FACE DETECTION AND RECOGNITION METHODS

by

Bedirhan Sayar

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronics Engineering

Yeditepe University
2006

FACE DETECTION AND RECOGNITION METHODS

APPROVED BY:

Assoc. Prof. Dr. Aydın Akan
(Thesis Supervisor)

Assoc. Prof. Dr Canbolat Uçak

Assist. Prof. Dr. Soner Özgünel

DATE OF APPROVAL: 09/06/2006

ACKNOWLEDGEMENTS

This project that i studied, is a new and developing subject about security systems which is very important for us. I have designed a software with MATLAB 7.0.1 language.

I want to add my special thanks to my instructor Dr. Aydın Akan for supporting me every time, giving precious suggestions and for special helps during my project. I also want to add my thanks to my family, for their special helps during my education period.

ABSTRACT

FACE DETECTION AND RECOGNITION METHODS

One of the most succesful applications of image analysis and understanding is face recognition that can be obtained by using several methods. Eigenface method is one of these mehods in which a small set of characteristic pictures are used to describe the variation between face images. Goal is to find out the eigenvectors (eigenfaces) of the covariance matrix of the distribution, spanned by a training set of face images. Later, every face image is represented by a linear combination of these eigenvectors. Evaluation of these eigenvectors are quite difficult for typical image sizes but, an approximation can be made that is suitable for practical purposes. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces and then classifying the face by comparing its position in face space with the positions of known individuals.

Eigenfaces approach seems to be an adequate method to be used in face recognition due to its simplicity, speed and learning capability. In this project a face recognition system, based on the eigenfaces approach is proposed.

ÖZET

YÜZ BULMA VE TANIMA YÖNTEMLERİ

Foto analizinin en onemli uygulamalarından biriside cesitli metodlar kullanılarak yapılan yuz tanıma islemidir. Özyüz tanıma yöntemide bu metodlardan birisidir ve resimler arasındaki farkları tanımlamak icin o resimlerin küçük gruplar halindeki karakteristik resimlerini kullanır. Buradaki amaç dağılımın kovaryans matrisinin özvektörlerini yani özyüzleri bulmaktır. Tipik resim boyutları için bu özyüzleri bulmak oldukça zordur ve bu yüzden pratik uygulamalarda yakınsamalar yapılabilir. Tanıma islemi, yeni resmi özyüzlerin oluşturduğu uzaya projeksiyonu ile yapılır. Bu projeksiyon sonucunda, yeni resmin pozisyon itibarıyla en yakın olduğu yüz seçilir.

Özyüz tanıma yöntemi kolaylığı, hızı ve kolay öğretilebilirliği yüzünden yeterli ve iyi bir yöntemdir.

Bu projede özyüz yöntemi kullanılarak bir yüz tanıma işlemi gerçeklenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

Γ	Training set
Ψ	Average matrix of the training set
Φ	Distribution obtained by subtracting the mean from each face
C	Covariance Matrix
u	eigenfaces
N	Matrix dimension
Ω	Feature vector
M	Number of images
PCA	Principle Component Analysis
UFR	Ultimate Face Recognition
ε	Euclidean distance between faces
A	Distribution matrix of the training set
λ_k	Eigenvalues of the covariance matrix with N^2 by N^2 dimension
L	New covariance matrix with M by M dimension
W	Weights
V	Eigenvectors of the L matrix

1. INTRODUCTION

Digital technology allows manipulating the multidimensional signals with systems that range from simple digital circuits to advanced parallel computers. One of the most important areas of the digital technology is biometric operations.

Biometry, instead of a key or something like a password, uses personal characteristics to identify a person. These characteristics may be faces, fingerprints, irises or veins, or behavioral characteristics like voice, handwriting or typing rhythm. They are extremely difficult to lose or to forget unlike a surgery or accident occurs. They can also be very difficult to copy. For this reason, many people consider them to be safer and more secure than keys or passwords.

Face recognition is one of the most important biometric technology. It has become an important issue in many applications such as security systems, credit card verification and criminal identification. For example, the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to improve criminal identification. Detecting faces in photographs for automating color film development can be very useful, since the effect of many enhancement and noise reduction techniques depends on the image content.

In this thesis project, a facial recognition system is designed to be used as a security device and different biometric technologies are investigated.

2. TYPES OF BIOMETRIC TECHNOLOGIES

2.1. Handwriting Recognition

At first glance, using handwriting to identify people might not seem like a good idea. After all, many people can learn to copy other people's handwriting with a little time and practice. It seems like it would be easy to get a copy of someone's signature or the required password and learn to forge it.

But biometric systems don't just look at how person shape each letter; they analyze the act of writing. They examine the pressure of the person uses and typing speed. They also record the sequence in which person form letters, like whether he/she add dots and crosses as he/she goes or after finishes the word.

Unlike the simple shapes of the letters, these traits are very difficult to forge. Even if someone else got a copy of his/her signature and traced it, the system probably wouldn't accept their forgery.

As shown in figure 2.1 a handwriting recognition system's sensors can include a touch-sensitive writing surface or a pen that contains sensors that detect angle, pressure and direction. The software translates the handwriting into a graph and recognizes the small changes in a person's handwriting from day to day and over time.

Figure 2.1 Touch sensitive writing surface

People's hands and fingers are unique but not as unique as other traits, like fingerprints or irises. That's why businesses and schools, rather than high-security facilities, typically use hand and finger geometry readers to authenticate users, not to identify them. Disney theme parks, for example, use finger geometry readers to grant ticket holders admittance to different parts of the park. Some businesses use hand geometry readers in place of timecards.

Systems that measure hand and finger geometry use a digital camera and light. To use one, person simply places his/her hand on a flat surface, aligning his/her fingers against several pegs to ensure an accurate reading. Then, a camera takes one or more pictures of his/her hand and the shadow it casts. It uses this information to determine the length, width, thickness and curvature of his/her hand or fingers. It translates that information into a numerical template.

Hand and finger geometry systems have a few strengths and weaknesses. Since hands and fingers are less distinctive than fingerprints or irises, some people are less likely to feel that the system invades their privacy. However, many people's hands change over time due to injury, changes in weight or arthritis. Some systems update the data to reflect minor changes from day to day.

2.2. Voice Recognition

The person’s voice is unique because of the shape of his/her vocal cavities and the way person’s move his/her mouth when he/she speaks. To enroll in a voiceprint system, he/she either says the exact words or phrases that it requires, or gives an extended sample of his/her speech so that the computer can identify the person no matter which words he/she says.

When people think of voiceprints, they often think of the wave pattern they would see on an oscilloscope. But the data used in a voiceprint is a sound spectrogram, not a waveform.

As shown in figure 2.2 a spectrogram is basically a graph that shows a sound's frequency on the vertical axis and time on the horizontal axis. Different speech sounds create different shapes within the graph.

Spectrograms also use colors or shades of grey to represent the acoustical qualities of sound.

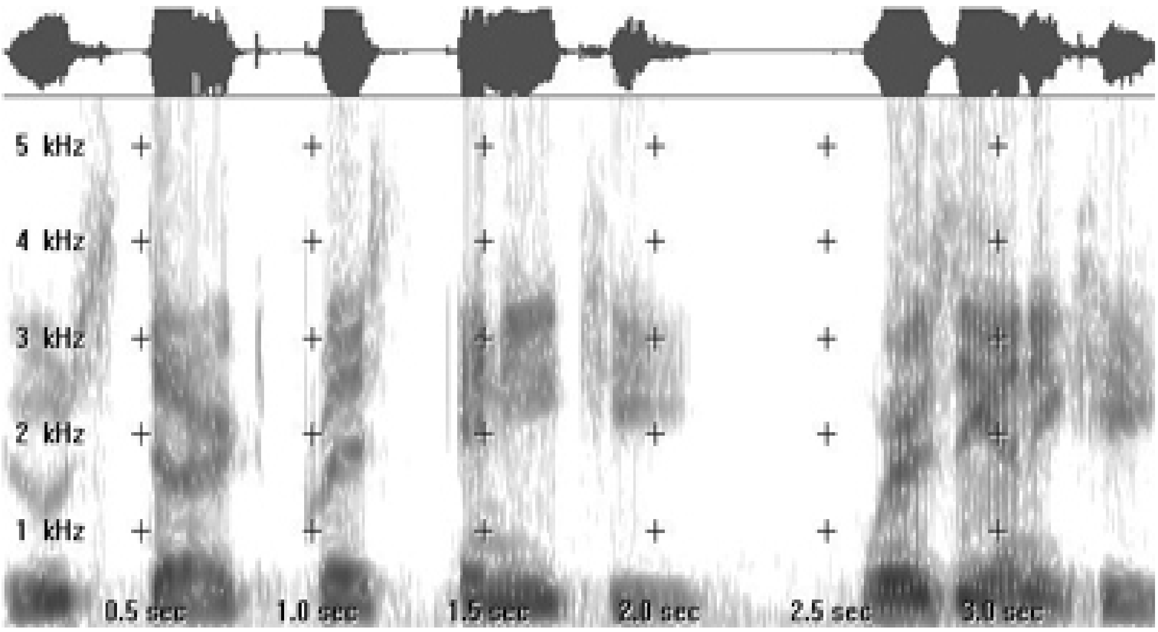

Figure 2.2 Spectrogram output

Some companies use voiceprint recognition so that people can gain access to information or give authorization without being physically present. Instead of stepping up to an iris scanner or hand geometry reader, someone can give authorization by making a phone call. Unfortunately, people can bypass some systems, particularly those that work by phone, with a simple recording of an authorized person's password. That's why some systems use several randomly-chosen voice passwords or use general voiceprints instead of prints for specific words. Others use technology that detects the artifacts created in recording and playback.

2.3. Iris and Retinal Scanning

Iris scanning can seem very futuristic, but at the heart of the system is a simple CCD digital camera. It uses both visible and near-infrared light to take a clear, high-contrast picture of a person's iris. With near-infrared light, a person's pupil is very black, making it easy for the computer to isolate the pupil and iris.

