CERTIFICATION OF APPROVAL

Biometrics: Facial Recognition

by

Wan Saarah Zuhri bt. Ahmad Zohri

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

Approved by,

(Mr. Patrick Sebastian)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2004

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 acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Wan Saarah Zuhri

WAN SAARAH ZUHRI BT. AHMAD ZOHRI

ABSTRACT

Biometrics, a biological measurement and refers to the automatic identification of a person based on his or her physiological or psychological characteristics. This project defines biometrics and its application in the real world focusing the study on facial recognition. Facial recognition is the identification of an individual based on the facial data characteristics such as facial features and face position. The objective of this project is to develop a program which could verify a face when compared to a database of known faces by using MATLAB. This project also explains the details of facial recognition especially on the four main facial recognition categories and other components related in characterizing a face. It also lists the various approaches in handling facial recognition where we see different methods applied and opinions on which method is better and what factors influenced them. Among the various approaches, eigenface technique is explained in detail including its work procedure, algorithm and the tools applied. Main part of this project discussed on the project findings. The result and output produced is elaborated according to the sequence of the program developed where many face images displayed. Finally, this project reviews the relevancy of the study contents with the objectives and listed out some recommendation for further work expansion.

ACKNOWLEDGEMENT

First of all, I am grateful to Allah, The Most Beneficent, The Most Merciful, for giving me strength and good health to carry out this project.

I would like to convey my highest and deepest gratitude to my supervisor, Mr. Patrick Sebastian who had assisted me in doing this project by giving useful guidance and support throughout these two semesters. Thank you for the suggestions and developing criticism which made this final year project a success.

I would like to thank two other researchers, Mr. Santiago Serrano and Mr. Matthew Dailey, for their tutorials and guidance which really helped me in completing this project, although we are only verbally connected from afar. The cooperation from both of you is really appreciated.

Besides that, I would also like to thank Mr. Jale Ahmad, an IT lecturer for his willingness to share his knowledge, discussing on the MATLAB program. The time spent on it was really precious.

My appreciation also goes to Mr. Wan Muhammad Nubli who had sat together and contributed some ideas throughout the project which is much a handful help for me.

A token of appreciation goes to each UTP EE lecturers that are so helping to guide and advices in any matters in pertaining to this project. All your guidance is highly appreciated and will be remembered.

Not to forget, all my beloved friends and family who are always there supporting me in any matters in order for me to commit works and able to motivate and inspire me to achieve the best for this project.

As it is impossible to list each person involved in this final year project, I would like to thank everyone contributed in this project, either directly or indirectly. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

S	Training set
Γ_i	Face image i of the training set
M	Number of face images
Ψ	Average image
C	Covariance matrix
u	Eigenvectors
λ	Eigenvalue
ω_i	Weight i
$\Omega^T{}_i$	Weight vector of the image i

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

1.1.1 Biometrics

A concise definition of biometrics is the automatic recognition of a person using distinguishing traits [11]. A more expansive definition of biometrics is "any automatically measurable, robust and distinctive physical characteristic or personal trait that can be used to identify an individual".

Measurable means that the characteristic or trait can be easily presented to a sensor, located by it, and converted into a quantifiable, digital format. This measurability allows for matching to occur in a matter of seconds and makes it an automated process.

The *robustness* of a biometric refers to the extent to which the characteristic or trait is subject to significant changes over time. These changes can occur as a result of age, injury, illness, occupational use, or chemical exposure. A highly robust biometric does not change significantly over time while a less robust biometric does not change significantly over time while a less robust biometric will change. For example, the iris, which changes very little over a person's lifetime, is more robust than one's voice.

Distinctiveness is a measure of the variations or differences in the biometric pattern among the general population. The higher the degree of distinctiveness, the more individual is the identifier. A low degree of distinctiveness indicates a biometric pattern found frequently in the general population. The iris and retina have higher degrees of distinctiveness than hand or finger geometry.

1.1.2 Facial Recognition

The face is our primary focus of attention in social intercourse, playing a major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable. We can recognize thousands of face learned throughout our lifetime and identity familiar faces at a glance even after years of separation. The technology of a facial recognition system is used to verify an identity of a person by matching a given face against a database of known faces. It has become a viable and an important alternative to traditional identification and authentication methods such as the use of keys, ID cards and passwords [1].

Facial recognition systems are built on computer programs that analyze images of human faces for the purpose of identifying them. The programs take a facial image, measure characteristics such as the distance between the eyes, the length of the nose, and the angle of the jaw, and create a unique file called "template". Using templates, the software then compares that image with another image and produces a score that measures how similar the images are to each other. Typical sources of images for use in facial recognition include video camera signals and pre-existing photos such as those in driver's license databases. Although the concept of recognizing someone from facial features is intuitive, facial recognition, as a biometric, makes human recognition a more automated, computerized process.

As a biometric, facial recognition is a form of computer vision that uses faces to attempt to identify a person or verify a person's claimed identity. Just as with fingers scan and video scan biometrics, there are various methods by which facial scan technology recognizes people. All share certain commonalities, such as emphasizing those sections of the face which are less susceptible to alteration, including the upper outlines of the eye sockets, the areas surrounding one's cheekbones, and the sides of the mouth. Most technologies are resistant to moderate changes in hairstyle, as they do not utilize areas of the face located near the hairline. All of the primary technologies are designed to be robust enough to conduct a one-to-many search, which is to locate a single face out of a database of hundreds, even thousands, of faces.

1.1.3 Examples of Existing Application using Facial Recognition System

- **ID-2000 SDK™**

  This product can be integrated in the existing application, or develop ones own third party solution based on the advanced technology. ID-2000 SDK includes a number of ActiveX controls enabling users to integrate the technology quickly within any COM-compliant program.

- **FaceIt**

  Using this product, it can pick someone's face out of a crowd, extract that face from the rest of the scene and compare it to a database full of stored images. In order for this software to work, it has to know what a basic face looks like. The facial recognition software is based on the ability to first recognize the face, which is a technological feat in itself, and then measure the various features of each face.

- **RecognitionMATE™**

  The RecognitionMATE is an automatic process for comparing captured face images from video feeds to Face Characteristic Data (FCD) generated from pre-enrolled images stored in a database. It declares a match when two such images are sufficiently similar. Video face recognition is implemented using the functional units of a face scanner, a face matching engine and a face search engine.

- **ID-2000™**

  The product is used as a complete face and image recognition solution for use in airports, driver's license and passport modernization programs, financial services, homeland defense, and more.

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

These days, the number of criminals and terrorist increased rapidly. Robbery, murder, bombing and other harassment are the main stories in the newspaper each day including our country. Many investigations and operations have been done by the government especially the police department in order to identify and capture the criminals including having the assistance of people to give information or identify them.

1.2.2 Significance of the Project

This facial recognition project is conducted as an initiative in helping the government to enhance the quality of safety and surveillance in this country. The project focused in providing a face identification system to assist the investigation and identification process. With the existence of this facial recognition system, it would help the government and other responsible authorities to verify the face of criminal by comparing the suspected face to the database of faces provided. Other government and private sectors such as administration offices and banks would also implement the stand alone system as one of the security element in protecting the restricted information and area.

1.3 OBJECTIVES AND SCOPE OF STUDY

1.3.1 Objectives of Study

The main objective of the study is to develop a program which could verify a face by comparing or matching the face to a database of known faces by using MATLAB. In achieving the focal objective, a few subordinate objectives has been set to find crucial information on any related issues which can contribute to the success production of a facial recognition MATLAB program by the end of this practice. The subordinate objectives are:

i. to study the facial recognition categories
ii. to research different techniques in handling facial recognition
iii. to provide a face database
iv. to develop the face recognition algorithm

1.3.2 Scope of Study

The study covers the overview of biometrics and facial recognition stages and followed by the research on different techniques in handling facial recognition. Apart from all the techniques researched, only one technique is applied by understanding the requirements and functionalities of the technique in order to develop the program phase by phase. A database of known faces is provided as the main element to test the functionality of the algorithms developed. In this study, the input image and database is mainly consisting frontal view known faces. The expectation is the program to have the ability in producing and displaying the whole process of facial recognition in phase based on the technique chosen.

CHAPTER 2

LITERATURE REVIEW

2.1 BIOMETRICS

In the old days, human used body characteristics such as face and voice to recognize each other. As time goes by, human became more developed and involved in many activities that need security such when we withdraw money, to use restricted equipments, to enter restricted area and many else. Nowadays, biometrics have an important role among other verification and identification system regarding safety and security of the people. It is a rapidly evolving technology that has been widely used in forensics such as criminal identification and prison security. Biometrics is unique as it based on the measurements on the biological measurements. Biometrics techniques can be divided into two categories: physiological and psychological. Physiological biometrics is based only on the physiologic aspects of the body, and the most common are face, fingerprint, hand geometry, iris and retina while psychological biometrics is based mainly on psychological aspects, where the most common are handwriting and voice [2]. Any human physiological or behavioral characteristic can be used as a biometric characteristic as long as it satisfies the following requirements:

- **Universality**: each person should have the characteristic
- **Distinctiveness**: any two persons should be sufficiently different in terms of the characteristic
- **Permanence**: the characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time
- **Collectability**: the characteristic can be measured quantitatively

A practical biometric system should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, and be sufficiently robust to various fraudulent methods and attacks to the system.

A *biometric system* is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. Depending on the application context, a biometric system may operate either in *verification* mode or *identification* mode [7]:

i) Verification Mode

In the verification mode, the system validates a person's identity by comparing the captured biometric data with the person's own biometric template(s) in a stored system database. In such a system, an individual who desires to be recognized claims an identity, usually via a PIN (Personal Identification Number), a user name, a smart card, etc., and the system conducts a one-to one comparison to determine whether the claim is true or not (e.g., "*Does this biometric data belong to Bob?*"). Identity verification is typically used for *positive recognition,* where the aim is to prevent multiple people from using the same identity.

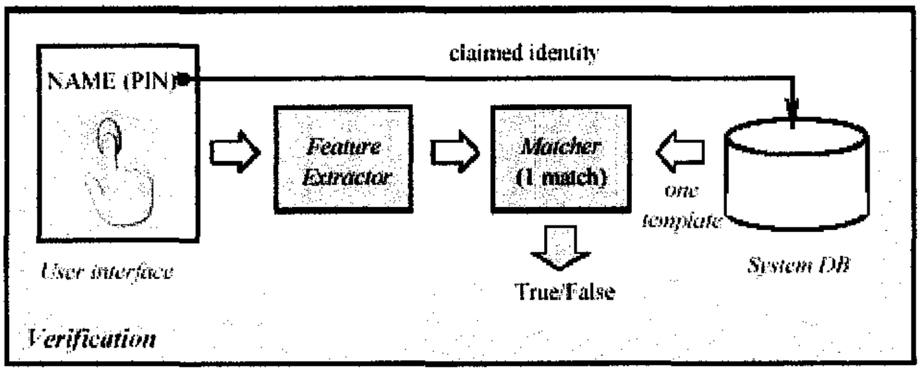


Figure 2.1: Example of Verification Mode

ii) Identification Mode

In the identification mode, the system recognizes an individual by searching the templates of all the users in the database for a match. Therefore, the system conducts a one-to-many comparison to establish an individual's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity (e.g., "*Whose biometric data is this?*"). Identification is a critical component in *negative recognition* applications where the system establishes whether the person is who she (implicitly or explicitly) denies to be. The purpose of negative recognition is to prevent a single person from using multiple identities. Identification may also be used in positive recognition for convenience (the user is not required to claim an identity). While traditional methods of personal recognition such as passwords, PINs, keys, and tokens may work for positive recognition, negative recognition can only be established through biometrics.

