1. INTRODUCTION

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced paralel computers. The goal of this manipulation can be divided into three categories:

- Image Processing *image in ➔ image out*
- Image Analysis *image in ➔ measurements out*
- Image Understanding *image in ➔ high-level description out*

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past few years.

Given the requirement for determining people's identity, the obvious question is what technology is best suited to supply this information? There are many different identification technologies available, many of which have been in wide-spread commercial use for years. The most common person verification and identification methods today are Password/PIN (Personal Identification Number) systems, and Token systems (such as your driver's license). Because such systems have trouble with forgery, theft, and lapses in users' memory, there has developed considerable interest in biometric identification systems, which use pattern recognition techniques to identify people using their physiological characteristics. Fingerprints are a classic example of a biometric; newer technologies include retina and iris recognition.

In the last couple of years we have seen an enormous growth in electronically available services, such as banking through ATMs, the internet and voice services (phone). Humans are integrated closer to computers every day, and computers are taking over many services that used to be based on face to face contact between humans. This has prompted an active development in the field of biometric systems.

Face recognition is more advantageous than the other biometrics used. Whereas many biometrics require the subjects co-operation and awareness in order to perform an identification or verification, such as looking into an eye scanner or placing their hand on a fingerprint reader, face recognition could be performed even without the subject's knowledge.

First, main titles will be explained , such as digital image, representation of digital images, RGB color model, recognition techniques, Eigenfaces, and a mathematical procedure (Gramm Schmidt Algorithm) for the technique which we offered.

2. WHAT IS DIGITAL IMAGE

We may define an image as a two-dimensional function, $f(x,y)$, where x and y spatial coordinates. The amplitude of $f(x,y)$ at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x , y and the amplitude values of f are all finite and discrete quantities, we call the image a ***digital image***. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as ***picture elements*** , ***image elements*** , ***pels*** , and ***pixels***. ***Pixel*** is the term most widely used to denote the elements of a digital image.

We call digital image processing which is a pool consisting of processes whose inputs and outputs are images and , in addition, processes that extract attributes from images, up to and including the recognition of individual objects.

One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York. The time required to transport a picture accross the Atlantic decreased from more than a week to less than three hours by introduction of the Bartlane cable picture transmission system in the early 1920s. Specialized printing equipment coded pictures for cable transmission and then reconstructed at the receiving end. Figure-1 below was transmitted in this way.

Figure-1 A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces. (McFarlane)

3. REPRESENTATION OF DIGITAL IMAGES

The result of sampling and quantization is a matrix of real numbers. Assume that an image ***f* (*x*,*y*)** is sampled so that the digital image has ***M*** rows and ***N*** columns.The values of the coordinates **(*x*,*y*)** now become discrete quantities (integer values). The values of the coordinates at the origin are **(*x*,*y*) = (**1,1**)**. The next coordinate values along the first row are represented as **(*x*,*y*) = (**1,2**)**. It is important to keep in mind that the notation **(**1,2**)** is used for the second sample along the first row.

Figure-2 shows the coordinate convention.

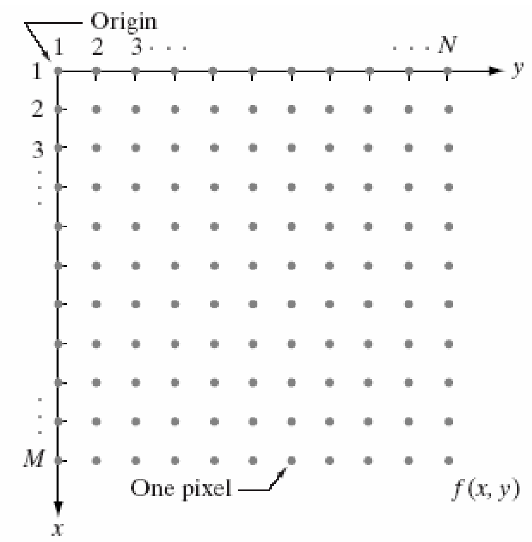


Figure-2 Coordinate convention to represent digital images.

We can write ***f* (*x*,*y*)** as a matrix as shown below.

$$f(x,y) = \begin{bmatrix} f(1,1) & \cdots & f(1,N) \\ \vdots & & \vdots \\ f(M,1) & \cdots & f(M,N) \end{bmatrix}$$

Right side of this equation is by definition a digital image. Each element of this matrix array is called a ***pixel.***

At the same time we can use a more traditional matrix notation , so it becomes:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & & \vdots \\ a_{M,1} & \cdots & a_{M,N} \end{bmatrix}$$

$$f(x=i, y=j) = f(i, j) = a_{i,j}$$

In general at this notation the first element is written as $a_{0,0}$ but we wrote $a_{1,1}$. We used Matlab at the project, first element of a matrix **A** is denoted by **A(1,1)** in Matlab so we changed the notation little.

This matrix represents a digital image, every element a value for color of that pixel. Now we will give detailed information about color models to understand the simulation. We used RGB color model.

