscus flower on the obverse will register perfectly with the same flower on the reverse of the note. This flower will also glow under ultra-violet light.

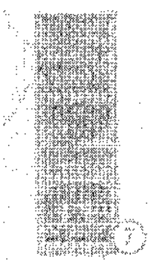
Section 6 of the note:



Modulated Micro-Letterings (7)

In this tactile rectangle, the micro-letterings with the text "RM1" are all legible under a magnifying glass and collectively form the word "BNM" if viewed from a distance.

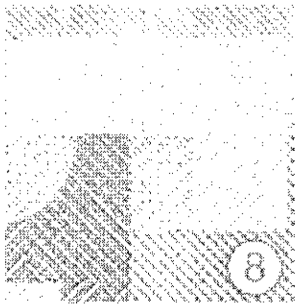
Section 7 of the obverse of the note:



Background Micro-Letterings (8)

The pattern of the bluish and purplish rectangles contains legible micro-letterings of "BNM" when viewed under a magnifying glass while some of the bluish and purplish rectangles will fluoresce under ultra-violet light.

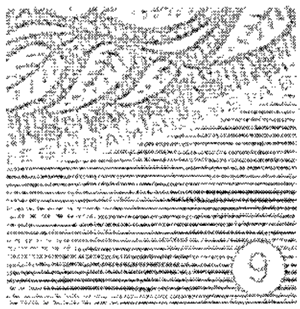
Section 8 of the obverse of the note:



Micro-Letterings (9)

The lower left part of the kite at the back of the note contains legible micro-letterings of the word "BANKNEGARAMALAYSIA" in blue when viewed under a magnifying glass.

Section 9 of the reverse of the note :

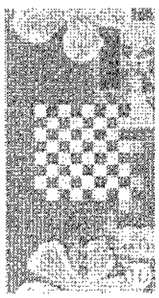


Invisible Fluorescent Elements (10)

Various elements of the background on the obverse and reverse will fluoresce in different colours when viewed under ultra-violet light.

Section 9 of the reverse of the note:

[a] Normal appearance

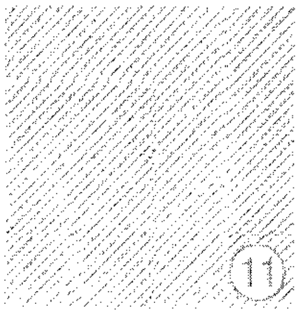


[b] When viewed under ultra-violet light

Anti-Scanner/Copier Features (11)

The note features certain areas, designed such that these will change appearance when copied/scanned.

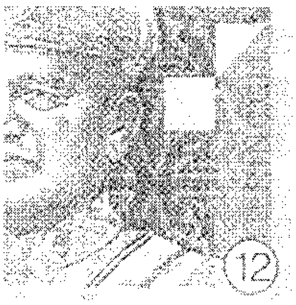
Section 11 of the obverse of the note :

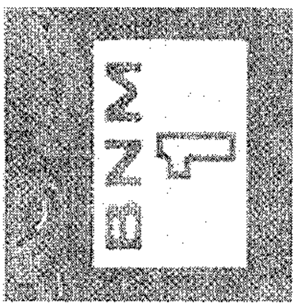


Phosphorescene Square (12)

In this square, the letters "BNM" and the numeral "1" will become visible under ultra-violet light.

Section 12 of the obverse of the note:

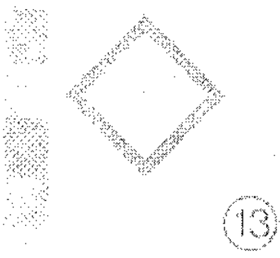




Braille Feature (13)

The diamond shape braille markings feature a layer of tactile ink printed in intaglio that can be felt by touching.

Section 13 of the obverse of the note:

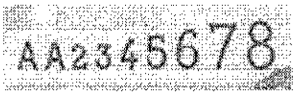


Novel Numbering (14)

The serial numbers increase in size to make it more difficult to counterfeit. The numbers fluoresce under ultra-violet light.

Section 14 of the reverse of the note:

[a] Normal appearance



[b] When viewed under ultra-violet light

Reference: http://www.bnm.gov.my/index.php?ch=23&pg=452&ac=26&security=1

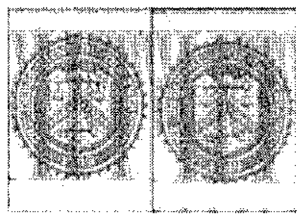
Appendix D - How To Detect Counterfeit Money (US dollar)

The public has a role in maintaining the integrity of currency. You can help guard against the threat from counterfeiters by becoming more familiar with United States currency.

Look at the money you receive. Compare a suspect note with a genuine note of the same denomination and series, paying attention to the quality of printing and paper characteristics. Look for differences, not similarities.

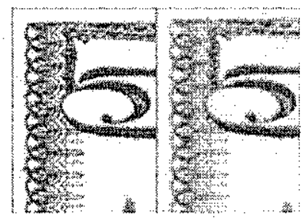
Portrait

The genuine portrait appears lifelike and stands out distinctly from the background. The counterfeit portrait is usually lifeless and flat. Details merge into the background which is often too dark or mottled.



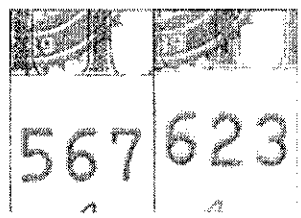
Federal Reserve and Treasury Seals

On a genuine bill, the saw-tooth points of the Federal Reserve and Treasury seals are clear, distinct, and sharp. The counterfeit seals may have uneven, blunt, or broken saw-tooth points.



Border

The fine lines in the border of a genuine bill are clear and unbroken. On the counterfeit, the lines in the outer margin and scrollwork may be blurred and indistinct.



Serial Numbers

Genuine serial numbers have a distinctive style and are evenly spaced. The serial numbers are printed in the same ink color as the Treasury Seal. On a counterfeit, the serial numbers may differ in color or shade of ink from the Treasury seal. The numbers may not be uniformly spaced or aligned.

Paper

Genuine currency paper has tiny red and blue fibers embedded throughout. Often counterfeiters try to simulate these fibers by printing tiny red and blue lines on their paper. Close inspection reveals, however, that on the counterfeit note the lines are printed on the surface, not embedded in the paper. It is illegal to reproduce the distinctive paper used in the manufacturing of United States currency.

Reference: http://www.secretservice.gov/money_detect.shtml

Appendix F - Suggested Milestones for Final Year Project

Suggested Milestone for the First Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
	-Propose Topic														
	-Topic assigned to students														
2	Preliminary Research Work														
	-Introduction														
	-Objective														
	-List of references/literature														
	-Project planning														
3	Submission of Preliminary Report			●											
4	Project Work														
	-Reference/Literature														
	-Practical/Laboratory Work														
5	Submission of Progress Report								●						
6	Project work continue														
	-Practical/Laboratory Work														
7	Submission of Interim Report Final Draft												●		

8	Oral Presentation													●	
9	Submission of Interim Report														●

● Suggested milestone

Process

Suggested Milestone for the Second Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project Work Continue														
	-Practical/Laboratory Work														
2	Submission of Progress Report 1			●											
3	Project Work Continue														
	-Practical/Laboratory Work														
4	Submission of Progress Report 2								●						
5	Project work continue														
	-Practical/Laboratory Work														
6	Submission of Dissertation Final Draft												●		
7	Oral Presentation													●	
8	Submission of Project Dissertation														●

● Suggested milestone
Process