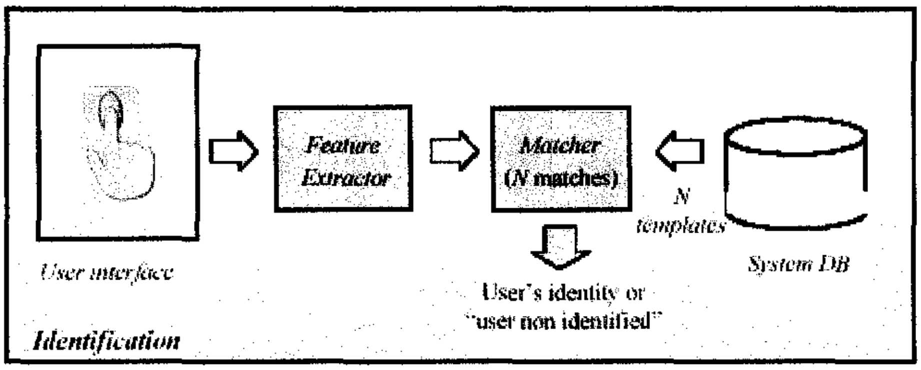


Figure 2.2: Example of Identification Mode

A number of biometric characteristics exist and are in use in various applications. Each biometric has its strengths and weaknesses, and the choice depends on the application. No single biometric is expected to effectively meet the requirements of all the applications. In other words, no biometric is "optimal". The match between a specific biometric and an application is determined depending upon the operational mode of the application and the properties of the biometric characteristic. A brief introduction of the commonly used biometrics is given below:

- **Fingerprints**: Humans have used fingerprints for personal identification for many centuries and the matching accuracy using fingerprints has been shown to be very high. A fingerprint is the pattern of ridges and valleys on the surface of a fingertip, the formation of which is determined during the first seven months of fetal development. Fingerprints of identical twins are different and so are the prints on each finger of the same person. The accuracy of the currently available fingerprint recognition systems is adequate for verification systems and small- to medium-scale identification systems involving a few hundred users.

- **DNA:** Deoxyribo Nucleic Acid (DNA) is the one-dimensional ultimate unique code for one's individuality - except for the fact that identical twins have identical DNA patterns. It is, however, currently used mostly in the context of forensic applications for person recognition.

- **Face:** Face recognition is a non-intrusive method, and facial images are probably the most common biometric characteristic used by humans to make a personal recognition. The applications of facial recognition range from a static, controlled "mug-shot" verification to a dynamic, uncontrolled face identification in a cluttered background (e.g., airport). The most popular approaches to face recognition are based on either the location and shape of facial attributes, such as the eyes, eyebrows, nose, lips, and chin and their spatial relationships, or the overall (global) analysis of the face image that represents a face as a weighted combination of a number of canonical faces.

- **Ear**: It has been suggested that the shape of the ear and the structure of the cartilegenous tissue of the pinna are distinctive. The ear recognition approaches are based on matching the distance of salient points on the pinna from a landmark location on the ear. The features of an ear are not expected to be very distinctive in establishing the identity of an individual.

2.2 FACIAL 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. A human being is a very sophisticated information system, partly because he possesses a superior pattern recognition capability [8]. Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, nose, mouth, and head outline and defining a face model by the position, size and relationships among these features. Such approaches have proven difficult to extend to multiple views, and have often been quite fragile, requiring a good initial guess to guide them. Research in human strategies of face recognition, moreover has shown that individual features and their immediate relationships comprise an insufficient representation to account for the performance of adult human face identification. Nonetheless, this approach of face recognition remains the most popular one in the computer vision literature [4].

Face recognition is a pattern recognition task performed specifically on faces. It can be described as classifying a face either "known" or "unknown", after comparing it with stored known individuals. It is also desirable to have a system that has the ability of learning to recognize unknown faces. Computational models of face recognition must address several difficult problems. This difficulty arises from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces. Faces pose a particularly difficult problem in this respect because all faces are similar to one another in that they contain the same set of features such as eyes, nose, mouth arranged in roughly the same manner. Facial recognition itself is a wide topic to be discussed. From the study, many researchers and developers have their own opinions and approaches in explaining this topic however they are all almost similar and related to each other. According Ilker Atalay, [8] there is a typical outline for facial recognition system. The outline is shown in the figure below.

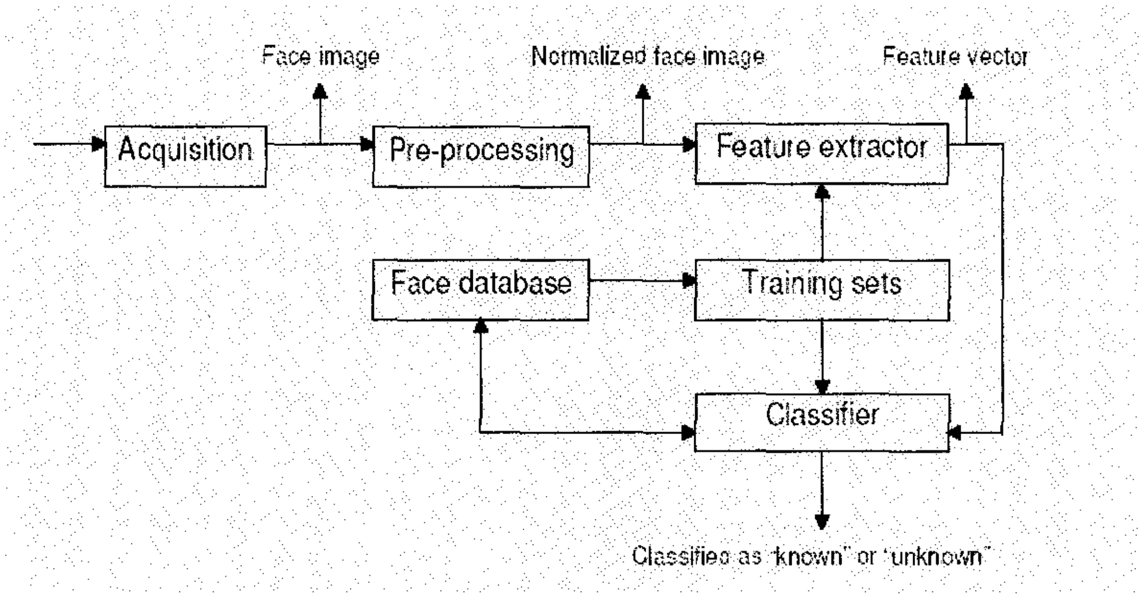


Figure 2.3: Outline of a Typical Facial Recognition System

From the figure above, the typical facial recognition system consist of six main blocks:

i. **The acquisition module:** This is the entry point of the face recognition process. It is the module where the face image under consideration is presented to the system. In other words, the user is asked to present a face image to the face recognition system in this module. The face image can be an image file that is located on a magnetic disk, it can be captured by a frame grabber or it can be scanned from paper with the help of a scanner.

ii. **The pre-processing module:** In this module, by means of early vision techniques, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system.

iii. **The feature extraction module:** After performing some pre-processing (if necessary), the normalized face image is presented to the feature extraction module in order to find the key features that are going to be used for classification. In other words, this module is responsible for composing a feature vector that is well enough to represent the face image.

iv. **The classification module:** In this module, with the help of a pattern classifier, extracted features of the face image is compared with the ones stored in a face library (or face database). After doing this comparison, face image is classified as either known or unknown.

v. **Training set:** Training sets are used during the "learning phase" of the face recognition process. The feature extraction and the classification modules adjust their parameters in order to achieve optimum recognition performance by making use of training sets.

vi. **Face library or face database:** After being classified as "unknown", face images can be added to a library (or to a database) with their feature vectors for later comparisons. The classification module makes direct use of the face library.

Another group of researchers, K. Kalairasi, Aliza Tajudin and R.K Subramaniam from University of Science Malaysia (USM) stated that modern research on facial recognition can be divided into few categories: **face detection, facial features extraction, face normalization and face identification** [1].

Face detection is often considered the pre-processing step, where it is executed in an entrance level. Although it is an early step, it is not necessarily simple. Face detection locate a face in complex scenery, by locating and cutting it out [2]. Different specific approaches were proposed for the face detection. Among those are graph matching, model-based approach, motion-based approach, elastic matching and probability densities.

The second category is **facial features extraction** where it normally used either for normalization step or identification process. Among the techniques used for facial extraction are eigenfeatures and symmetry operators. This category involves locating the facial features of interest such s the eyes, nose and mouth.

The third category, **Face normalization** is a crucial phase as the robustness behaviour of a face recognition system greatly depends on it. By performing explicit normalization on the input image, robustness of the face recognizers is increased. A traditional way of performing lighting normalization is histogram equalization. Having the facial features location, scaling and orientation can be normalized using template-matching approach.

The final category in facial recognition is **face identification**. This category is the actual recognition process where the input face is compared with the database of known faces. In the paper of Face Recognition Based on Symmetryzation and Eigenfaces [2], the author adds another category for facial recognition, which is **Face Representation**. Face Representation consists in face modeling, in its codification. Any record stored into a database can be represented easily by its primary key, but face image representation isn't so easy, demanding complex algorithms to facilitate a good representation. The face representation way determines the successive algorithms of detection and identification. For the entrance level recognition, a category of faces should be characterized by generic properties of all faces; and for the subordinate level recognition, for detailed characteristics of eyes, nose and mouth of each face. The similarities of the two categories are both using some similar approaches such as template-based and feature-based method in applying these categories. A few methods can be used under this category:

i. Feature-based method

ii. Template-based method

iii. Appearance-based method

Feature-based method is a geometrical approach and needs the facial features location to be known in advance. It considers the positions and sizes of the facial organs, as eyes, nose, mouth, etc. Having the location, spatial configuration of facial features are captured and used to form feature vectors. In other words for each image, all the distances are calculated among the organs of the face. The extracted characteristics for vertical gradients are useful for the detection of the top of the head, eyes, nose base and mouth. The horizontal gradients are useful for detection of the lateral edges of the face and nose. The goal is to acquire a correspondence type "one-to-one" among the characteristics of the questioned face and the characteristics of the faces stored in a database. Comparison is then made with the input image's feature vector with the database of the image's vector. This method requires only minimal memory and higher speed. However, it relies heavily on feature location algorithm. It is also unable to perform well under varying imaging condition.

For the researchers from USM, the **template-based method** is a pictorial approach. The image vector is formed by ordering the gray-scale image of the unknown face. The face image can be represented by its original gray levels, gradient magnitude or gradient vectors. The image vector is then compared to the database by calculating correlation. The image with the largest correlation is chosen to be the closest match. Another researcher divides the template – based method into two versions. The first and simpler version intends to represent the faces by means of a main two-dimensional template with values representing the facial ellipse borders and of all face organs. The second version, more completed, it has multiple templates for the face representation, under several angles and points of view. Another important approach is the use a group of smaller facial characteristics models, corresponding to the eyes, nose and mouth, for an only point of view. Although this method provides easy implementation, it is highly sensitive to scaling, noise and lighting. Besides, it needs a great amount of memory and has n inefficient method of comparison.

The **appearance-based method** intends to project the face images in a lineal low dimension subspace, to obtain the face representation. The eigenfaces space is an application of this method. It is built on PCA–Principal Component Analysis, from the projection of the images of the training set into the face space (with low dimension).

There are more techniques used in the **face detection** category which also the different techniques applied in handling facial recognition. Other than the above methods, examples of facial recognition techniques are neural network, elastic graph approach and eigenface technique. These examples are explained briefly.

The **neural network** has been a popular technique in facial recognition and there are many variations. The algorithm works by applying one or more neural networks directly to portions of the input image, and arbitrating their results. Each network is trained to output the presence or absence of a face. The algorithms and training methods are designed to be general, with little customization for faces. The most popular is the back-propagation neural network. The disadvantages of this approach are its sensitivity to lighting variation, difficulties in implementation, complex and arduous training. Even a small size image needs a large number of neurons input for processing. The fundamental problem of neural network is due to its method not explicitly using the configuration properties of faces.

Face is represented as elastic labelled graph of local textural features in the **elastic graph approach.** Nodes are particular points of face such as eyes, nose and mouth. This method is suitable for faces with different views. New faces are matched to find the facial landmark and then used for comparison. The recognition rate by this method is high but it is computationally expensive.

The most prominent technique is the **eigenfaces** approach. The scaling or normalization of facial features according to their relative importance in face recognition is the basic premise behind the eigenfaces technique. This technique does not depend upon having full three-dimensional models or detailed geometry. Unlike other techniques, eigenfaces are robust against noise, poor lighting conditions and partial occlusion. Eigenfaces are relatively insensitive to small variations in scale, rotation and expression. Eigenfaces have been shown to produce 96% accurate classification under varying lighting condition. Coupled with their robustness against variations in scale and rotation, it is clearly superior over techniques based on distance between facial features. Eigenfaces requires no training and so it is not subjected to the problems associated with neural networks. Eigenfaces have also been shown to maintain accuracy even with large-scale database.