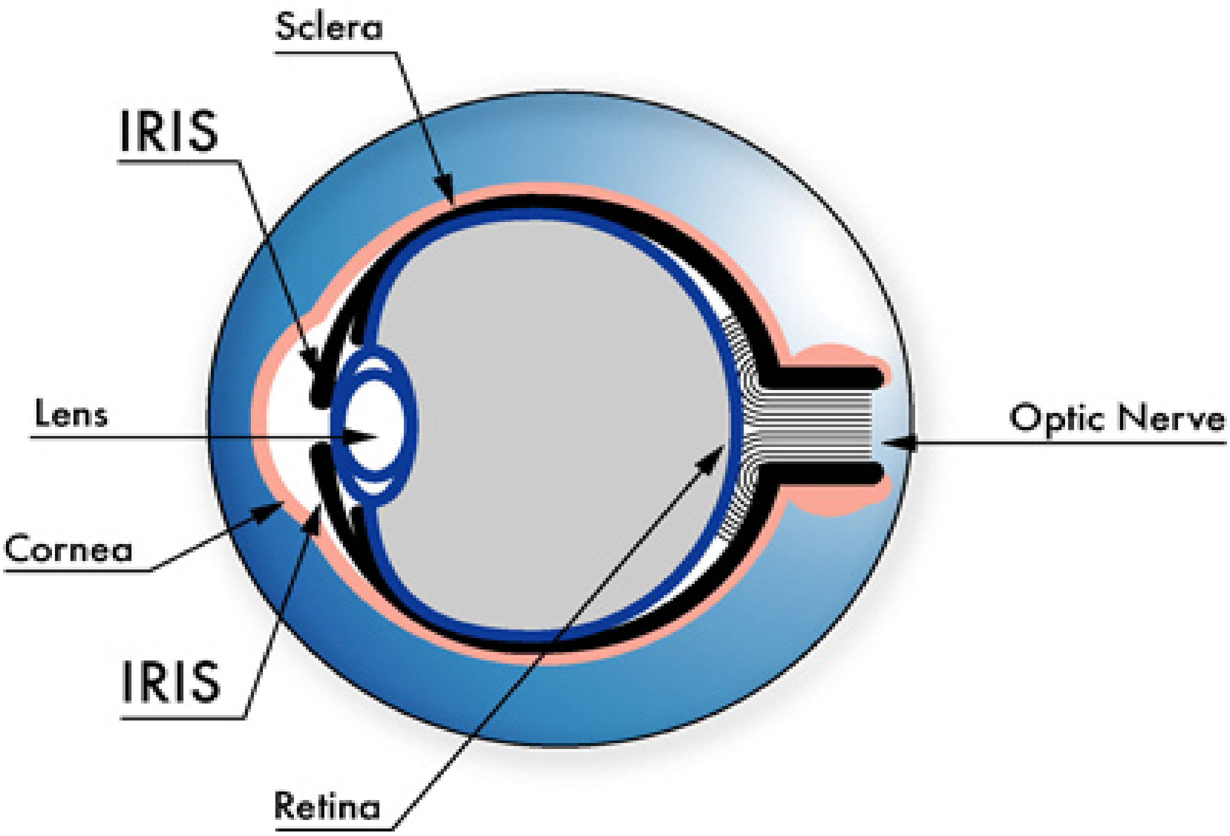

Figure 2.3 Cross-section of an eye

When a person looks into an iris scanner, either the camera focuses automatically or person use a mirror or audible feedback from the system to make sure that person are positioned correctly. Usually, his/her eye is 3 to 10 inches from the camera. When the camera takes a picture, the computer locates; the center of the pupil, the edge of the pupil, the edge of the iris and the eyelids. It then analyzes the patterns in the iris and translates them into a code.

Iris scanners are becoming more common in high-security applications because people's eyes are so unique. They also allow more than 200 points of reference for comparison, as opposed to 60 or 70 points in fingerprints.

The iris is a visible but protected structure, and it does not usually change over time, making it ideal for biometric identification. Most of the time, people's eyes also remain unchanged after eye surgery, and blind people can use iris scanners as long as their eyes have irises. Eyeglasses and contact lenses typically do not interfere or cause inaccurate readings.

Some people confuse iris scans with retinal scans. Retinal scans, however, are an older technology that requires a bright light to illuminate a person's retina. The sensor would then take a picture of the blood vessel structure in the back of the person's eye. Some people found retinal scans to be uncomfortable and invasive. People's retinas also change as they age, which could lead to inaccurate readings.

2.4. Vein Geometry

As with irises and fingerprints, a person's veins are completely unique. Twins don't have identical veins, and a person's veins differ between their left and right sides. Many veins are not visible through the skin, making them extremely difficult to counterfeit or tamper with. Their shape also changes very little as a person ages.

To use a vein recognition system, person simply places his/her finger, wrist, palm or the back of his/her hand on or near the scanner. A camera takes a digital picture using near-infrared light.

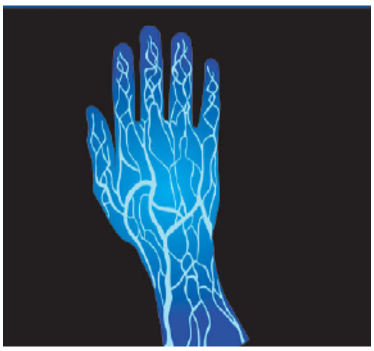

Figure 2.4 Appearance of veins at near-infrared light

The hemoglobin in his/her blood absorbs the light, so veins appear black in the picture. As with all the other biometric types, the software creates a reference template based on the shape and location of the vein structure.

Scanners that analyze vein geometry are completely different from vein scanning tests that happen in hospitals. Vein scans for medical purposes usually use radioactive particles. Biometric security scans, however, just use light that is similar to the light that comes from a remote control.

3. FACIAL RECOGNITION SYSTEMS

Automated face recognition is a relatively new concept. Developed in the 1960s, the first semi-automated system for face recognition required the administrator to locate features (such as eyes, ears, nose, and mouth) on the photographs before it calculated distances and ratios to a common reference point, which were then compared to reference data.

In the 1970s, Goldstein, Harmon, and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition[1]. The problem with both of these early solutions was that the measurements and locations were manually computed. In 1988, Kirby and Sirovich applied principle component analysis, a standard linear algebra technique, to the face recognition problem[2]. This was considered somewhat of a milestone as it showed that less than one hundred values were required to accurately code a suitably aligned and normalized face image. In 1991, M. Turk and Pentland discovered that while using the eigenfaces techniques, the residual error could be used to detect faces in images[3]. Although the approach was somewhat constrained by environmental factors, it nonetheless created significant interest in furthering development of automated face recognition technologies. The technology first captured the public's attention from the media reaction to a trial implementation at the January 2001 Super Bowl, which captured surveillance images and compared them to a database of digital mugshots. This demonstration initiated much-needed analysis on how to use the technology to support national needs while being considerate of the public's social and privacy concerns.

Face recognition is a successful application of image analysis and understanding. It also has an important place in computer vision applications. Today, face recognition technology is being used to combat passport fraud, support law enforcement, identify missing children, and minimize benefit/identity fraud.

Face recognition systems uses different methods for identification. In this project eigenface method is used.

3.1. Eigenface Method

If we think about a grayscale image with a dimension of *N* by *N*, it describes a vector dimension of N^2. This means it represents a point in N^2 dimensional space. Images of faces will not be randomly distributed in this huge space. They can be described by a relatively low dimensional subspace [3].

Main objective of this method is to find the principal components of the distribution of faces or the eigenvectors of the covariance matrix of the set of face images. Each face of the training set can be represented exactly in terms of a linear combination of the eigenfaces. The number of possible eigenfaces is equal to the number of the people in the training set.

3.2. Mathematical Explanation

Preparing the data, the faces constituting the training set (Γ_i) should be prepared for processing $\mathbf{\Gamma_1, \Gamma_2, .., \Gamma_M}$

The average of the training set will be;

$$\mathbf{\Psi} = \frac{1}{M} \left(\sum_{n=1}^{M} \mathbf{\Gamma_n} \right) \tag{3.1}$$

Each face differs from the average by the vector,

$$\mathbf{\Phi_i} = \mathbf{\Gamma_i} - \mathbf{\Psi} \tag{3.2}$$

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors $\mathbf{u_n}$, which best describes the distribution of the data. The k^{th} vector, $\mathbf{u_k}$, is chosen such that

$$\lambda_k = \frac{1}{M} \left(\sum_{n=1}^{M} \left(\mathbf{u_k}^T \mathbf{\Phi_n} \right)^2 \right) \tag{3.3}$$

The vectors $\mathbf{u_k}$ and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix;

$$\mathbf{C} = \frac{1}{M} \left(\sum_{n=1}^{M} \mathbf{\Phi_n} \mathbf{\Phi_n}^T \right) = \mathbf{A}\mathbf{A}^T \tag{3.4}$$

$$\text{where; } \mathbf{A} = [\Phi_1 \, \Phi_2 \ldots \Phi_M] \tag{3.5}$$

The covariance matrix has a dimensionality of N^2 by N^2, so one would have N^2 eigenfaces and eigenvalues. For a 256 by 256 image that means that one must compute a 65,536 × 65,536 matrix and calculate 65,536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task.

Because of this reason we determine an M by M matrix **L**, **v** are M eigenvectors of **L** and **u** describes eigenfaces. The advantage of this method is that one has to evaluate only M numbers and not N^2. Usually, $M < N^2$ as only a few principal components (eigenfaces) will be relevant. The amount of calculations to be performed is reduced from the number of pixels ($N^2 \times N^2$) to the number of images in the training set (M) with a mathematical trick. [9].

$$\mathbf{L} = \mathbf{A}^T \mathbf{A} \tag{3.6}$$

Consider eigenvectors(**v**) and eigenvalues(μ) of the L matrix

$$\mathbf{A}^T \mathbf{A} . \mathbf{v_i} = \mu_i \, \mathbf{v}_i \tag{3.7}$$

Both sides are multiplicated by **A**,

$$\mathbf{AA^TA.v_i} = \mu_i \,\mathbf{Av_i} \quad (3.8)$$

Thus,

$$\mathbf{C.A.v_i} = \mu_i \,\mathbf{Av_i} \quad (3.9)$$

Where $\mathbf{v_i}$ is the eigenvector of **L** and $\mathbf{Av_i}$ is the eigenvector of **C** and

$$\mathbf{L_{mn}} = \mathbf{\Phi_m}^T \mathbf{\Phi_n} \quad (3.10)$$

After the calculation of the eigenvectors of **L**, eigenfaces(eigenvectors of **C**) can be calculated by this formula,

$$\mathbf{u_i} = \sum_{k=1}^{M} \mathbf{v_{ik}} \,\mathbf{\Phi_k} \quad (3.11)$$

$$\mathbf{i}=1,\ldots..,M \quad (3.12)$$

Now we will project the face images to the face spaces. This is essentially dot product of each mean subtracted face image with one of the eigenfaces. These weights (w) form the feature vector ($\mathbf{\Omega}$). Simply the same procedure will be applied to the new image which will be compared with the saved images.

$$W_k = \mathbf{u_k}^T(\,\mathbf{\Gamma} - \mathbf{\Psi}) \quad (3.13)$$

where, $k=1,\ldots M$

$$\mathbf{\Omega}^T = [w_1\ w_2\ \ldots.w_M] \quad (3.14)$$

Classification is performed by comparing the feature vectors of the face library members with the feature vector of the input face image. This comparison is based on the Euclidean distance "**ε**" between the two members to be smaller than a user defined threshold. If the comparison falls within the user defined threshold, then face image is classified as "known", otherwise it is classified "unknown" and can be added to face library with its feauture vector for later use, thus making the system learning to recognize new face images.

$$\boldsymbol{\varepsilon} = \|(\boldsymbol{\Omega}_{new} - \boldsymbol{\Omega}_k)\| \tag{3.15}$$

3.3. Other Types of Facial Recognition System

3.3.1. Feature Based (Geometric)

In feature-based approaches, geometric features, such as position and width of eyes, nose and mouth, eyebrow's thickness and arches, face breadth are extracted to represent a face. Feature face approaches allow for smaller memory requirements and a higher recognition speed than template based approaches, and they are particularly useful for face scale normalization and 3-D head model based pose estimation. However, perfect extraction of features is shown to be difficult in implementation.