4. COLOR MODELS

The purpose of a color model is to facilitiate the specification of colors, generally in an accepted way. A color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.

Most color models in use today are oriented either toward hardware or toward applications where color manipulation is a goal. In terms of digital image processing, the hardware oriented models most commonly used in practice are the RGB (red, green, blue) model for color monitors and a broad class of color video cameras; the CMY (cyan, magenta, yellow) and CMYK (cyan, magenta, yellow, black) models for color printing; and the HSI (hue, saturation, intensity) model, which corresponds closely with the way human describe and interpret color.

4.1 The RGB Color Model

In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on Cartesian coordinate system. It is shown in figure-3.

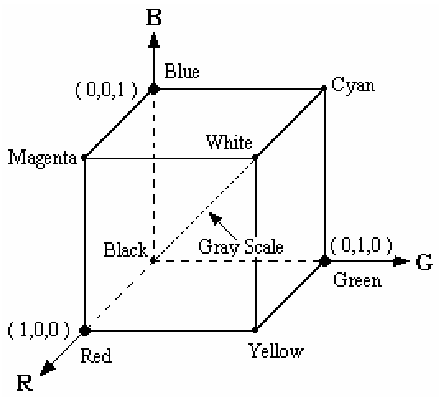


Figure 3 Schematic of the RGB color cube.
Points along the main diagonal have gray values, from black at the origin to white at point (1,1,1).

RGB values are at three corners in directions (R,G,B); cyan, magenta and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale (points of equal RGB values) extends from black to white along the line connecting these two points. The different colors in this model are points on or inside the cube. Colors are defined by vectors extending from the origin.

We call the cube in figure3 ***unit cube,*** if all color values have been normalized. Values of R,G,B are in the range [0,1].

In the RGB color model images have three component images representing each primary color. Images gray scaled have three component images corresponding to three matrices as Red, Green, and Blue. Values of Red, Green, and Blue matrices are equal. Because the gray scale is the line formed by the points of equal RGB values. We pressed this situation since our database consisting of gray scaled images.

Actually this model is enough to understand the project. So we give less detailed information for other models.

4.2 The CMY and CMYK Color Models

Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using the simple operation

$$\mathbf{C = 1 - R \ , \ M = 1 - G \ , Y = 1 - B}$$

where the assumption is that all color values have been normalized to the range [0,1].
Difference between the CMY and CMYK color models is the fourth color, black. It is added to the CMY and the CMYK model is formed.

4.3 The HSI Color Model

The RGB, CMY, and other similar color models are not well suited for describing color in terms that are practical for human interpretation. For example, one does not refer to the color of an object by giving the percentage of each of the primaries composing its color. Furthermore, we do not think of color images as being composed of three primary images that combine to form that single image.

When humans view a color object, we describe it by its hue, saturation, and brightness. Hue is a color attribute that describes a pure color, whereas saturation gives a measure of the degree to which a pure color is diluted by white light. Brightness is a subjective descriptor that is practically impossible to measure. Intensity is a most useful descriptor of monochromatic (= all one color) images. This quantity definitely is measurable and easily interpretable. As a result the HSI model is an ideal tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans who are the developers and users of these algorithms.

There is RGB to HSI conversion, but we do not see it necessary to give. We discussed the general information about Digital Image Processing, now we can examine Biometric Systems.

5. BIOMETRIC SYSTEMS

The meaning of *Biometrics* comes from the Greeks. The Greek hybrid of the words is bio meaning life and metry meaning to measure. The Webster's definition is the statistical measurement and analysis of biological observations and phenomena.

A brief history of *Biometrics*;

- The first recorded use of biometrics was fingerprints by the ancient Assyrians and Chinese for the signing of legal documents
- The first modern study of fingerprints was made by the Czech physiologist Johannes Evangelista Purkinje who proposed a classification system in 1823

- The use of fingerprints for identification purposes was proposed late in the 19th century by the British scientist Sir Francis Galton
- In the 1890s the police in Bengal, India, under the British police official Sir Edward Richard Henry (1857-1930) began using fingerprints to identify criminals.
- Sir Edward Richard Henry established the first British fingerprint files in London in 1901
- This spread rapidly throughout Europe and the U.S, superseding the old Bertillon system of identification by means of body measurements.

Biometrics refers to the automatic identification or verification of living persons using their enduring physical or behavioral characteristics. Many body parts, personal characteristics, and imaging methods have been suggested and used for biometric systems: fingers, hands, faces, eyes, voices, signatures, typing styles, DNA, and so on. The body parts most often used in current applications are fingerprints and facial characteristics.

Biometric systems are systems that identify or verify human beings. They are typically based on some single biometric feature of humans, but several hybrid systems also exist. Some examples of biometrics used are:

signature the pattern, speed, acceleration and pressure of the pen when writing one's signature
fingerprint the pattern of ridges and furrows on the surface of the fingertip
voice the way humans generate sound from vocal tracts, mouth, nasal cavaties and lips
iris the annular region of the eye bounded by the pupil and the sclera
retina the pattern formed by veins beneath the retinal surface in an eye
hand geometry measurements of the human hand
ear geometry measurements of the human ear
facial thermogram the heat that passes through facial tissue
face the most natural and well known biometric

To measure the real-life performance of biometric systems—and to understand their strengths and weaknesses better—we must understand the elements that comprise an ideal biometric system.