Another common technique in facial recognition is using **Principal Component Analysis (PCA)**. This method is somehow interrelated with eigenface. PCA is one of the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the broad title of factor analysis [5]. The objective of the PCA is to take the total variation on the training set of faces and to represent this variation with just some little variables. An observation described by little variables is easier to manipulate and to understand than described by a big quantity of variables [2]. In the other word, the purpose of PCA is to reduce the large dimensionally of the data space (observed variables) to the smaller intrinsic dimensionally of feature space (independent variables), which are needed to describe the data economically. The jobs, which PCA can do, are prediction, redundancy removal, feature extraction, data compression and etc. The main idea of using PCA for facial recognition is to express the large 1-D vector of pixels constructed from 2-D facial image to compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors). This approach proposes to extract the significant information from the face images, to code them in the most possible efficient way, and to compare these coefficients with a database with well-known faces. This codification is performed by means of capturing variations of the whole faces group used for tests, independently of the individual face characteristics. It is used later in the comparison of the analyzed faces.

2.3 EIGENFACE METHOD

What is eigenface? Eigenface recognition derives its name from the German prefix "eigen", meaning "own" or "individual"[6]. The eigenface recognition approach was developed by Turk and Pentland, both colleagues from MIT, in 1987. The eigenface method of facial recognition is considered the first working facial recognition technology [3]. The eigenface approach was first proposed by Sirovich and Kirby as an application of the principal component analysis for representation of face images. From considering face portion containing eyes and nose only, they extended their approach to full face. Turk and Pentland later refined this method by adding preprocessing and expanding database. They have also discovered that only a relatively small number of eigenfaces are required for the recognition process. This was a remarkable finding because it significantly reduced the computational time [1].

Generally, the eigenface recognition system begins by collecting a large number of facial images in a database. The system then creates a set of eigenfaces by combining all of the facial images in the database and comparing commonalities and differences between groups of individual facial images. The eigenfaces developed by the system appear as two-dimensional sets of light and dark areas arranged in a particular pattern. When a face is presented to the eigenface system for identification, the location of the eyes is found first. The eye location provides a reference point so the head can be located and scaled to a standard size. Next, the system concentrates on the face only, leaving out clothing and hair, and removing brightness and contrast variations caused by the cameras settings. Then the program compares the live face's eigenface characteristics with those in the database and determines a "degree of fit" score, between -1.0 and +1.0, for the target face. If the target face produces a high enough degree of fit score when compared to a face stored in the database, it is recognized and accepted by the system. Practically any individual can be identified using a database of 100 to 150 eigenfaces.

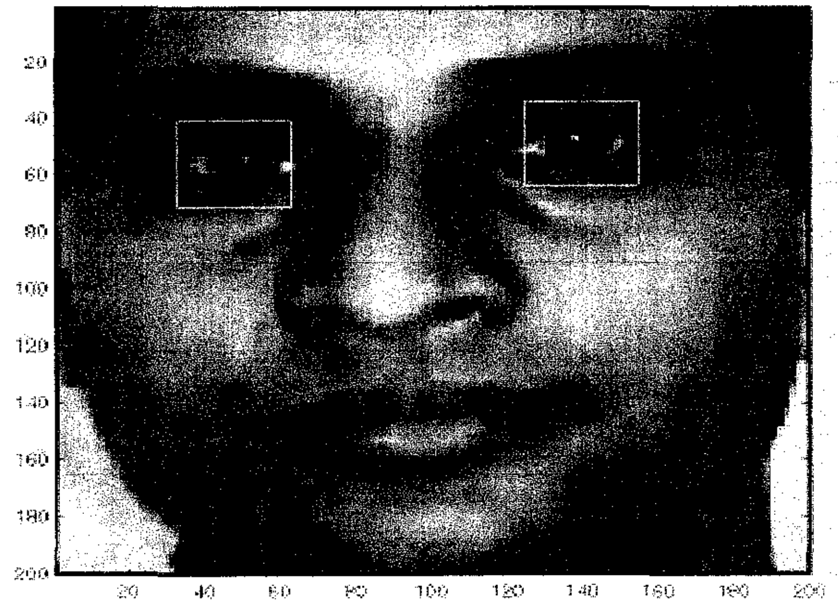


Figure 2.4: The location of eyes is found first in the eigenface system for identification

In facial recognition, the aspects of the face stimulus are important for identification. It suggested an information theory of coding and decoding face images may give an insight into the information content of face images, emphasizing the significant local and global "features". Such features may or may not directly relate to our intuitive notion of face features such as the eyes, nose, lips, and hair. This may have important implications for the use of identification tools. In the language of information theory, the relevant information in a face images is extracted, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extract the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare the individual face images [4]. The eigenface representation method for face recognition is based on the principal component analysis. 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. Eigenfaces are like ghostly faces formed by the eigenvectors. These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Under the assumption that faces form a simply connected region in image-space (the space contains all possible images, facial and non-facial), we can represent any face as a linear combination of eigenvectors. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face.

Figure 2.5: Examples of images collected in the database

Figure 2.6: Images after reconstructed into eigenfaces

From Dimitri Pissarenko [6], eigenfaces are nothing less than characteristic features of the faces. Therefore one could say that the original face 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 face, 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 corresponding 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 all) part of 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. By comparing the weight vector of an unknown face to the weight vectors of a database of known faces, a match can then be determined. 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 eigenface exactly. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image.

A variation of the eigenface approach, called eigenfeatures, is also being developed. The eigenfeature approach combines facial metrics, which involves measuring the distance between specific facial features, such as the eyes, nose, and mouth, with the eigenface approach. The eigenfeatures system measures the distance between these points on a live face and compares them to the sets of eigenfeatures stored in the database to determine whether the face is a match. The primary advantage of the eigenface method is the system's speed and efficiency. The eigenface approach reduces the amount of data needed to identify an individual to 1/1000th of a full sized image.

The eigenface recognition system was the first working facial recognition system. The eigenface method provides accurate recognition rates, but has difficulty when presented with face deformities, such as scarring. The eigenface method also has problems identifying faces in different light levels and pose positions. The face must be presented to the system as a frontal view in order for the system to work. When the eigenface method is combined with the eigenfeatures method, the system becomes much more versatile. Greater accuracy can be achieved because of the eigenfeatures method's ability to identify faces with variations such as beards and glasses.

CHAPTER 3
METHODOLOGY

3.1 THE EIGENFACE TECHNIQUE

Methodology of the study is mainly based on the eigenface technique as been chosen to be applied in handling facial recognition in this project. Theoretically the step-by step eigenface approach is summarized from the methodology explained by Turk and Pentland in their paper [4].

3.1.1 Approaching Eigenface Technique

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

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

2. Next, the face images in the training set are normalized.
3. From all the face images, the average image is obtained.

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

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

$$\Phi_i = \Gamma_i - \Psi$$

5. Next we seek 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

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

is a maximum, subject to

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

Note: u_k and k are the eigenvectors and eigenvalues of the covariance matrix **C**

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

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T$$
$$= AA^T$$
$$A = \{\Phi_1, \Phi_2, \Phi_3, \ldots\ldots, \Phi_n\}$$

7. From step 6, we construct the

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

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

8. From that, we have found the eigenvectors, **vl, ul.**

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

3.1.2 Eigenface Recognition Procedure

1. A new face 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.

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

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

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

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

3.2 DATABASE

To start performing the eigenface approach, firstly a database of known faces is provided. The database consist frontal view face images of many different people with some variation in lighting and expressions. In this study, the face images were taken from AT&T Laboratories Cambridge's Face Database and some were taken by me. The face images taken from the AT&T Lab were in the '.pgm' format which could not processed by MATLAB while the face images taken by me were in coloured version and the size differed from those taken from AT&T Lab. Therefore, all the face images need some conversion and modification where face images taken by me were initially edited to the same size as the others taken from the lab and later changed the colour resolution to gray (8 bit) format. Then all the images for the database were converted to 'bitmap' or '.bmp' format. The colour and format conversion is done by using Wavel Pic2Pic software and now the face database is complete and ready to be used.

3.3 ALGORITHM OVERVIEW

From the eigenface approach above, the algorithm overview is constructed to guide the whole program developing process. Basically it contains the main process involved in eigenface technique and the result expected.

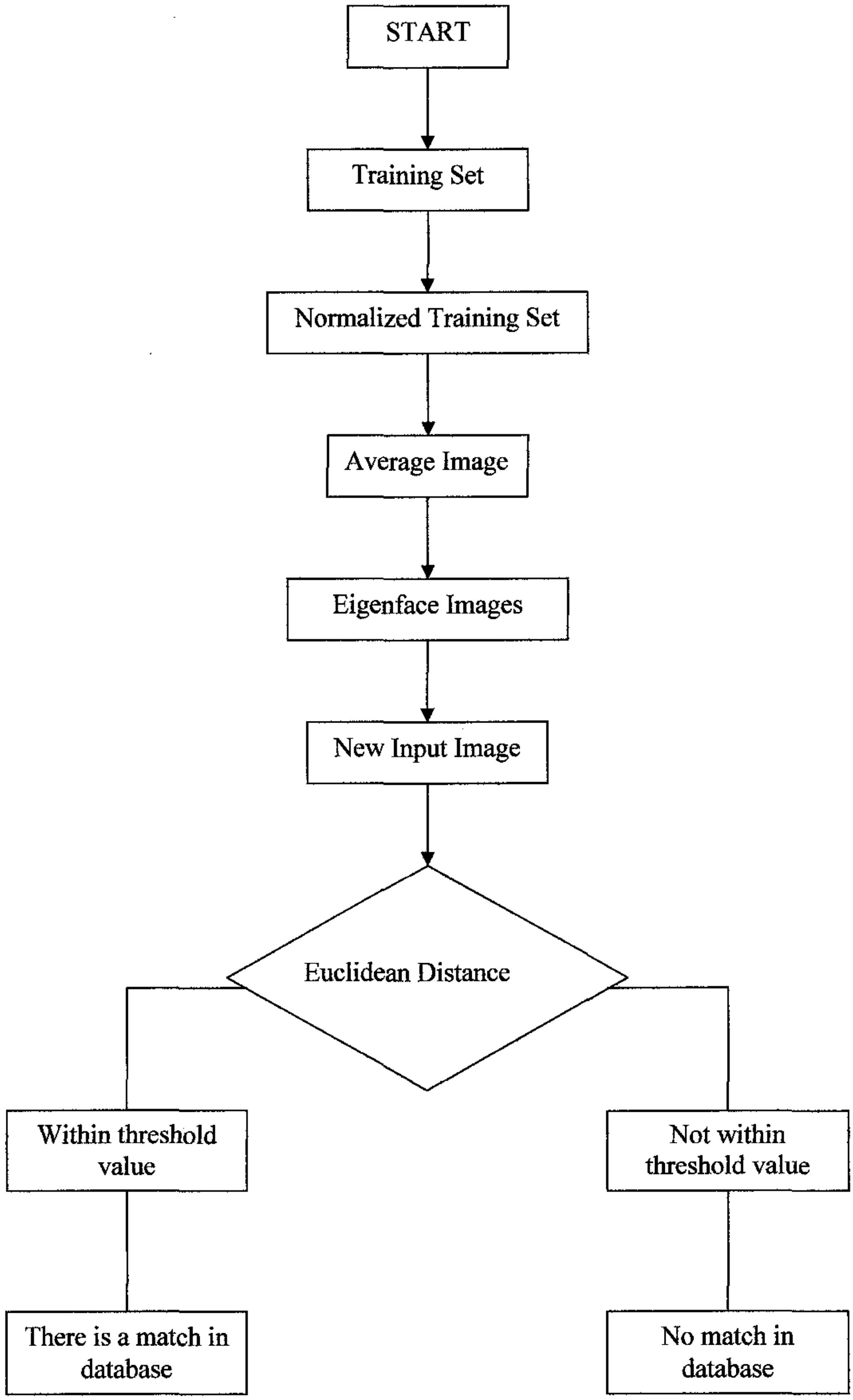


Figure 3.1: Overview of Eigenface Algorithm

3.4 TOOLS APPLIED

3.4.1 MATLAB 6.5.1

MATLAB is a useful tool to be used in this project as it provides the tools and applications on image processing. In the study MATLAB is used to develop the program and perform the eigenface application. It has the ability to generate interface and message box without using the Graphical User Interface Development Environment (GUIDE) toolbox directly.