3.3.2. Template Based (Photometric)

The simplest template matching represents a whole face using a single template, which is usually an edge map of the original face image. In a more complex way of template matching multiple templates may be used for each face to account for recognition from different viewpoints. Another important variation is to employ a set of smaller facial feature templates, corresponding to eyes, nose, and mouth for a single viewpoint. The most attractive advantage of temple matching is its simplicity; however it suffers from large memory requirements and an inefficient matching algorithm.

3.3.3. Appearance Based

The idea of this approach is to project face images into a linear subspace with low dimensions. The first version of such a subspace is the eigenface space constructed by the principal component analysis from a set of training images. Later the concept of eigenfaces were extended to eigenfeatures such as eigeneyes, eigenmouth etc. for the detection of facial features. More recently, fisher face space and illumination subspace have been proposed for dealing with recognition under varying illumination.

3.4. Advantages and Disadvantages

The primary advantage of the eigenface method is the system's speed and easiness. The eigenface approach reduces the amount of data needed to identify an individual to 1/1000th of a full sized image [5].

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.

4. A FACE RECOGNITION SYSTEM USING EIGENFACE METHOD

In this project eigenface approach is used to design a security system. According to this, Ultimate Face Recognition software is written and the corresponding hardware is designed. Ultimate Face recognition software is designed by MATLAB 7.0.1 R14 language.

4.1. Software

4.1.1. Starting UFR

First you have to open the Matlab software and select the main directory.

Figure 4.1 Selecting the main directory of Matlab

Now type ‘START’ at the command window to open the UFR. Then a bar that contains a redline which moves on it can be seen.

Figure 4.2 Bar that shows the software is loading

Then the main menu of the UFR can be seen.

```
                         <<MAIN MENU>>

.........Select Image...................[1]
.........New Record.....................[2]
.........Number of ID(s)................[3]
.........Face Recognition...............[4]
.........Delete Database................[5]
.........ID Information.................[6]
.........Mean Face and EigenFaces.......[7]
*******************************************
Quit....................................[Q]
Reset...................................[R]

Your Choice-->
```

Figure 4.3 Main menu

4.1.2. Selecting Image

To select and image, Type '1'. A window appears then you can select an image. Jpeg format is supported.

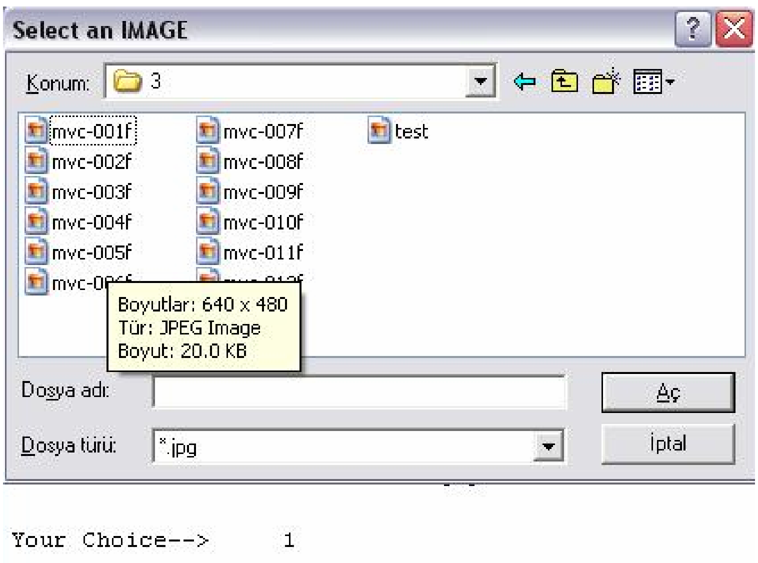

Figure 4.4 File selection window

Then the selected image will be shown to you.

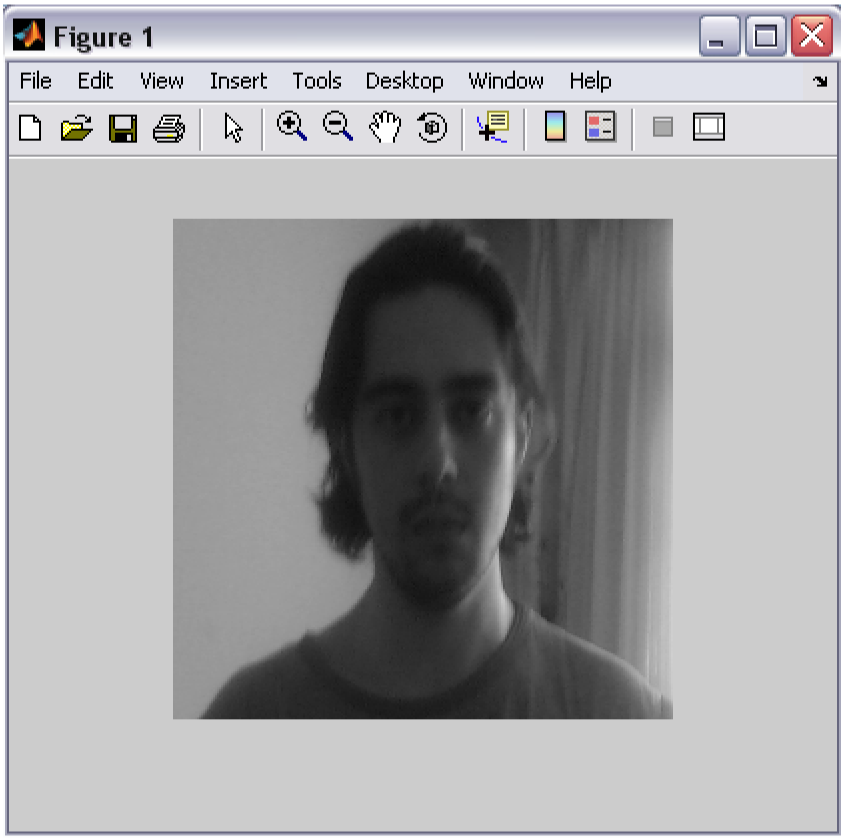

Figure 4.5 Selected image

The software makes some operations to the images. Face recognition can be made if all of the images have the same size. First the image is converted to grayscale by eliminating the hue (color) and saturation information while retaining the luminance. After this, the size of the image will be changed by down sampling. The arranged size of the software is 250 by 250.

4.1.3 New Record

First you have to select an image. Then type “2” if you want to save this image to the database. The software will automatically give an ID number to your face. If the database is empty, the first face will have an ID number of 1. If you add a second image then its ID number will be 2 and so on. Then the software wants you to enter the information about the person. This information will be stored to other databases. One of these databases is called IDnumber.DAT. For the first image it is “1.DAT”. The name of the second database is same as the name of the person. Here it is “Bedirhan.DAT”. Contents of these two databases are same. The first one is used by the software automatically for giving the information about the matched person the recognition procedure. For example if the matched person’s ID is 3 then it opens “3.DAT” and shows the recorded information to the user. The second one (name.DAT) will be explained in ID information part.

After recording main information, UFR asks if the person is allowed or not for access. The corresponding led will turn on according to typing “y” or “n”. It will be explained in hardware section.

Last a reference image must be selected for full information of the person. The reference image will be shown when the recognition is done. Selecting a reference image is recommended because you may not remember all of the names. Thus you can easily understand if this is the correct person or not. If you don’t want to add reference then push “cancel” button. The other data will be saved.

The corresponding menu is like that;

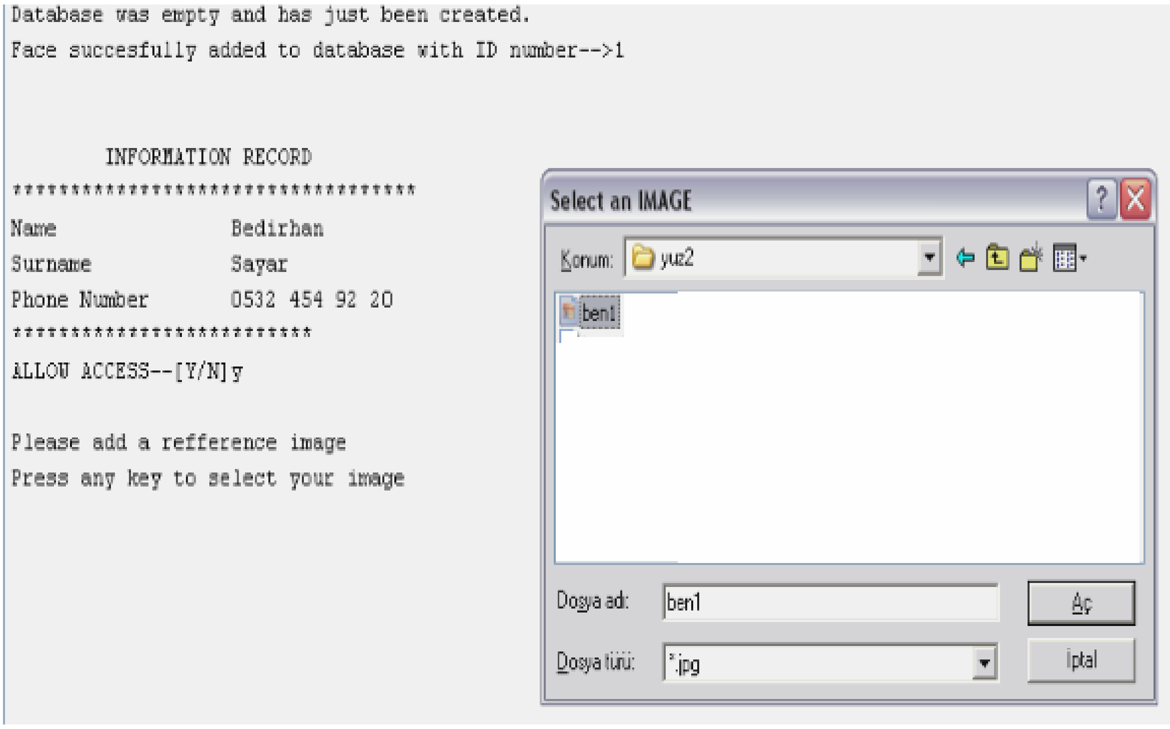

Figure 4.6 New record menu

4.1.4 ID Information

By selecting 3 you can see how many IDs are recorded to the database.

```
There are -->3 ID(s) at the database.
press any key to continue
```

Figure 4.7 Displaying the number of IDs saved inside the database

If it is empty than you will see “database is empty” message

4.1.5. Face Recognition

First of all you have to select an image that you want to compare with the saved images. Then type “4” for the face recognition. The software checks the database and the existence of the new image, and then tells the ID number of the matched face, turns on the corresponding leds according to the answer of “allow access” then displays the information of this person.