3.4.2 Digital Camera

A digital camera is used in this project to take the frontal view face images of a few people. The camera is manufactured by Samsung with the maximum resolution of 2.0 mega pixels. The images captured are in digital format which would ease the conversion and editing process of the images. The images are then retrieved directly to the computer for further application.

CHAPTER 4

RESULTS AND FINDINGS

4.1 RESULTS

The facial recognition based on eigenface technique was explained earlier in Section 3.1.1 and 3.1.2 respectively. The source codes developed for processing the eigenface technique, which are implemented in MATLAB is attached in the Appendices. The source code is basically based on the algorithm overview in Section 3.3 where the output images produced followed the algorithm sequence to detect any match found for the input image selected when it is compared to the database of known faces provided earlier. The MATLAB source code is named Facial Recognition System program for ease of reference.

4.1.1 Output of Facial Recognition System

The following are the output images of the facial recognition system program developed. The output is performed and displayed accordingly to the program sequence.

1. **Training Set**

   20 frontal view face images of different people with variation in lighting and expression are chosen to form the database of known faces and be the training set. The training set is displayed as Figure 1 in MATLAB. Referring to the source code in Appendices, the program lines to produce or display the training set are as shown below. The capital M represents the number of face images existed in the training set.

```
%read and show images (bmp);
S=[];
figure(1);
for i=1:M
  str=strcat(int2str(i),'.bmp'
  eval('img=imread(str);');
  subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
```

```
    imshow(img)
      if i==2
      title('Training set','fontsize',14)
    end
    drawnow;
    [irow icol]=size(img);
    temp=reshape(img',irow*icol,1);
    S=[S temp];
end
pause;
close all;
```

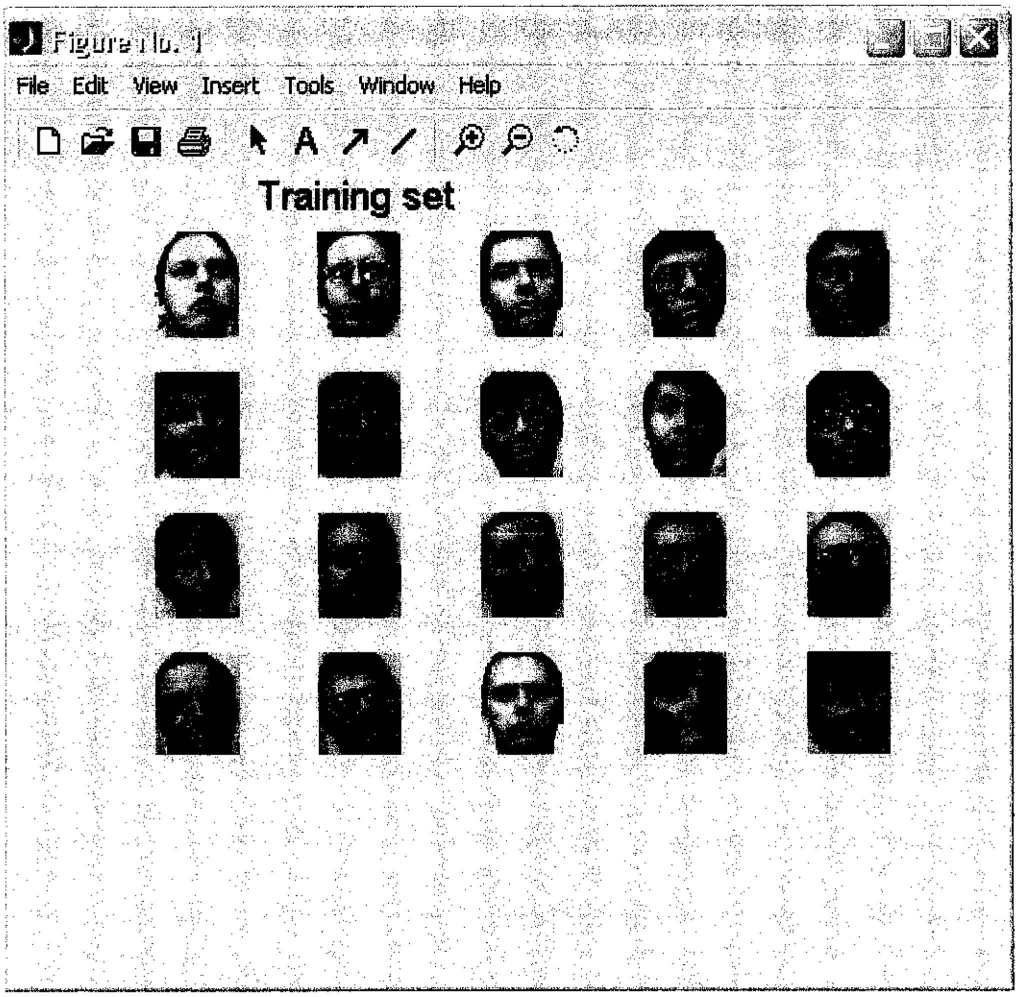


Figure 4.1: Face images in the training set

2. **Normalized Training Set**

As the training set is completely displayed, next the face images in the training set are normalized. The normalization process is usually done to increase the facial recognition system's robustness and for this project the images are normalized to deal with the lighting error. In figure 4.2 below, the normalized training set is the second output shown and when observed, we could see all the face images are brighter and have almost equally lighting coverage than showed in the training set earlier. Output of the training set normalization is obtained when MATLAB processed these lines. For the whole program, the 'pause' command is to enable the output to be displayed one-by-one according to the sequence instead of having all the output images displayed at the same time.

```
%show normalized images
figure(2);
for i=1:M
   str=strcat(int2str(i),'.jpg');
   img=reshape(S(:,i),icol,irow);
   img=img';
   eval('imwrite(img,str)');
   subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
   imshow(img)
   drawnow;
   if i==2
      title('Normalized Training Set','fontsize',14)
   end
end
pause;
close all;
```

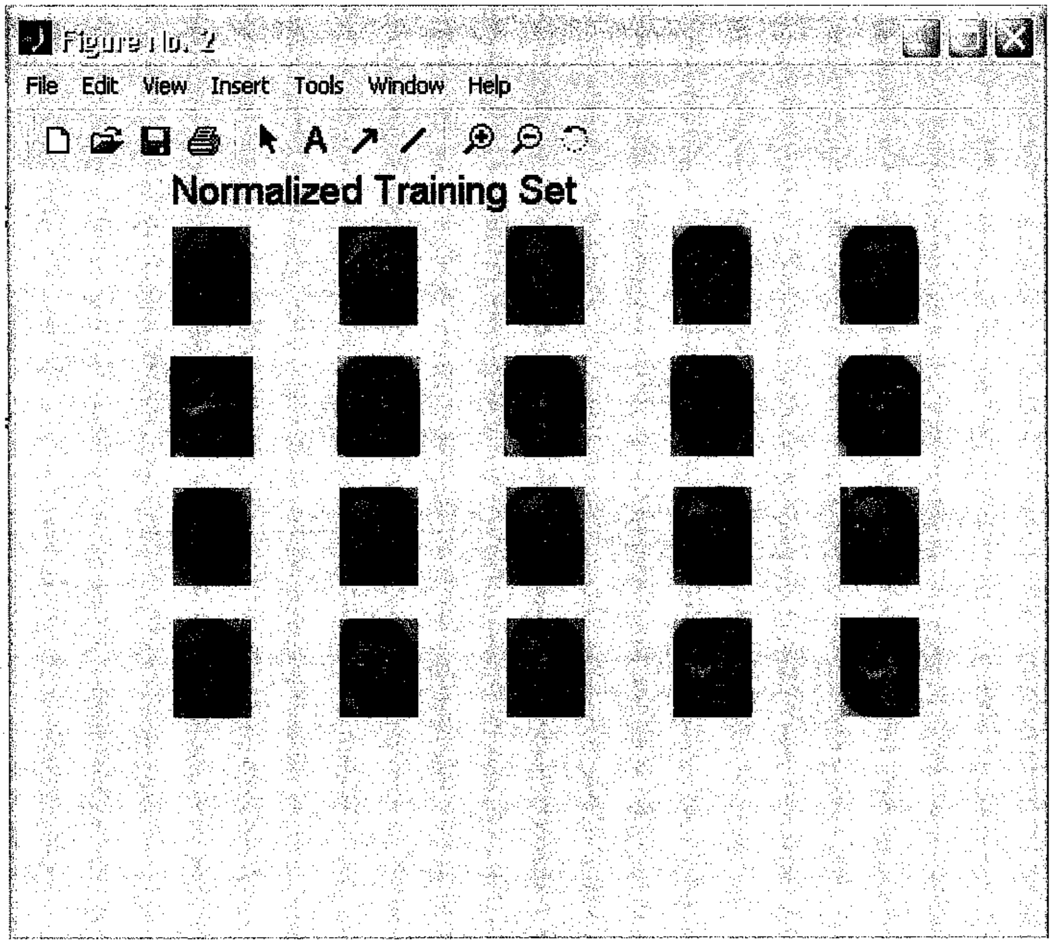


Figure 4.2: Images of the normalized training set

3. **The Mean or Average Image**

After normalization process is completed, the program proceeds to find the mean or average image of all face images in the training set. The following lines calculated and processed the mean image as well displayed the image in Figure No. 3 of all output. The mean image is shown in Figure 4.3 below.

```
%mean image;
m=mean(S,2);
tmimg=uint8(m);
img=reshape(tmimg,icol,irow);
img=img'
figure(3);
```

```
subplot(1,1,1);
imshow(img);
title('Mean Image','fontsize',14)
```

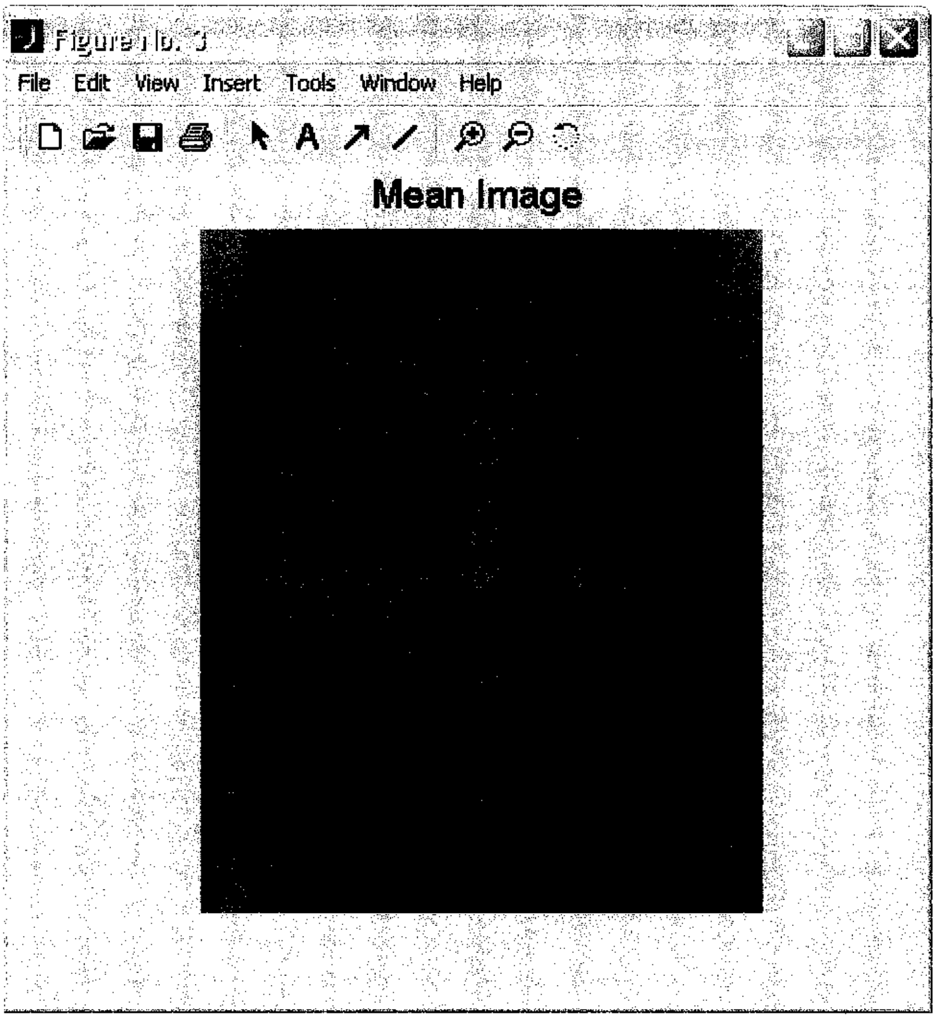


Figure 4.3: The mean or average image

4. The Eigenface Images

According to Dimitri Pissarenko [6], the original face can be reconstructed if all eigenfaces is added up in the right proportion. Basically the eigenface images are formed by the eigenvectors and eigenvalues of each face in the training set. These eigenvectors can be thought of as a set of features that together characterize the variation between face images. In detail, each eigenface represents only certain features of the face, which may or not be present in the original image. If the feature is present in the original image to a higher degree, the share of corresponding eigenfaces is greater. However if contrary, where the particular feature is not or almost not present in the original image, then the corresponding eigenface only contribute a small (or not at all) part of the sum of eigenfaces. Referring to Turk and Pentland, since the eigenfaces seem adequate for describing face images under certain controlled condition, one could choose to practice with smaller number of eigenface as it is sufficient enough for face verification since accurate construction of the image is not a requirement. In other way, it means we do not have to include all calculated eigenface of each image to obtain the result. Sirovich and Kirby [4] gave an example of a limited framework on an ensemble of training set, M = 115 images of Caucasian males, digitized in control manner and found out that about 40 eigenface were sufficient for a very good description of the set of face images with small error percentage.