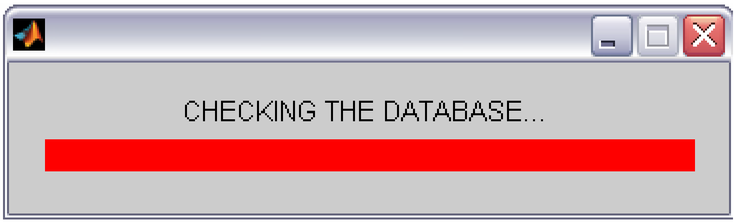

Figure 4.8 Bar showing the image check operation

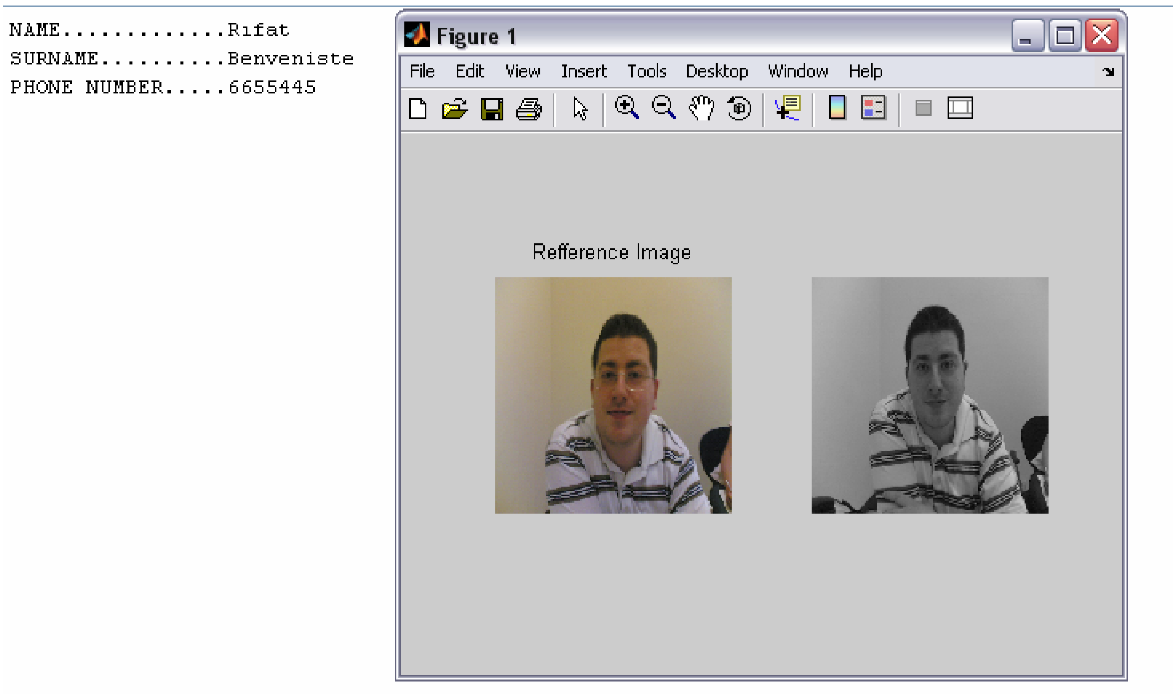

Figure 4.9 Information about the matched person

If the new face does not exist at the database then, the software warns you about it but shows the similar person stored inside the database.

4.1.6. Deleting the Databases

Type “5” at the main menu then the program asks you to delete databases or not. If you type “y” they will be deleted.

4.1.7. Personal Information

The ID information menu appears if you type “6” at the main menu.

```
ID NUMBER........-->3
NAME.............Bedirhan
SURNAME..........Sayar
PHONE NUMBER.....0532 454 92 20
ACCESS PERMISION.-->Y
Refference Image NOT FOUND!!
```

Figure 4.10 ID information menu

If you want to display the information about the person, then type”1”. The software asks you the name or ID number of the person. Typing the name or ID number is case sensitive. The computer will show the information if it exists inside the database.

```
NAME.............Bedirhan
SURNAME..........Sayar
PHONE NUMBER.....0532 454 92 20
ACCESS PERMISION.-->y
Refference Image NOT FOUND!!
```

Figure 4.11 Information about a recorded person

The picture of the person will be shown if you have added a reference image to this person when adding him/her to the database.

4.1.8. Mean Face and Eigenfaces

Typing “7” at the main menu will show the eigenfaces and the average of faces recorded. There must be at least 2 faces inside the database. Sample eigenfaces are shown below.

Figure 4.12 Eigenfaces obtained with UFR

The corresponding average is shown like,

Figure 4.13 Average face obtained with UFR

4.1.9. Possible Errors

The UFR is designed with feedbacks and corrections for all of the menus included in the system. Thus, if the user pushes a wrong key, the software warns him/her, and so that a correct selection can be done. This means the software never ends or crashes unless the user wants. If you want to end the software, type “q” or “Q” from the main menu. Also you can reset the software with typing “r” or “R”. This operation closes all the pictures opened and deletes the variables stored in the memory.

4.2. Experimental Results

For testing the performance of the proposed system, two image sets are used. The images at the first set are captured by using the camera of Nokia 6630 mobile phone with a pixel resolution of 1280 by 960. Other images belonging to the second one are partially taken from the Indian face database, downloaded from the internet [10]. Here the image resolution is 640 by 480. Both of the sets are in JPEG format.

First of all, images at the first set are used. The performance of the algortihm is observed according to these images. Then all of the information is removed. Same procedure is applied to the second one. The threshold value is arranged according to the properties of the images.

Inside the first image set, 26 images belonging to 14 people are used. 12 of them are used as training set. They are saved in the database. The others are compared with the training set.