The training set for this project only consisted of 20 images which in my opinion it is sufficient to use all the eigenfaces to verify the face. The practice on using only part of the eigenface is considered if we have a large number of face images in the training set. Since we have 20 face images in the training set, therefore there are 20 eigenface images which are likely ghostly faces displayed in the output image as shown in Figure 4.4. To process the eigenface images, one needs to follow the steps and formulas listed in Chapter 3. It involved manipulation of the formulas and eigenvectors calculation to finally able to show the eigenface images. To see the program lines involved in this phase, please refer to the source code attached in the Appendices.

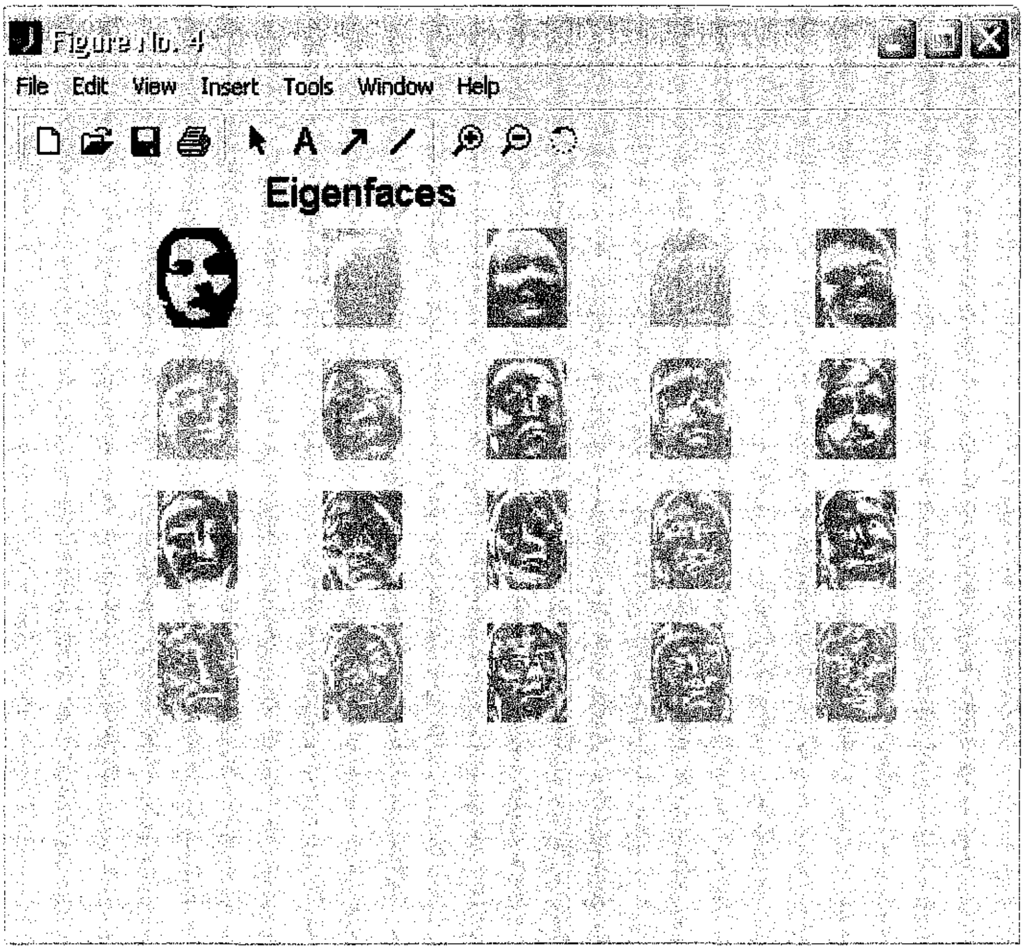


Figure 4.4: The Eigenface Images

5. Menu Box

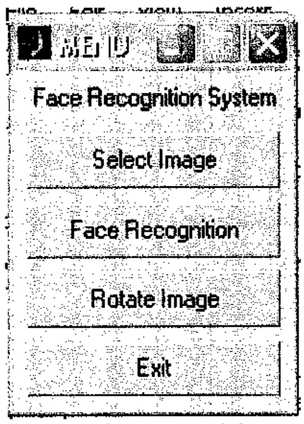


Figure 4.5: Facial Recognition Menu Box

The Menu Box will appear when the eigenface images is completely displayed and closed. As shown in Figure 4.5, there are four taskbars in the Menu Box:

i. Select Image – to select the input image required

ii. Face Recognition – to process the recognition of selected input image

iii. Rotate Image – to rotate the selected input image to certain degrees

iv. Exit – to exit from the Face Recognition System

6. Select Image

The input image selection can be divided into two parts.

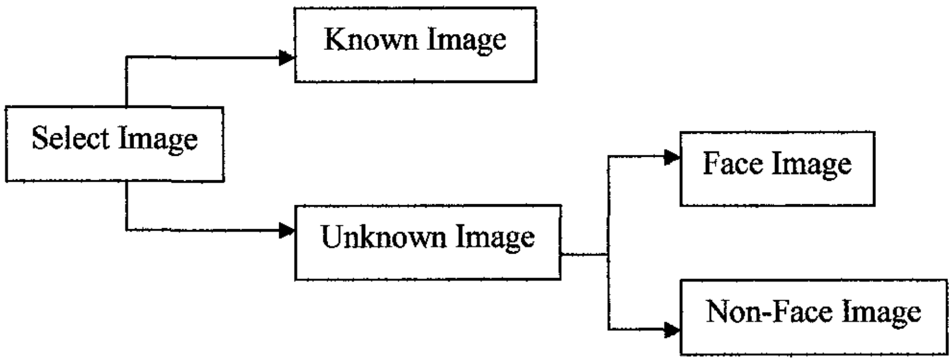


Figure 4.6: Parts of Image Selection

User could choose to select a known or an unknown image to be processed. The known image is the image already existed in the database while the unknown image is not in the database and could be a face image or non-face image. For any selection made, the Face Recognition taskbar is selected to process the input image and produce the output which consists of the input image, reconstructed image and two diagrams of the weight of input face and Euclidean Distance. The Matching Result box also appears at the same time. Examples of the input image selections with the results produced are discussed in the next section.

7. Rotate Image

Figure 4.5 shows one of the taskbar in the Menu Box is 'Rotate Image'. The 'Rotate Image' taskbar is provided to rotate the selected input image before it being processed under face recognition. It could be used in situation where the input image is having an obvious angle different or whenever is necessary. When the 'Rotate Image' taskbar is selected, another Menu Box will appear as shown in Figure 4.7.

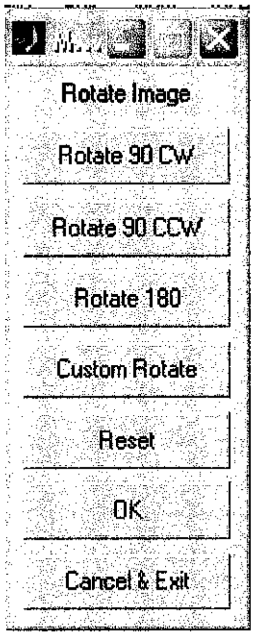


Figure 4.7: Menu Box for Rotate Image Taskbar

The above Menu box provided 7 taskbars to help the user to rotate the input image until the angle different is reduced and look almost vertically as the other face images. The image can be rotated to 90 degrees clockwise and counterclockwise and even to 180 degrees. Besides that, the user can also choose the rotating angle using the 'Custom Rotate' taskbar to fill in the angle required other than provided in that Menu box. The Reset, OK and Cancel & Exit taskbars are for resetting the rotating angle, to confirmed rotation and start processing and to exit the rotating phase respectively.

4.2 INPUT IMAGE SELECTION AND RESULTS PRODUCED

In this section we will see examples of the three input image selection; a known face, an unknown face or a non-face image together with the results produced after they were processed under facial recognition. Firstly, a known image is selected as the input image. The following three figures show the output produced and its detail description is explained later. The output basically shows the input image, reconstructed image, weight of input face, Euclidean distance of input image and the matching result.