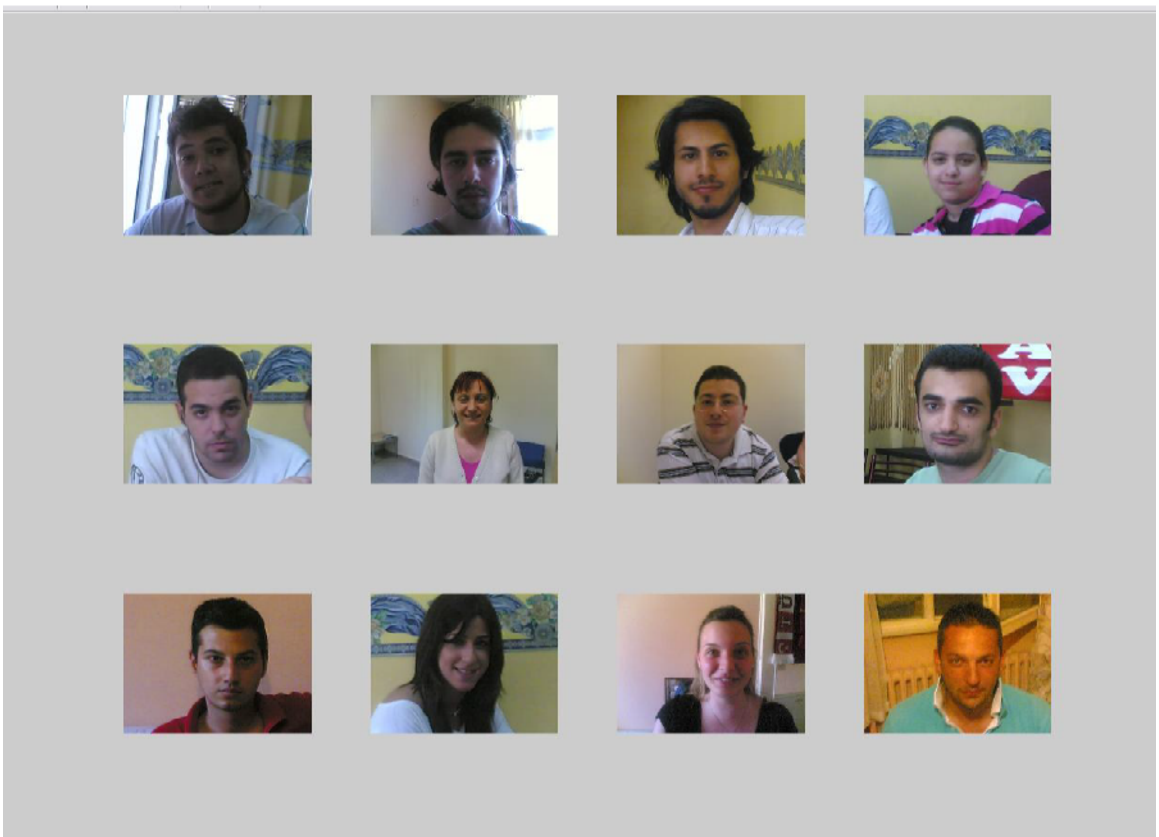

Figure 4.14 Faces of the first set saved to the database

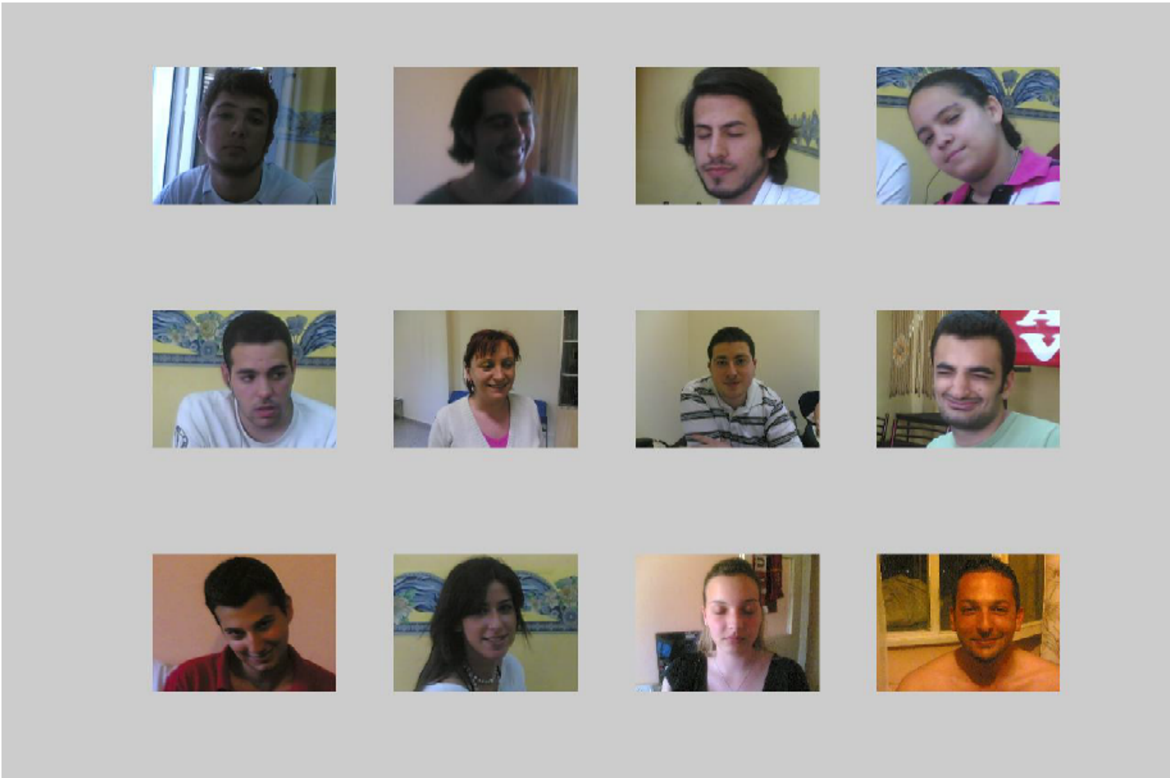

Figure 4.15 Faces of the first set not used in the database

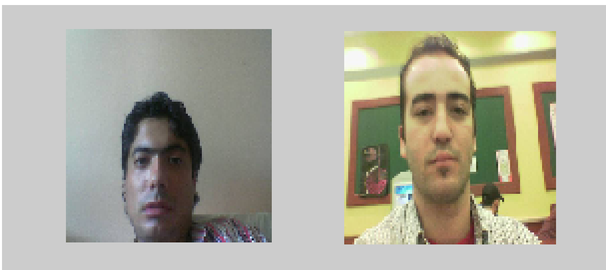

Figure 4.16 Faces of the first set not saved to the database

We expect that, all of the saved images match with their corresponding image and unsaved images don’t appear inside the database. However the performance is not perfect. Below we can see which of the images are matched and which are not. The algorithm is tested using 14 images that are not in the data base. We observed that 11 out of 14 are classsified correctly. The other 3 are matched to the closest face. Thus the success of the system on this data base is about 79 percent.

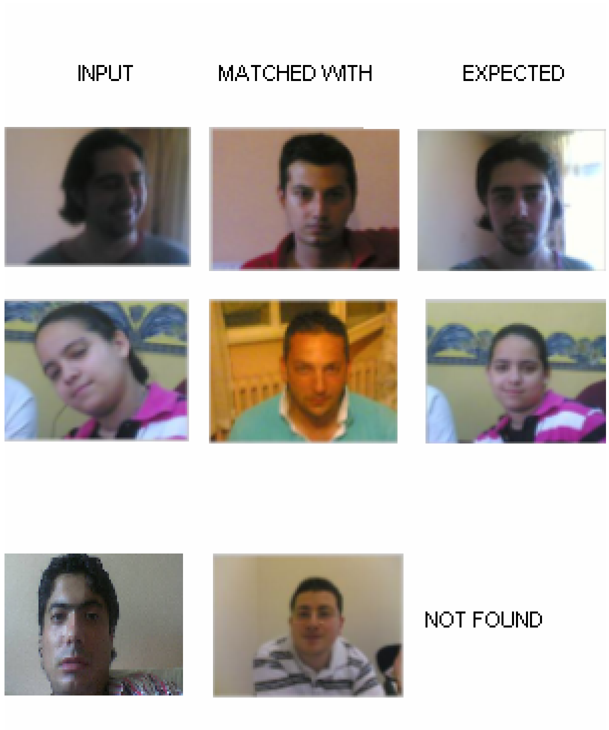

Figure 4.17 Faces of the first set recognized as wrong

For the second image set, again 26 images belonging to 14 people are used. 12 of them are used as training set. They are saved to the database. The others are compared with the training set. According to these images the recognition performance of the algortihm is better than the first image set. Here only one out of 14 faces is matched to a different person. This means the efficiency is 93 percent.

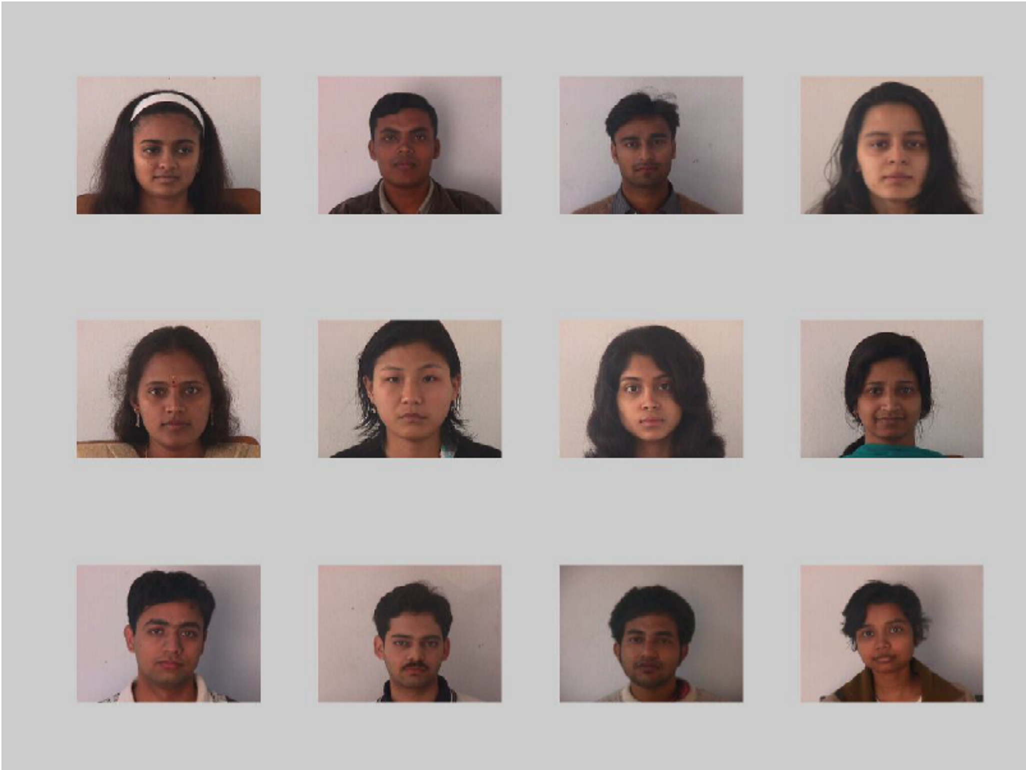

Figure 4.18 Faces of the second set saved to the database

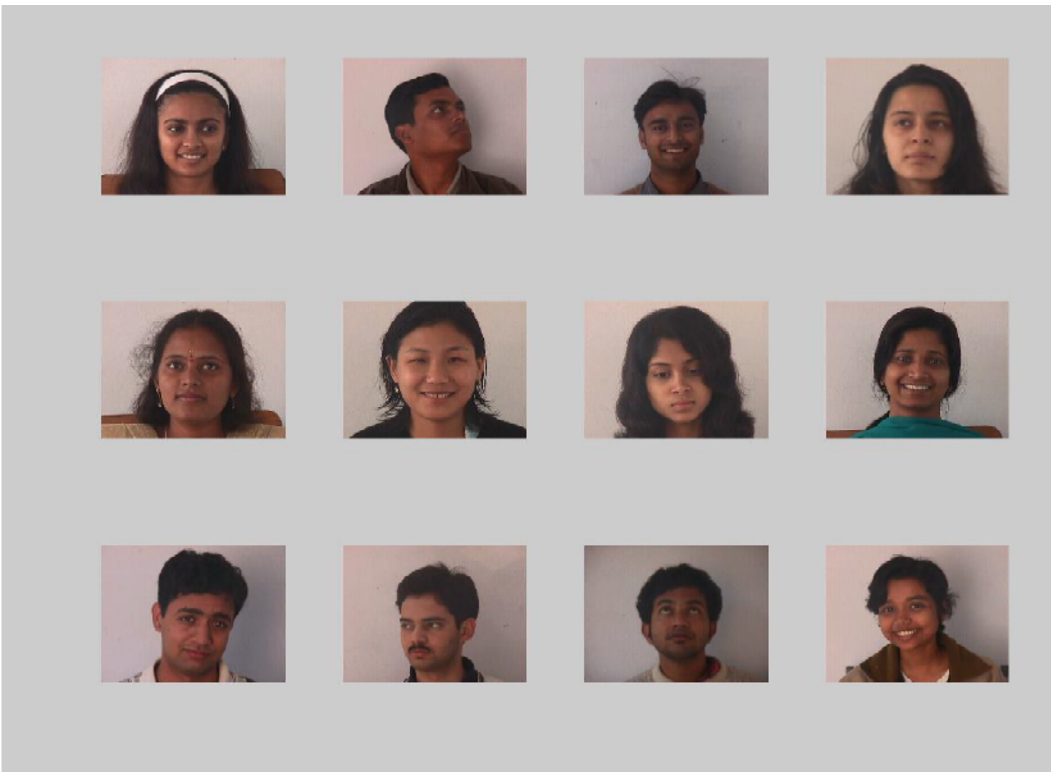

Figure 4.19 Faces of the second set compared with the database

Figure 4.20 Faces of the second set not saved to the database

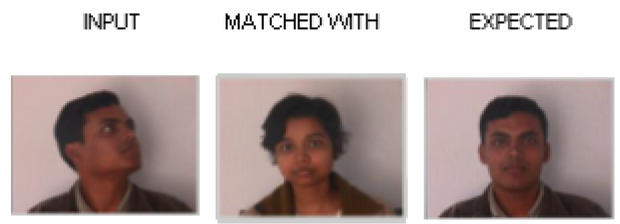

Figure 4.21 False recognized face of the second set

According to these values we can easily say that the true recognition rate is not a hundred percent. If we look at the differences between the two sets, it can be easily said that positioning and lighting seriously affects the system’s performance. There is a 14 percent of success difference between the usages of these two sets. It is an important value for a security system.

4.3 Flowchart of the algorithm

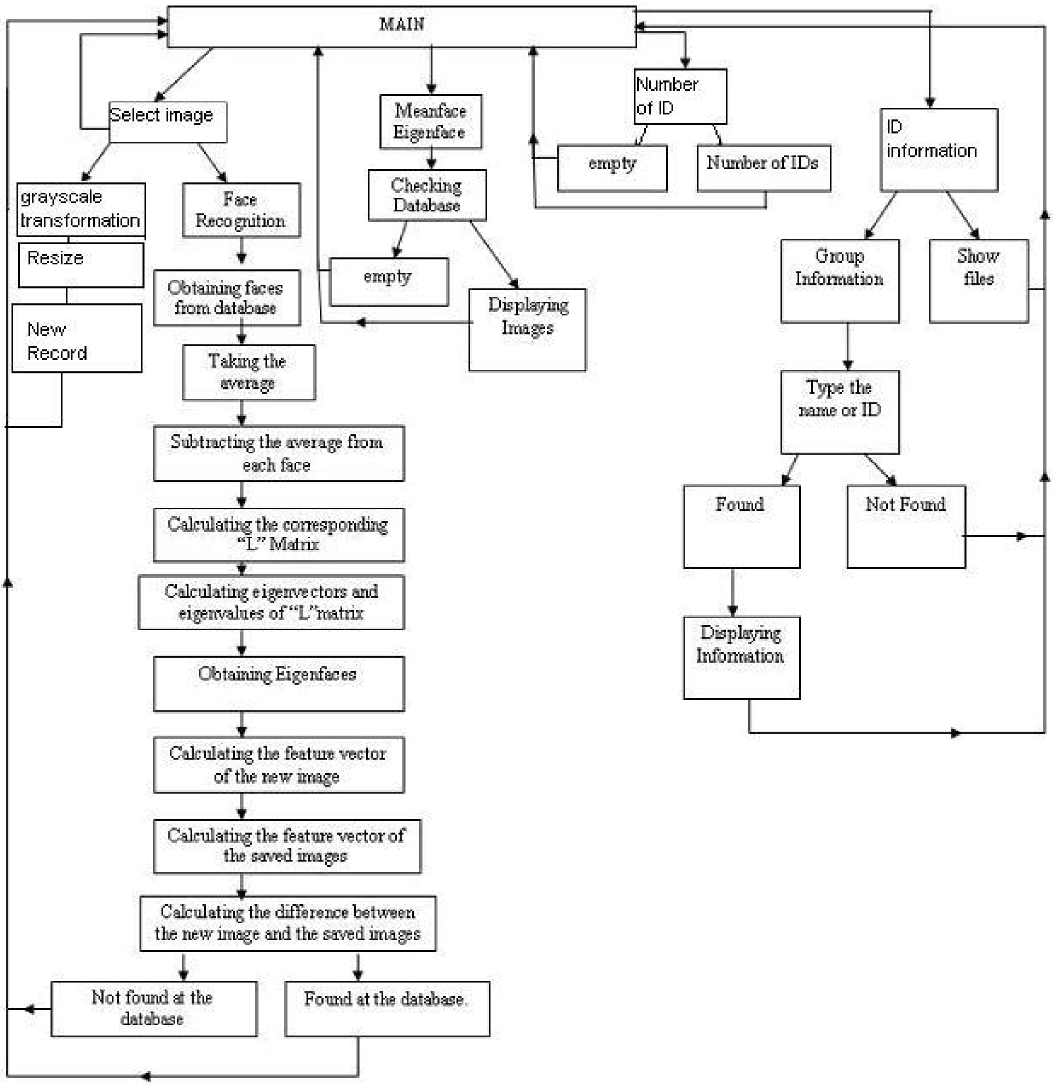

Figure 4.22 Flowchart of the algorithm

4.4. Hardware

The system has a very simple hardware. There are 3 leds, 1 switch, 1box and a parallel port cable.

The parallel port is programmed to control the circuit. Here we can see the port structure and corresponding pin diagrams.

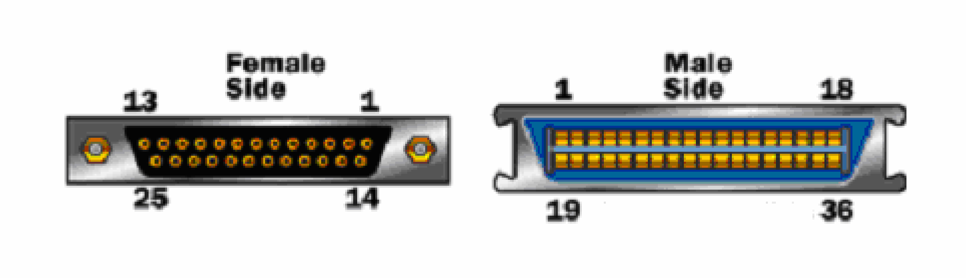

Figure 4.23 Female and male side of the parallel port

PIN	SIGNAL
1	Strobe
2	Data0
3	Data1
4	Data2
5	Data3
6	Data4
7	Data5
8	Data6
9	Data7
10	Acknow.
11	Busy
12	Paper End
13	Select
14	Auto Feed
15	Error
16	Init
17	Select In
18	GND
19	GND
20	GND
21	GND
22	GND
23	GND
24	GND
25	GND

Figure 4.24 Pin diagram of the parallel port

In this project, leds are connected to data0, data1 and data2. The circuit is grounded by the pin numbered 18.

Corresponding figure of the circuit is,

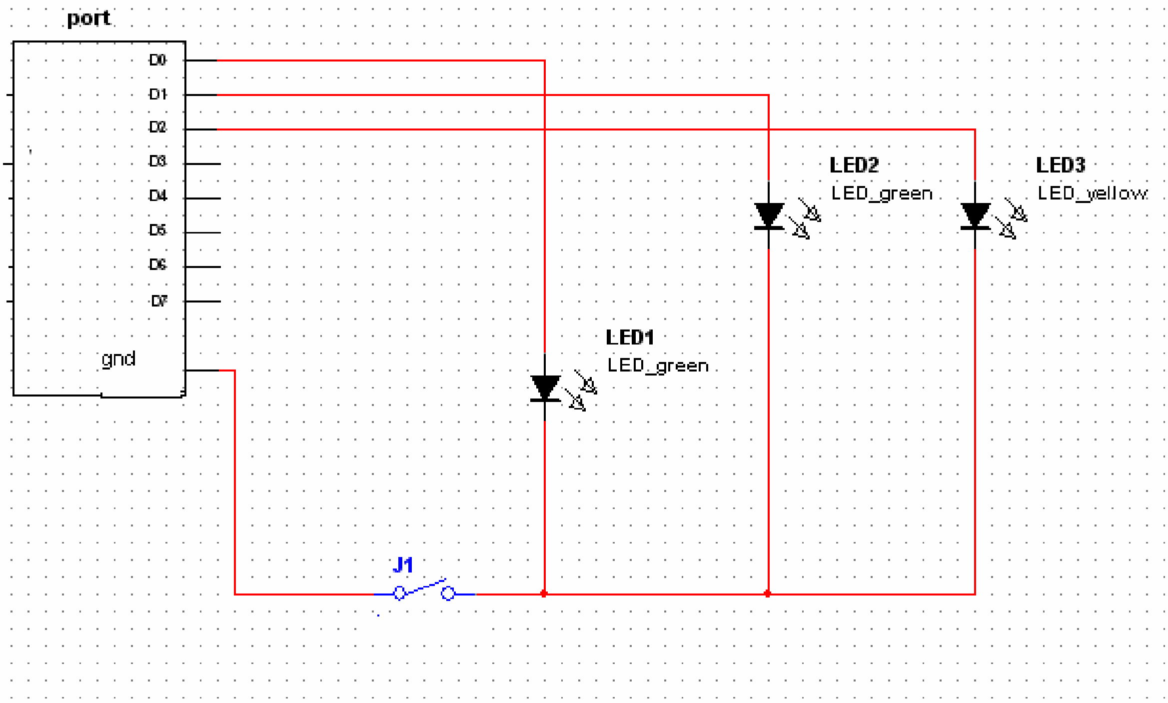

Figure 4.25 Circuit schematic

When you start the program, the switch will activate the system. The switch is turned on at the beginning because the computer sends logical 1 to each port. This hardware can be thought as a security system. For example think Led 2 as an electronic door switch and Led 3 as an alarm.

When the program starts, only the green led (led1) is on. This means the system is working and closed to access. During the recording procedure you can give permission to access. If a person has permission (typing “y” or “Y” will give permission), after the face recognition, the other green led (led2) lights for a few seconds. Then led 1 turns on again. If the person has not permission (typing “n” or “N”) then the yellow led (led 3) turns on for a few seconds.

5. CONCLUSION

Face recognition method represents a face by projecting original images into a low-dimensional linear subspace defined by eigenfaces. A new face is compared to known faces by computing the distance between their projections in the subspace.

To implement such a system, The Ultimate Face Recognition (UFR) software is developed by using the MATLAB 7.0.1 computing environment. The UFR also simulates security operations beside the face recognition. The UFR is designed with feedbacks and corrections for all of the menus included in the system. Thus, if the user pushes a wrong key, the software warns him/her, so that a correct selection can be done.

The software creates a database for the face groups. The images of the faces are saved in this database. If the user wants to make recognition, a new face is compared with the faces inside the database. Also the properties of the person can be added (first name, last name, telephone number, etc…) by saving them to another database. When creating this database, the software asks first name, surname, and phone number. It also asks for a reference image that will be shown after the recognition and accessibility of the person. According to the answer, software controls the corresponding hardware that is connected to the computer through the parallel port.