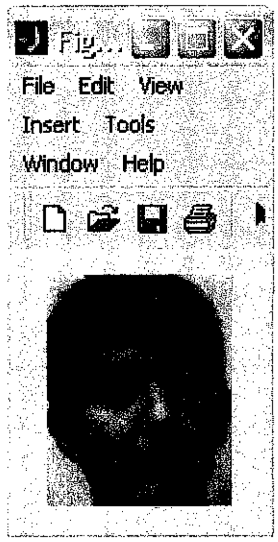


Figure 4.8: Example of a Known Input Image

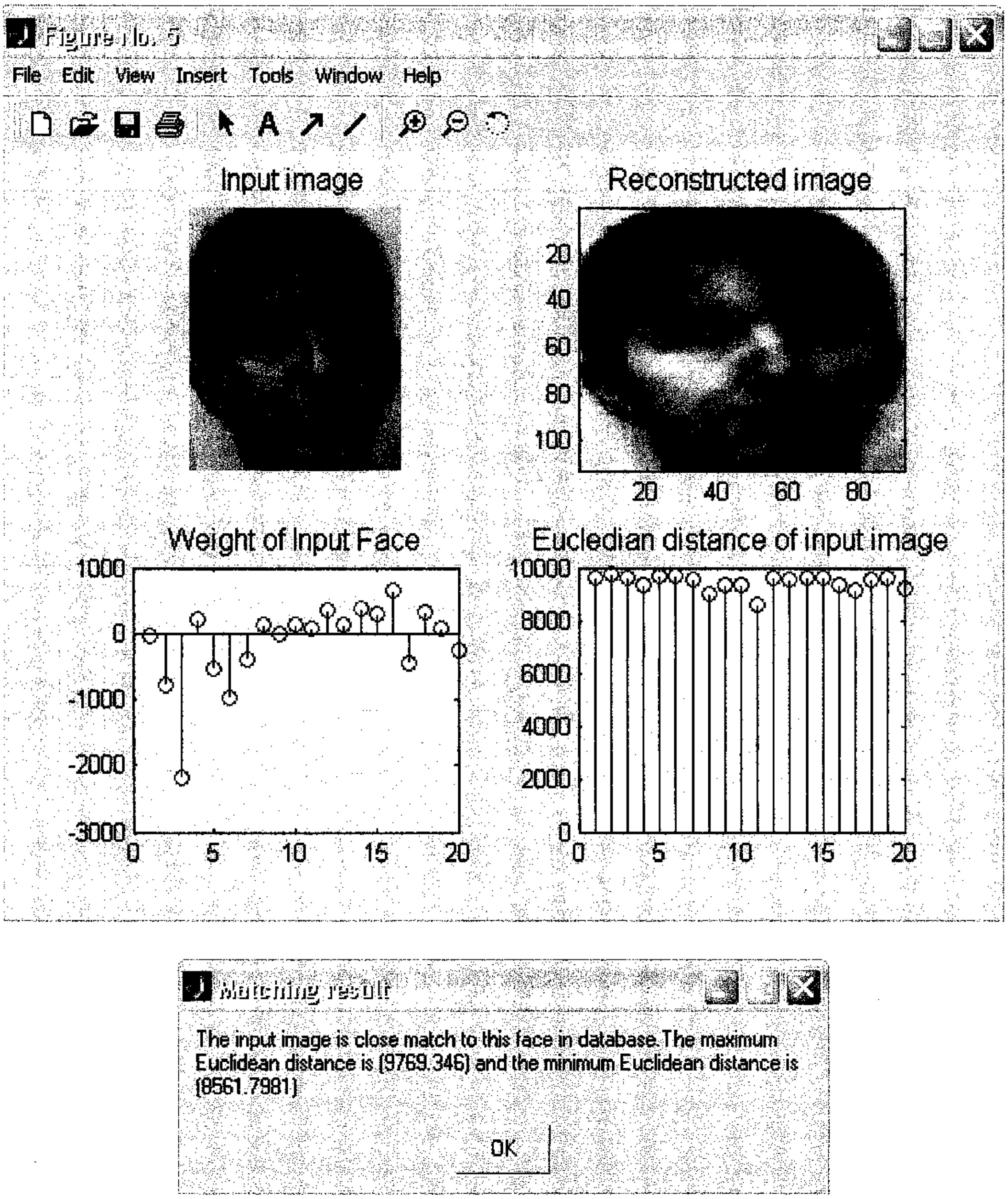


Figure 4.9: The Output and Matching Result Box

4.2.1 Reconstructed Image and Weight of Input Face

The reconstructed image is an approximation of the original image as it is equal to the sum of all eigenfaces, which have a certain weight. This weight specifies to what degree the specific feature (eigenface) is present in the original image. If we use all the eigenfaces extracted from original images, we can reconstruct the original images from the eigenface exactly but as mentioned earlier, we could also use only part of the eigenfaces.

4.2.2 Euclidean Distance

Euclidean distance tells us how close the input image is from the images in the training set. Using the Euclidean distance we can determine the input image class, whether it is a known or an unknown face, or not even a face at all. The determination depends on the maximum and minimum Euclidean distance value. In this project, the maximum and minimum Euclidean distance is considered as the threshold value of the facial recognition system. Based on research and testing on all face images in the training set, it is found out that the threshold value of the system should be within the range of less than 8950 to 10000. The table below shows the input image class determined by the threshold value.

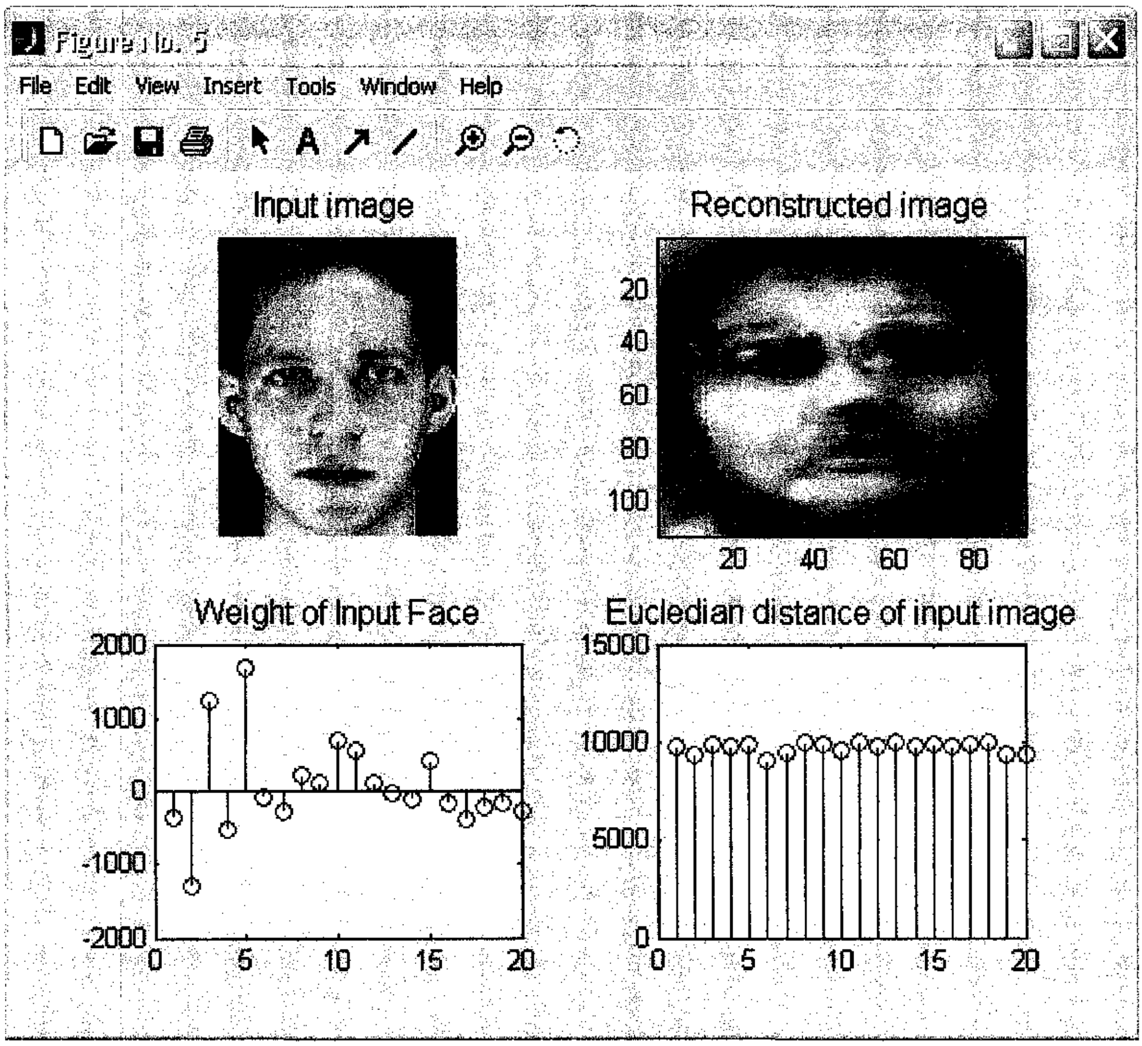
Threshold Value (Euclidean Distance)		**Input Image Classification**
Minimum	**Maximum**	
min < 8950	9000 < max < 10,000	Known Face
8950 < min < 9200	max > 9,000	Unknown Face
min > 9,200	max > 9,000	Non-face

Table 1: The Input Image Classification Based on Threshold Value

As shown in Figure 4.9, the maximum and minimum Euclidean distance of input image in Figure 4.8 is 9769 and 8561 respectively which is between 9000 and 10000 and less than 8950. The Matching Result box acknowledges us that there is a match found for the corresponding input image when it is compared to the database and is shown in the reconstructed image. This has confirmed that the input image is a known face.

A second different image which is not from the database is selected as the input image as shown in Figure 4.10.

Figure 4.10: Second Input Image Selected



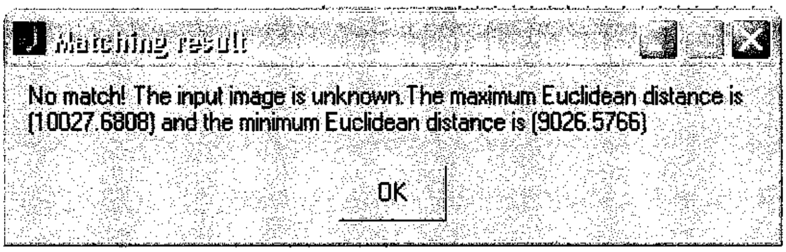


Figure 4.11: Results for the Second Input Image

After being processed, the output in Figure 4.11 shows both the minimum and maximum Euclidean distance of the input image exceeded the threshold value, however the minimum Euclidean distance is still less than 9200 which makes it still a face. There is no match found for that corresponding input image in the database, hence the second input image is considered as an unknown face. The reconstructed image shows the unmatched face.

Finally, an image of non-face is selected as the input image. It is shown in Figure 4.12 below.

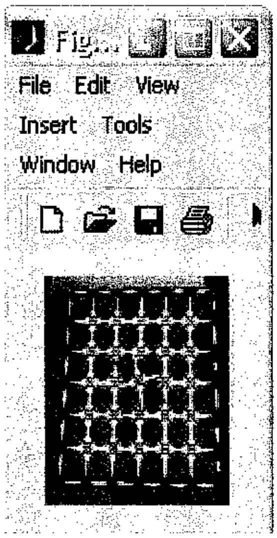


Figure 4.12: A Non-face Input Image

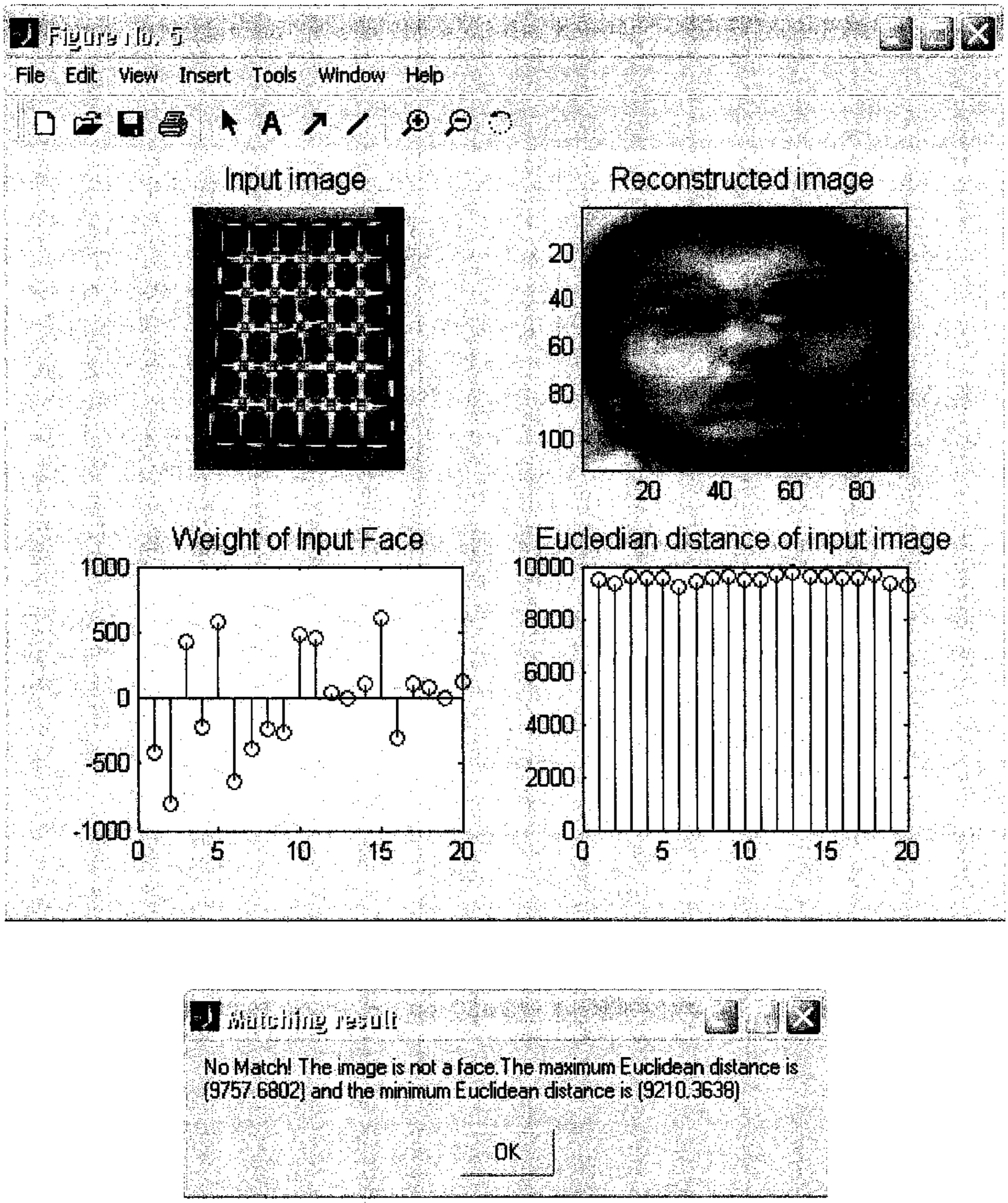


Figure 4.13: Result for Non-face Input Image

Similar with other input images, the image in Figure 4.12 is processed and produced result as shown in Figure 4.13 above. Observing the result, its maximum Euclidean distance is more than 9000 and the minimum Euclidean distance value has exceeded the minimum threshold value limit, 9200. Therefore, there is no match for this input image in the database and it is classified as a non-face image.