If we investigate the performance of the software about the face recognition, we can easily say that true recognition rate is not hundred percent (actually very close.) In one data set, the true recognition rate is obtained to be 79 % whereas for another face set, the recognition rate is increased to 93 %. In face recognition applications, like the other image processing applications, lighting and positioning is very important. In addition to these, for the face recognition, physical characteristics (hair type, beard, mustache, etc...) are also very important. According to the true recognition rate this system is not suitable for the systems that need full security. For example, twins can both access to a system secured by face recognition technique. Face recognition systems are suitable for criminal records. If we want to design a full security system, iris or fingerprint scanners and vein scanners can be combined with the face recognition system designed here to increase the correct acceptance rate.

Finally, in this thesis project, different biometric technologies are investigated and a facial recognition system is designed to be used as a security device. Correct face matching performance of the system according to the captured images is observed to be about 79 %.

APPENDIX A: CODES USED IN THE SIMULATION

START.M

```
% Main code for accessing the software
%Clears the parallel port first and then turns on(gives logical 1)
%to D2


clc
clear all
close all
b=1;
for b=1:100
i=1;
k=1;
parport=digitalio('parallel','LPT1');
line=addline(parport,0:3,'out');
putvalue(parport,[0 0 1 0]);
end


putvalue(parport,[0 0 1 0]);
  gval = getvalue(parport);
  delete(parport);

h = waitbar(0,'Ultimate Face Recognition version 1.0.. LOADING.....');
for i=1:2450,
waitbar(i/100)
end
close(h)
bdrfacerec
```

BDRFACEREC.M

```
%MAIN MENU OF THE SOFTWARE
%According to the users request calls corresponding script(s)
clc
disp('                    <<MAIN MENU>>       ')
disp(' ')

disp('.........Select Image....................[1]')
disp('.........New Record......................[2]')
disp('.........Number of ID(s).................[3]')
disp('.........Face Recognition................[4]')
disp('.........Delete Database.................[5]')
disp('.........ID Information..................[6]')
disp('.........Mean Face and EigenFaces........[7]')
disp('*******************************************')
disp('Quit.....................................[Q]')
disp('Reset....................................[R]')
disp('                                  ')
```

```
y=input('Your Choice-->     ','s');


switch(y)
    case '1'
        [ans,pathname]=uigetfile( ...
{'*.jpg';'*.jpeg'...

   }, ...
   'Select an IMAGE');


if isequal(ans,0);
disp('ACTION CANCELLED');
h = waitbar(0,'PLEASE WAIT...');
for i=1:2450,
waitbar(i/100)

end
close(h)
clear all
close all
bdrfacerec
return

end
 try
img=imread([pathname ans]);
img=rgb2gray(img);
img=imresize(img,[250 250]);
figure,imshow(img)
bdrfacerec


catch
clc
disp('INCORRECT FILE FORMAT')
disp(' ')
disp('Possibilities:')
disp(' ')
disp('1-Selected file is not an image file')
disp('2-Selected image is grayscale')
disp('')
disp('Press any key to continue...')
pause
bdrfacerec
 return
 end


   case '2'
        checkdata

    case'3'
        datainfo
    case'4'
```

```
            facerec

        case'5'
            deldata

        case'Q'
            clc
            clear all
            close all
            warndlg('THANKS FOR USING UFR v1.0',' ');
     case'q'
            clc
            clear all
            close all
            warndlg('THANKS FOR USING UFR v1.0',' ');


        case 'r'
            clc
            clear all
            close all
            bdrfacerec


           case 'R'
            clc
            clear all
            close all
            bdrfacerec

        case '6'
            clc
            ginfo

        case '7'
            eigen


        otherwise
    clc
            disp('Wrong SELECTION!!')
            disp(' ')
            disp('Press any key to continue...')
       pause
            bdrfacerec
            return
    end
```

CHECKDATA.M

```
%Checking the image before adding to database.
%Checks if the image is selected or not
clc
h = waitbar(0,'CHECKING.....');
for i=1:1450,
waitbar(i/100)
```

```
end
close(h)
if exist('img')
   dataaddtest
else
   h = waitbar(0,'No image has been selected Please Wait.....');
for i=1:3450,
waitbar(i/100)

end
close(h)


    bdrfacerec

end
```

DATAADDTEST.M

```
%Stores the selected image to the database
%Stores the information about the face

clc
if exist('img')

    if(exist('fdata.dat')==2)
        load('fdata.dat','-mat');
         fnumber=fnumber+1;
         max_class=max_class+1;
         class_number=max_class;
         data{fnumber,1}=img(:);


                 data{fnumber,2}=class_number;
                 save('fdata.dat','data','fnumber','max_class','-append');

                clc
                 disp(strcat('Face succesfully added to database with ID
number --> ',num2str(class_number)));

                disp(' ')
                disp(' ')


                disp('        INFORMATION RECORD')
                disp('***********************************')
                 a=input('Name                   ','s');
                 b=input('Surname                ','s');
                 c=input('Phone Number           ','s');
                 op=(strcat(num2str(class_number)));
                 disp('*************************');
                 h=input('ALLOW ACCESS--[Y/N]','s');
                 switch(h)
                 case 'y'
                 case 'Y'
                 case 'N'
                 case 'n'
                 otherwise
```

```
                disp('WRONG KEY!! NO ACCESS')
                end
                disp(' ');
disp('Please add a refference image ')
disp('Press any key to select your image')
pause

     [ans,pathname]=uigetfile( ...
{'*.jpg';'*.jpeg'...

   }, ...
   'Select an IMAGE');


if isequal(ans,0)

y=num2str(class_number);

save([y '.dat'],'a','b','c','h','op')
save([a '.dat'],'a','b','c','h','op')
clc
disp('INFORMATION IS SAVED!!!')
disp('Press any key to continue')
pause
bdrfacerec
return
end

try

kl=imread([pathname ans]);
kl=imresize(kl,[250 250]);
imshow(kl)
y=num2str(class_number);
save([y '.dat'],'a','b','c','h','kl','op')
save([a '.dat'],'a','b','c','h','kl','op')
close all
clc
disp('INFORMATION IS SAVED!!!')
disp('Press any key to continue')
pause
bdrfacerec


   catch

clc
y=num2str(class_number);
save([y '.dat'],'a','b','c','h')
save([a '.dat'],'a','b','c','h')
disp('UNKNOWN FORMAT!! The image cannot be loaded')
disp('Other variables are saved  ')
disp('Press any key to continue...')
pause
clear all
bdrfacerec
return
   end
```

```
     else
       clc
        fnumber=1;
        class_number=1;
        data{fnumber,1}=img(:);

                max_class=1;
                data{fnumber,2}=class_number;
                save('fdata.dat','data','fnumber','max_class');

                clc
                disp('Database was empty and has just been created.')
                disp(strcat('Face succesfully added to database with ID
number-->',num2str(class_number)));
               disp(' ')
               disp(' ')
               disp('        INFORMATION RECORD')
               disp('***********************************')
                a=input('Name               ','s');
                b=input('Surname            ','s');
                c=input('Phone Number       ','s');
                op=(strcat(num2str(class_number)));
                disp('***************************');
                h=input('ALLOW ACCESS--[Y/N]','s');
               switch(h)
                 case 'y'
                 case 'Y'
                 case 'N'
                 case 'n'
                 otherwise
                  disp('WRONG KEY!! NO ACCESS')
                  end
                 disp(' ');
 disp('Please add a refference image ')
 disp('Press any key to select your image')
 pause

      [ans,pathname]=uigetfile( ...
 {'*.jpg';'*.jpeg'...

    }, ...
    'Select an IMAGE');


  if isequal(ans,0)
 disp('ACTION CANCELLED');
 y=num2str(class_number);

 save([y '.dat'],'a','b','c','h','op')
 save([a '.dat'],'a','b','c','h','op')
 clc
```

```
 disp('INFORMATION IS SAVED!!!')
 disp('Press any key to continue')
 pause
 bdrfacerec
  return
  end

     try

 kl=imread([pathname ans]);

 imshow(kl)
 y=num2str(class_number);
 save([y '.dat'],'a','b','c','h','kl','op')
 save([a '.dat'],'a','b','c','h','kl','op')
 close all
 clc
 disp('INFORMATION SAVED!!!')
 disp('Press any key to continue')
 pause
 bdrfacerec


    catch

 clc
 y=num2str(class_number);

 save([y '.dat'],'a','b','c','h','op')
 save([a '.dat'],'a','b','c','h','op')

 disp('UNKNOWN IMAGE FORMAT!! The face cannot be loaded')
 disp('Other variables are saved  ')
 disp('Press any key to continue...')
 pause
 clear all
 bdrfacerec
return
    end
    end


end
```

DATAINFO.M

```
% Displays how many faces stored in the facedatabase
clc
close all
clear('img');
if(exist('fdata.dat')==2)
    load('fdata.dat','-mat');

    disp(strcat('There are -->'   ,num2str(fnumber),' ID(s) at the
database.'));
```

```
disp('press any key to continue')
pause
bdrfacerec
else
    disp('Database is empty');
    disp('press any key to continue')
    pause
    bdrfacerec

end
```

DELDATA.M

```
%Deletes all of the databases recorded.
clc
close all


if(exist('fdata.dat')==2)
    a=input('Delete all information---->[Y/N]','s');
    switch (a)
        case 'y'

          delete('*.dat');
        disp('Database REMOVED')
        disp('Press any key to continue')
        pause
        bdrfacerec

        return
          case 'Y'

          delete('*.dat');
        disp('Database REMOVED')
        disp('Press any key to continue')
        pause
        bdrfacerec

        case 'N'
            bdrfacerec

        case 'n'
            bdrfacerec
        otherwise
            clc
        disp('Wrong SELECTION!!')
        disp(' ')
        disp('Press any key to continue...')
   pause
deldata

    end

else
         clc
        disp('Database is already empty')
```

```
        disp(' ')
        disp('Press any key to continue...')
   pause
bdrfacerec
end
```

DINFO.M

```
%displays the information about people MANUALLY
%ldinfo.m displays the same information after the recognitiopn
automaticly
clc

clear all
jk=input('Please Type the ID Number or Name of the person-->','s');

if(exist([jk '.dat']))
    load ([jk '.dat'],'-mat');

  if exist('kl')
try
imshow(kl)
disp (strcat('ID NUMBER........-->',num2str(op)))
disp (strcat('NAME.............',num2str(a)))
disp (strcat('SURNAME..........',num2str(b)))
disp (strcat('PHONE NUMBER.....',num2str(c)))
disp (strcat('ACCESS PERMISION.-->',num2str(h)))
pause
ginfo
catch
clc
    disp('')
    disp('An error occured while reading the database')
    disp('Please check your variables and try again')
    pause
    ginfo
return
end

  else

try
clc
disp (strcat('ID NUMBER........-->',num2str(op)))
disp (strcat('NAME.............',num2str(a)))
disp (strcat('SURNAME..........',num2str(b)))
disp (strcat('PHONE NUMBER.....',num2str(c)))
disp (strcat('ACCESS PERMISION.-->',num2str(h)))
disp('Refference Image NOT FOUND!!')
pause
ginfo

catch
   clc
    disp('')
    disp('An error occured while reading the database')
    disp('Please check your variables and try again')
    pause
```

```
    ginfo
    return
end
  end
else
    disp('THERE IS NO RECORD ABOUT THE PERSON')
    pause
   ginfo
return
  end
```

EIGEN.M

```
%displays eigenfaces and the mean face if the database is not EMPTY
%partially taken from the software karhunenloeve.m
%Author:Alex Chirokov
%Available at
%http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId
=6995&objectType=file

close all
if(exist('fdata.dat')==2)
        try
        load('fdata.dat','-mat');
       mtr=zeros(size(data{1,1},1),fnumber);
        for ii=1:fnumber
            mtr(:,ii)=double(data{ii,1});
            imsize=[250 250];
            nPixels = imsize(1)*imsize(2);
            mtr2=double(mtr)/255;
            avrgx = mean(mtr2')';
           for i=1:fnumber
            mtr2(:,i) = mtr2(:,i) - avrgx;
           end

           imshow(reshape(avrgx, imsize)); title('mean face')
        end


cov_mat = mtr2'*mtr2;

[V,D] = eig(cov_mat);


V = mtr2*V;


figure
if fnumber>7

    for ii=1:fnumber
    subplot(fnumber-6,3,ii) ;imshow(ScaleImage(reshape(V(:,ii),imsize)));

end
elseif fnumber<7
```

```
    for ii=1:fnumber
    subplot(3,3,ii) ;imshow(ScaleImage(reshape(V(:,ii),imsize)));
    end
end


bdrfacerec

        catch
disp('Mean face and eigenfaces cannot be shown!!!')
disp('Possible Reasons:')
disp(' ')
disp('1--> Check the size of the new image and stored image(s) if you
change the imresize line at bdrfacerec.m')
disp('2--> Database is empty')
disp('3--> There is only one person in your database. Please add atleast
one person more to see the average of faces')
       pause

        bdrfacerec
 end
else
    clc
    disp(' CORRESPONDING FACE DATABASE NOT FOUND !!!')
    disp(' ')
    disp(' Press any key to continue ')
        pause
bdrfacerec
end
```

FACEREC.M

```
%FACE RECOGNITION SCRIPT
%First takes the last selected image
%Calls the images from database
%Applies PCA to images
%Compares the last selected image with database images and says if it is
%found at the database or not.


clc
close all



try
img2=double(img(:));

if(exist('fdata.dat')==2)

        load('fdata.dat','-mat');
        mtr=zeros(size(data{1,1},1),fnumber);
        for ii=1:fnumber
            mtr(:,ii)=double(data{ii,1}); %Calling the images from
database.Represents each of them as a vector gamma1,gamma2,...gammaM
        end
```

```
            total=sum(mtr,2) ;              %Sum of the face vectors

            avr=total/fnumber ;            %Taking the average value.

            for i=1:fnumber
            A(:,i) = mtr(:,i) - avr;       %calculates fi for each face
and produces A matrix.

            end



Lmat = A'*A    ;                          %Obtaining the L matrix

[V,D] = eig(Lmat);                        %eigen values(D) and
eigenvectors(V) of the L matrix

  %obtaining eigenfaces
Ui = A*V;




        f1=Ui'*(img2-avr) ;             % feature vector of the new image
Omeganew



         fdata=zeros(max_class,max_class); %empty matrice


         for ii=1:fnumber
             imdata=double(data{ii,1});
             classdata=data{ii,2};
             cor=Ui'*(imdata-avr); % weights  of the stored faces

             fdata(:,classdata)=fdata(:,classdata)+cor; %feature vector ?

         end

         dist=zeros(max_class,1);
         for ii=1:(max_class)
             dist(ii)=norm(f1-fdata(:,ii)); %calculation of euclidean
distance

         end

         [minf,pminf]=min(dist);  %finding the minimum distance
         k=minf/10000000;
     pause
         hll = waitbar(0,'CHECKING THE DATABASE...');
 for i=1:2450,
 waitbar(i/100)
```

```
 end
 close(hll)

         if k<=10.1

             disp(strcat('The corresponding ID number of the face is -->
',num2str(pminf)));

             pause
           ldinfo

             return
         end
         if k>=10.1
             disp('The corresponding face does not exist at your
database')
             disp('Note: Save more pictures to the group for the best
recognition')
             disp(strcat('The similar of this face has an ID, number-->
',num2str(pminf)));

             pause
        ldinfo


            return
         end

else
                        clc
      disp('IMAGE PROCESSING IS NOT AVAILABLE')
      disp('*********************************')
      disp('Possible Reasons:                ')
      disp(' ')
      disp('1- Database is empty')
      disp('2- The size of the selected image is not suitable for
processing')
      disp('3- The color or format is not matching with database')
      disp(' ')
      disp('Press any key to continue')
      pause
      clear all
      bdrfacerec
      end


    catch
    clc
    disp(' ')
    disp('NO FACE HAS BEEN SELECTED!!')
pause
clear all
bdrfacerec
end
```

GINFO.M

```
%Display information menu
%Shows the database files
%Jumps to "dinfo.m" for displayin information

clc
clear all
disp('****************************')
disp('Display information       [1]')
disp('Show database files       [2]')
disp('****************************')
disp('Back                      [B]')
disp(' ')
k=input('Your Choice-->','s');

switch(k)
    case '1'
        dinfo

    case '2'
            clc
        disp(' ')
        disp(' ')
        disp(' Files on the selected directory are , ')
              dir *.dat
        pause
ginfo
    case 'b'
        bdrfacerec
    case 'B'
        bdrfacerec




    otherwise
        clc
        disp('Wrong SELECTION!!')
        disp(' ')
        disp('Press any key to continue...')
   pause
   ginfo
end
```

LDINFO.M

```
% After facerecognition, this code captures the ID number of a person.
%Then reads the information stored in IDnumber.DAT

clc
```

```
jk=num2str(pminf);

if(exist([jk '.dat']))
    load ([jk '.dat'],'-mat');
switch (h)
    case 'y'

prl
    case 'Y'
        prl

    case 'n'
        prl2
    case 'N'
        prl2
    otherwise
        prl2
    end


  if exist('kl')
try

img=imresize(img,[250 250]);

subplot(1,2,2); imshow(img);
subplot(1,2,1); imshow(kl);title('Refference Image')

disp (strcat('NAME............',num2str(a)))
disp (strcat('SURNAME.........',num2str(b)))
disp (strcat('PHONE NUMBER.....',num2str(c)))

pause
clear all
bdrfacerec
catch
clc
    disp('')
    disp('An error occured while reading the database')
    disp('Please check your variables and try again')
    pause
    clear all
    bdrfacerec
return
end

  else

try
clc
disp (strcat('NAME............',num2str(a)))
disp (strcat('SURNAME.........',num2str(b)))
disp (strcat('PHONE NUMBER.....',num2str(c)))

disp('Refference Image NOT FOUND!!')
pause
clear all
bdrfacerec
```

```
catch
   clc
    disp('')
    disp('An error occured while reading the database')
    disp('Please check your variables and try again')
    pause
    clear all
    bdrfacerec
    return
end
  end
else
    disp('THERE IS NO RECORD ABOUT THE PERSON')
    pause
    clear all
    bdrfacerec
return
  end
```

PRL.M

```
% This code gives logical 0 1 0 1 to parrallel port.
% Waits a moment and gives logical 0 0 1 0 (values at start.m)


z=1;
for z=1:500
i=1;
k=1;
parport=digitalio('parallel','LPT1');
line=addline(parport,0:3,'out');
putvalue(parport,[0 1 0 1]);
end


putvalue(parport,[0 0 1 0]);
  gval = getvalue(parport);
```

PRL2.M

```
% This code gives logical 1 0 0 1 to parrallel port.
% Waits a moment and gives logical 0 0 1 0 (values at start.m)

zi=1;
for zi=1:500
i=1;
k=1;
parport=digitalio('parallel','LPT1');
line=addline(parport,0:3,'out');
putvalue(parport,[1 0 0 1]);
end


putvalue(parport,[0 0 1 0]);
```

```
  gval = getvalue(parport);
```

SCALEIMAGE.M

```
%This code arranges the size of mean face and eigenfaces

%Author:Alex Chirokov
%Available at
%
http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=
6995&objectType=file

function out=ScaleImage(in)
in_min=min(min(in));
in_max=max(max(in));

out = (in-in_min)/ (in_max-in_min);
```

REFFERENCES

1. A. J. Goldstein, L. D. Harmon, and A. B. Lesk, “Identification of Human Faces,” Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.

2. L. Sirovich and M. Kirby, "*A Low-Dimensional Procedure for the Characterization of Human Faces,*" J. Optical Soc. Am. A, 1987,Vol. 4, No.3, 519-524.

3. M. A. Turk and A. P. Pentland, "*Face Recognition Using Eigenfaces*," Proc. IEEE, 1991, 586-591.

4. P. Pentland, B. Moghaddam, T. Starner "*View-based and Modular Eigenspaces for Face Recognition*," Conf. IEEE, 1994, No.245

5. *Advantages and Disadvantages of Eigenfaces* , http://et.wcu.edu/aidc/BioWebPages/ eigenfaces.htm

6. X.Lu, “*Image analysis for face Recognition*.” http://www.face-rec.org/interesting-papers/

7. T. V. Wilson, “*How Biometrics Work,*” http://computer.howstuffworks.com/biometrics.htm/

8. A. Chirokov, “*Karhunen-Loeve Expansion Software,”* http://www.mathworks.com / matlabcentral/fileexchange/loadFile.do?objectId=6995&objectType=file

9 J.Kruger, M.Robinson, D. Kochelek, M. Escarra, “*Obtaining the Eigenface Basis,*” Proj, 2004,4-5.

10 V. Jain, A. Mukherjee, Indian Face Database, http://vis-www.cs.umass.edu /~vidit/AI/dbase.html