4.3 EXPERIMENT AND TESTING

In section 1.3.2 stated that the input image and database is mainly consisting frontal view face images. As shown previously, the program developed does conform to the frontal view scope hence an experiment is done involving face images with some angles as the input image to see the possible outcomes. There are two face images with angle variation selected for this experiment which the non-angle face for both images were already in the database or training set. Figure 4.14 and 4.15 shows the first and second face image with angle variation respectively. Both images labeled with (a) and (b) for ease of discussion later.

Figure 4.14: First Input Image with Angle Variation (a)

Figure 4.15: Second Input Image with Angle Variation (b)

Similar to what has been discussed before, the above face image are processed under facial recognition and produced the following result shown in Figure 4.16.

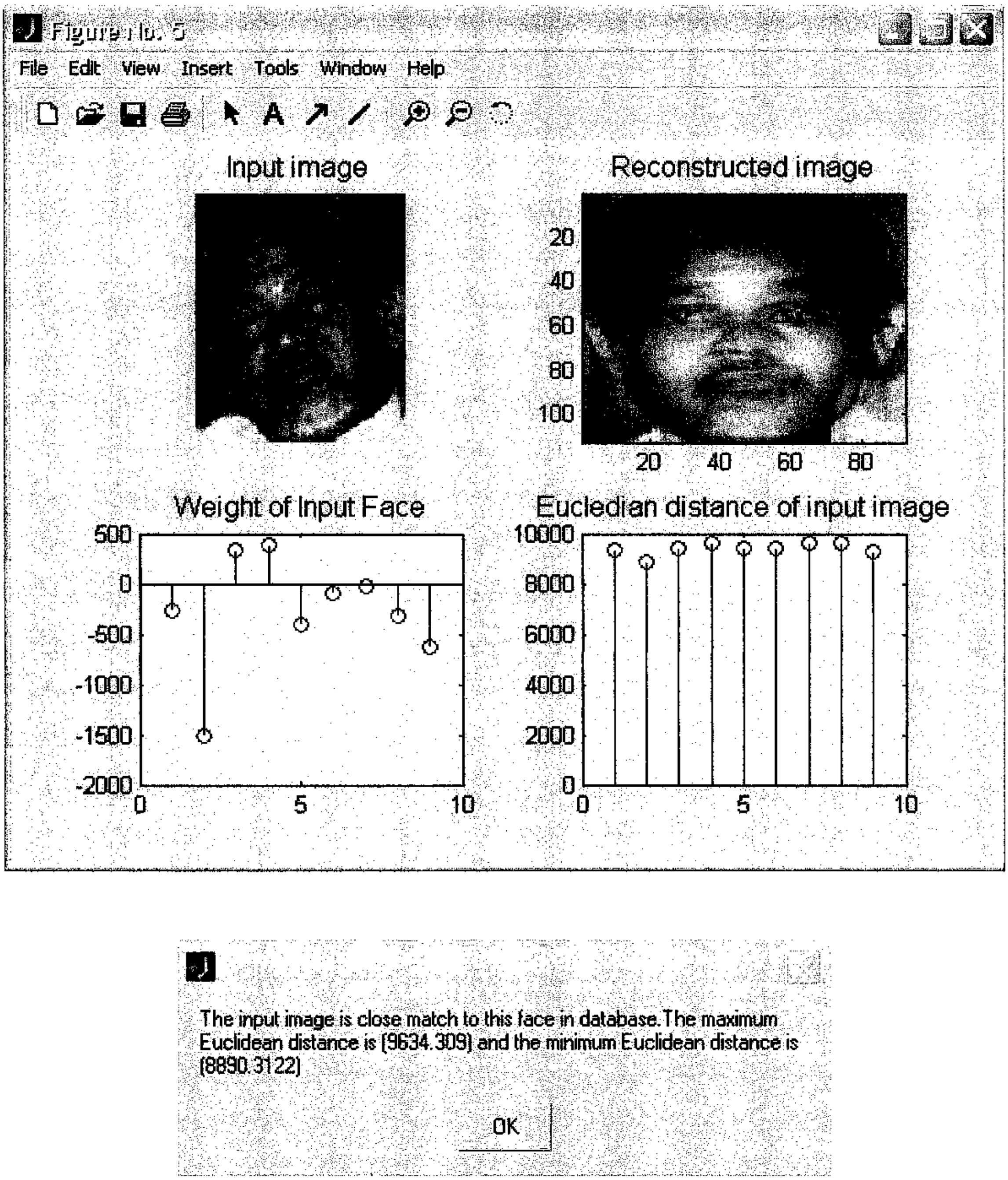


Figure 4.16: Results for the Input Image (a)

Based on Figure 4.16 above, it is shown that the maximum and minimum Euclidean distance is within the threshold value which considered the input image is acceptable for the system. However, the reconstructed image does not give the exact match as other results for the frontal view images. Having a close look to the reconstructed image, it is almost exactly match the input image but it still has some errors as the image is a bit blur.

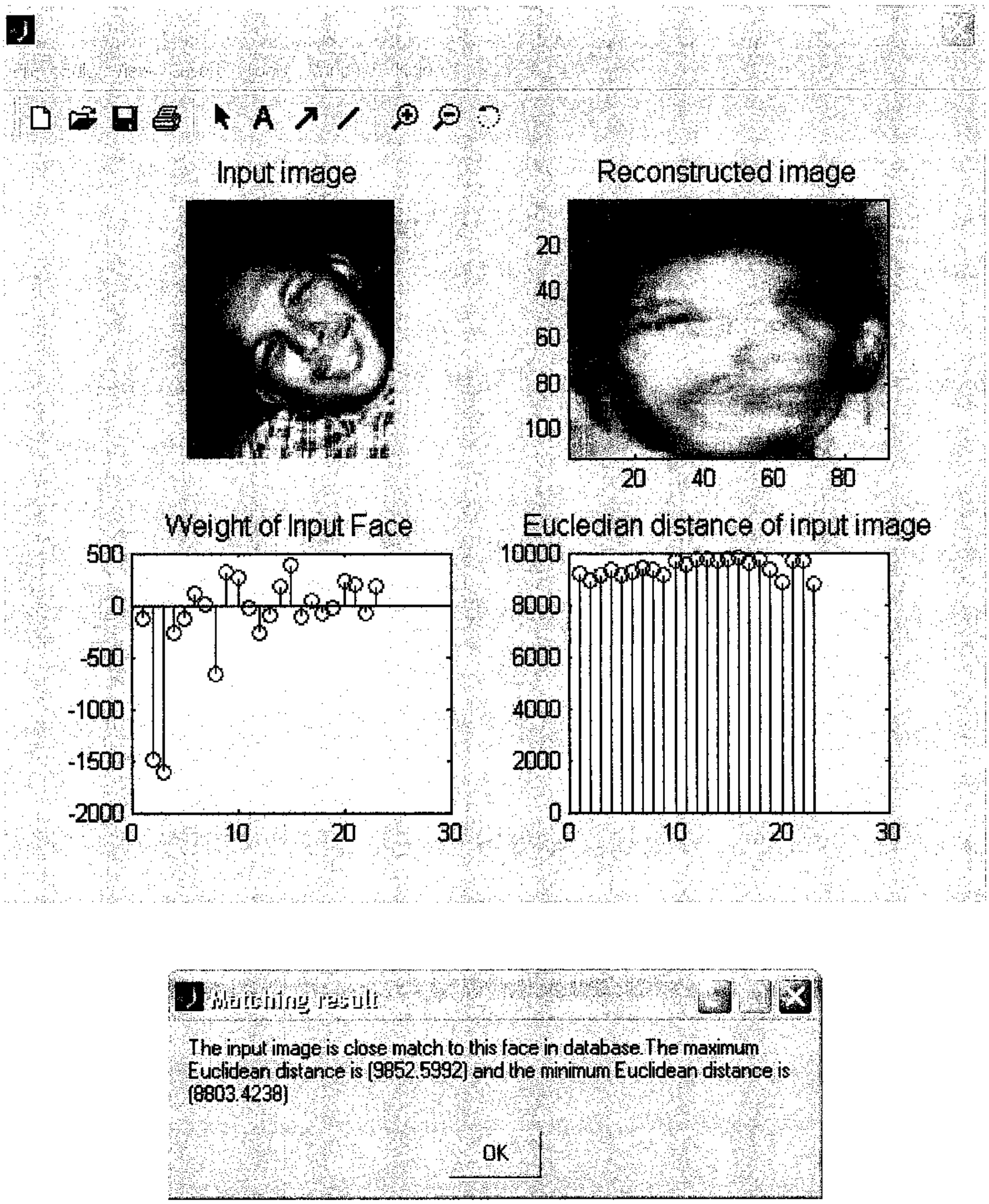


Figure 4.17: Results for Input Image (b)

For a different view, the result for input image (b) is analyzed. It is shown in Figure 4.17 that the minimum and maximum Euclidean distance for this input image also falls in the acceptable threshold value range but as can be seen the reconstructed image does not give a match. In fact, the image is almost unknown compared to the reconstructed image in Figure 4.16. The reconstructed image in Figure 4.16 almost matches with its input image (a) as the angle variation of the input image is small. Comparing with input image (b), it has a larger angle variation which the angle different is more obvious than in input image (b). Thus, the reconstructed image is not matched and can be considered as unknown face.

Based on these result, it is necessary to use the Rotate Image function in the Main Menu Box to test whether it could gives a better result. Using the Menu Box for Rotate Image Taskbar, the input image (a) is rotated until the angle different is no more obvious as shown in Figure 4.18 below.

Figure 4.18: Rotated Input Image (a)

After rotating process is completed, the rotated input image is processed and the result in Figure 4.19 is obtained. The result shows that it is within the threshold value but there is still no improvement to the reconstructed image. Suppose the rotated input image would give a better result as it has transformed to be a frontal view image. Therefore, from the results for both not rotated and rotated input image it is shown that only frontal view images will be successfully processed by the program. For face images with angle variation, it remains as the problem faced in this project because a lot more research and work need to be done to deal with the problem.

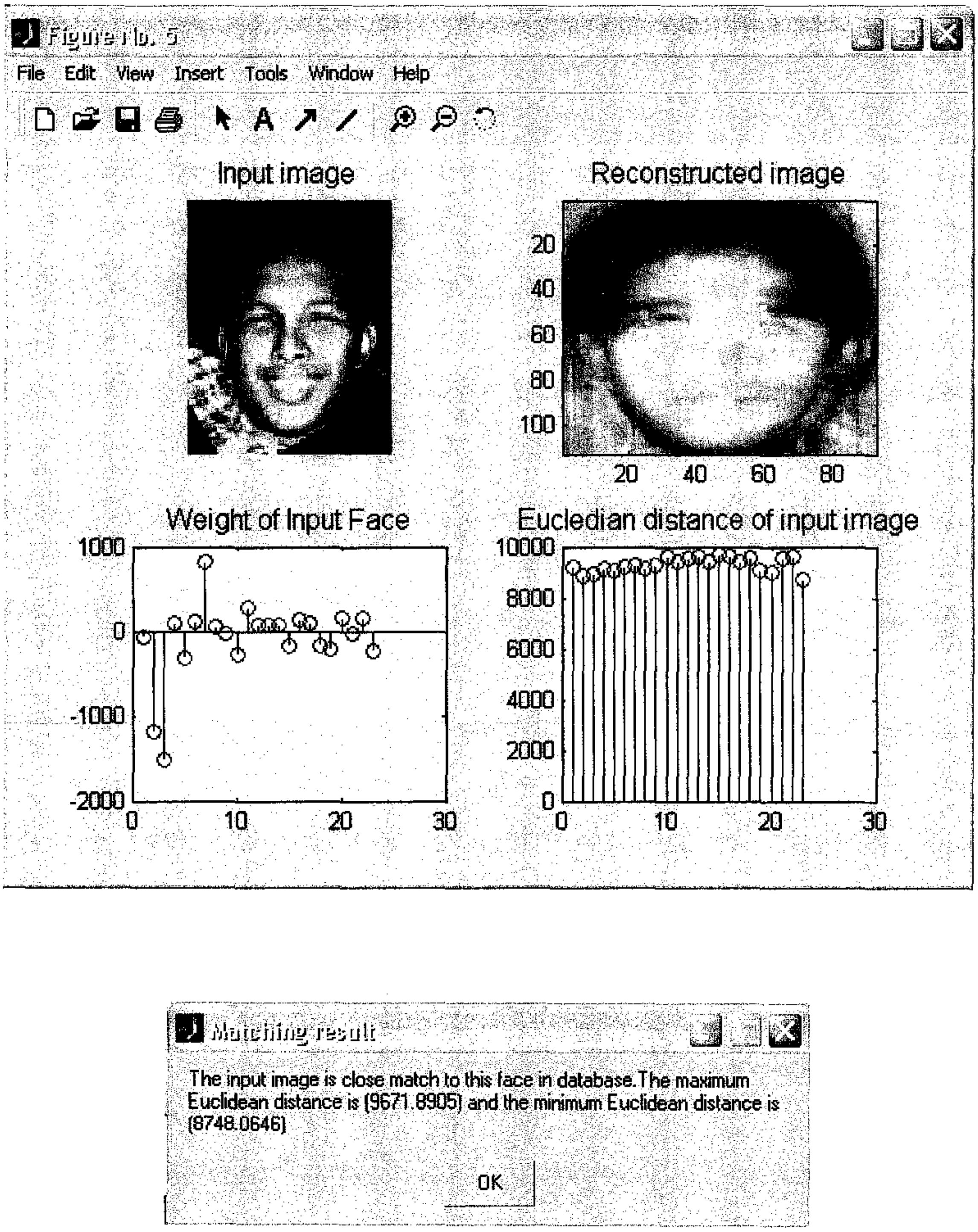


Figure 4.19: Result for Rotated Input Image (a)

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 RELEVANCY TO THE OBJECTIVES

Based on the project title, 'Biometrics: Facial Recognition System', the main objective to achieve in this project is to develop a MATLAB program which can be used to verify a face when it is compared to the database of known faces. Starting from understanding the biometrics and face recognition categories, until practicing the project based on the eigenface technique; the main objective of this project had successfully achieved. Four categories of facial recognition learnt; face detection, facial feature extraction, face normalization and face identification. This project falls under the face identification category. The Facial Recognition System program is developed and has the ability to verify a face when compared to the database provided. The overall process take part in the eigenface recognition technique, is shown phase by phase and finally produced the verification result. Verification of a face depends on the threshold value (Euclidean distance) which it needs to be within the range of less than 8950 and 10,000. The verified face is divided into three input image class; a known face, an unknown face and a non-face image. A known face has a match for it in the database. The unknown face is however still a face image although no match is found for it in the database and a non-face is not a face at all.

5.2 SUGGESTED FUTURE WORK FOR EXPANSION AND CONTINUATION

Future work can be done to improve the accuracy of the results as discussed in Chapter 4, especially on the threshold value. Different training set of face images would give a different threshold value and this is being left as a suggested work for continuation. Facial recognition system is improving through time thus it need to provide more feasible features regarding human faces. For the time being, the face images involved in this project are basically frontal view images. Therefore, for further work expansion, the project should be continued to handle face images with different position or angles and involved face images with additional factors such as wigs, mask and others.

REFERENCES

[1]. K. Kalairasi, Aliza Tajudin. R.K Subramaniam, A New Real – Time face Recognition System Based On Eigenfaces, School of Computer Science, University of Science Malaysia

[2]. Face Recognition Based On Symmetryzation and Eigenfaces

[3]. Eigenface Recognition – Eigenfaces.htm

[4]. Matthew Turk and Alex Pentland, Eigenfaces for Recognition, Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology.

[5]. Kyungnam Kim, Face Recognition using Principal Component Analysis, Department of Computer Science, University of Maryland, College Park

[6]. Dimitri Pissarenko, 2003, Eigenface – Based Recognition

[7]. Anil K. Jain, Arun Ross, Salil Prabhakar, January 2004, An Introduction to Biometric Recognition, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, Vol. 14.

[8]. Ilker Atalay, January 1996, M.Sc. Thesis, Face Recognition Using Eigenfaces, Istanbul Technical University.

[9]. Luigi Rosa, Face Recognition System based on Eigenface Method (MATLAB program), Italy.

[10]. www.pages.drexel.edu

[11]. Biometrics-A Look at Facial Recognition, RAND, 2003, John D. Woodward Jr, Christopher Horn, Julius Gatune and Aryn Thomas.

[12]. The MATLAB Handbook, 1st Edition, 1996, Permilla Isaksson, Bo Lin, Addison Wesley Publisher.

[13]. Matthew Dailey, Matlab Tutorial Source Code Page, Matlab_Tutorial.htm

APPENDICES

MATLAB Facial Recognition System Source Code

```
clear all
close all
clc

% number of images on your training set.
M=23;

%Chosen std and mean.
%It can be any number that it is close to the std and mean of most of the images.
um=83;
ustd=32;

%read and show images(bmp);
S=[]; %img matrix
figure(1);
for i=1:M
   str=strcat(int2str(i),'.bmp');  %concatenates two strings that form the name of the image
   eval('img=imread(str);');
   subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
   imshow(img)

   if i==2
      title('Training set','fontsize',14)
   end
   drawnow;
   [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
pause;
close all;

%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

%show normalized images
figure(2);
for i=1:M
   str=strcat(int2str(i),'.jpg');
   img=reshape(S(:,i),icol,irow);
   img=img';
```

```
    eval('imwrite(img,str)');
    subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
    imshow(img)
    drawnow;
    if i==2
       title('Normalized Training Set','fontsize',14)
    end
end
pause;
close all;

%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
img=img';      %creates a N1xN2 matrix by transposing the image.
figure(3);
subplot(1,1,1);
imshow(img);
title('Mean Image','fontsize',14)

pause;
close all;

% Change image for manipulation
dbx=[];  % A matrix
for i=1:M
    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


% show eigenfaces;
figure(4);
for i=1:size(u,2)
   img=reshape(u(:,i),icol,irow);
   img=img';
   img=histeq(img,255);
   subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
   imshow(img)
   drawnow;
   if i==2
      title('Eigenfaces','fontsize',14)
   end
end

pause;
close all;

% 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
```

```
% Acquire new image

if exist ('Rotate.bmp')
   InputImage=imread('Rotate.bmp');
   imshow(InputImage);
end
Selection=0;
Opt_No=4;
   while Selection~=Opt_No,
   Selection=menu('Face Recognition System','        Select Image        ','Face Recognition','Rotate
Image','Exit');
      if Selection==1
         [namefile,pathname]=uigetfile('*.bmp','Select an image (.bmp or .jpg)');
         if namefile~=0
            InputImage = imread(strcat(pathname,namefile));
            imshow(InputImage);
         else
         warndlg('Input image must be selected.',' Warning ')
         end
      end

   if Selection==2
      if exist ('InputImage')
         close all;
         figure(5);
         subplot(2,2,1)
         imshow(InputImage); colormap('gray'); title('Input image','fontsize',12)
         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.
         subplot(2,2,2)
         imagesc(ReshapedImage); colormap('gray');
         title('Reconstructed image','fontsize',12)

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

```
InImWeight = [InImWeight; WeightOfInputImage];
end

ll = 1:M;

subplot(2,2,3)
stem(ll,InImWeight)
title('Weight of Input Face','fontsize',12)

% 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);
subplot(2,2,4)
stem(kk,e)
title('Eucledian distance of input image','fontsize',12)

clc;

MaximumValue=max(e)
MinimumValue=min(e)

   %Set threshold values
   if (MaximumValue>9000 )&& (MinimumValue>8950 ) && (MinimumValue<9200)
      msg1='No match! The input image is unknown. ';
      msg2=strcat('The maximum Euclidean distance is (',num2str(MaximumValue));
      msg3=strcat(') and the minimum Euclidean distance is (',num2str(MinimumValue),')');
      msgbox(strcat(msg1,msg2,msg3),'Matching result');
%          msgbox(strcat('No match! The input image is unknown. The maximum Eucledian distance is
',num2str(MaximumValue,' and the minimum Euclidean distance is ',num2str(MinimumValue)),'Matching
result'));
   end
   if (MaximumValue>9000 ) && (MinimumValue>9200 )
      msg1='No Match! The image is not a face. ';
      msg2=strcat('The maximum Euclidean distance is (',num2str(MaximumValue));
      msg3=strcat(') and the minimum Euclidean distance is (',num2str(MinimumValue),')');
      msgbox(strcat(msg1,msg2,msg3),'Matching result');
%         msgbox(strcat('No match! The input image is unknown. The
%         maximum Eucledian distance is ',num2str(MaximumValue,' and
%         the minimum Euclidean distance is
%         ',num2str(MinimumValue)),'Matching result'));

   end

   if (MaximumValue>9000) && (MaximumValue<10000) && (MinimumValue<8950)
      msg1='The input image is close match to this face in database. ';
      msg2=strcat('The maximum Euclidean distance is (',num2str(MaximumValue));
      msg3=strcat(') and the minimum Euclidean distance is (',num2str(MinimumValue),')');
      msgbox(strcat(msg1,msg2,msg3),'Matching result');
```

```
        %msgbox(strcat('The input image is match to this face in database. The maximum Eucledian distance is ',num2str(MaximumValue,' and the minimum Euclidean distance is ',num2str(MinimumValue)),'Matching result'));
        end

      pause;
      close all;
      imshow(InputImage);
    else
      warndlg('Must select an input image.',' Warning ')
    end
  end

  if Selection==3
     if exist('InputImage')
       %close all;
       imshow(InputImage);
       Selection1=0;
       Opt_No1=7;
         while Selection1~=Opt_No1,
           Selection1=menu('Rotate Image','Rotate 90 CW','Rotate 90 CCW','Rotate 180','Custom Rotate','Reset','OK','Cancel & Exit');
           if Selection1==1
             close;
             InputImage=imrotate(InputImage,-90,'bicubic','crop');
             imshow(InputImage);
           end

           if Selection1==2
             close all;
             InputImage=imrotate(InputImage,90,'bicubic','crop');
             imshow(InputImage);
           end

           if Selection1==3
             close all;
             InputImage=imrotate(InputImage,180,'bicubic','crop');
             imshow(InputImage);
           end

           if Selection1==4
             close all;
             prompt={'Specify angle. Use negative degrees for CCW'};
             title='Rotation Angle';
             lines=1;
             def={'0'};
             answer=inputdlg(prompt,title,lines,def);
             AngleRotate=double(str2num(char(answer)));
             AngleRotate=AngleRotate*(-1);

             if size(AngleRotate,1)~=0
               InputImage=imrotate(InputImage,AngleRotate,'bicubic','crop');
               imshow(InputImage);
             else
               msgbox('Please specify the angle');
               imshow(InputImage);
```

```
            end
          end

          if Selection1==5
            close all;
            InputImage = imread(strcat(pathname,namefile));
            imshow(InputImage);
          end

          if Selection1==6
            close all;
            msgbox('This program will restart. Please wait...');
            pause(2);
            close all hidden;
            imwrite(InputImage,'Rotate.bmp');

            Saarah_Face_Recog_System;                %CHANGE THIS TO THIS PROGRAM
FILENAME
          end

          if Selection1==7
            close all;
            InputImage = imread(strcat(pathname,namefile));
            imshow(InputImage);
          end
        end

    else
      msgbox('Please select an input image');
    end
  end

  if Selection==4
    clc;
    if exist ('Rotate.bmp')
      delete Rotate.bmp;
      close all;
    else
      close all;
    end

  end
end
```