In an ideal system

- all members of the population possess the characteristic that the biometric identifies, like irises or fingerprints;
- each biometric signature differs from all others in the controlled population;
- the biometric signatures don't vary under the conditions in which they are collected; and
- the system resists countermeasures.

Biometric-system evaluation quantifies how well biometric systems accommodate these properties. Typically, biometric evaluations require that an independent party design the evaluation, collect the test data, execute the test, and analyze the results.

For example, if you plan to use a biometric to reduce—as opposed to eliminate— fraud, then a low-performance biometric system may be sufficient. On the other hand, completely replacing an existing security system with a biometric-based one may require a high-performance biometric system, or the required performance may be beyond what current technology can provide.

Biometric systems process raw data to extract a *biometric template*—a small set of data that can be uniquely derived given a biometric feature. Various algorithms process biometric data to produce a template. For example, in a face-recognition system, facial geometry algorithms work by defining a reference line—for example, the line joining the pupils of the eyes—and using it to measure the distance and angle of various facial features relative to this reference. Templates are easier to process and store than the original raw data.

Biometric systems fall into two categories: authentication and identification, with authentication systems being far more common. To be authenticated by a system, a subject presents a password or a token such as an ID card, along with a live biometric sample such as a fingerprint. The system accesses a record based on the token, then compares the sample's biometric data with the record's sample to authenticate the subject's identity.

Authentication systems are reliable and efficient if the subject base is small and the biometric readers are accurate and durable. Airports, prisons, and companies that need secure access use systems such as these.

Implementing identification systems is more difficult. To be identified by a system, a subject provides biometric data, and the system must find a record based on that data only—which can require a search of the entire database. Performing this search takes a long time and even then will only rarely result in a single-record match. This means that the system must perform additional filtering.

Keep in mind that these searches are not text-based. Because biometric data is *pattern-based*, finding a hit requires specialized algorithms that focus on finding specific patterns in certain aspects of the data.

5.1 Retinal Scans

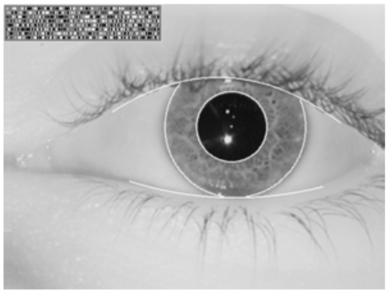
Electronic scan of the innermost layer of the eyeball's wall.

Advantages: Retina generally remains stable through life, ensuring accuracy.

Disadvantages: Requires close physical contact MW scanning device; may not be generally accepted by public.

5.2 Iris recognition

Recording of iris using standard video technology.

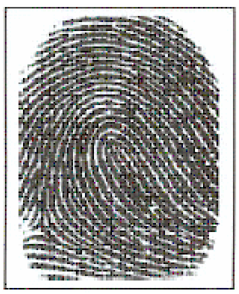


- Iris code developed by John Daugman at Cambridge.
- Extremely low error rates.
- Fast processing.
- Monitoring of pupils oscillation to prevent fraud.
- Monitoring of reflections from the moist cornea of the living eye.

Advantages: High accuracy; Long term stability; Nearly non-intrusive; Fast processing

Disadvantages: Not exactly easy to use; High false non-match rates; High cost

5.3 Finger imaging



Recording of fingerprint using optical scanner.

- Identification points consist of bifurcations, ending ridges, dots, ridges and islands.

- A single rolled fingerprint may have as many as 100 or more identification points that can be used for identification purposes.

Advantages: Mature technology; Easy to use/non-intrusive; High accuracy; Long-term stability; Ability to enrol multiple fingers.
Disadvantages: Inability to enrol some users; Affected by skin condition; Association with forensic applications.

5.4 Hand geometry

Three-dimensional recording of length, width and height of hand and fingers, using optical scanner.
Advantages: User-friendly; requires small amount of computer storage space.
Disadvantages: Isn't as unique as other biometric methods; hand injury can cause recognition problems.

5.5 Voice verification or recognition

Acoustic signal of voice converted into digital code.
Advantages: Works well over the telephone.
Disadvantages: Requires large amount of computer storage; people's voices can change; background noises can interfere.

5.6 Signature recognition

- Signatures in wide use for many years.
- Signature generating process a trained reflex - imitation difficult especially 'in real time'.
- Automatic signature recognition measures the dynamics of the signing process.

Advantages: Resistance to forgery; Widely accepted; Non-intrusive; No record of the signature.
Disadvantages: Signature inconsistencies; Difficult to use; Large templates (1K to 3K).

5.7 Face recognition

We will give detailed explanation in the next chapter.

All the Biometric Technologies compared at Table 1.

Biometric Technology	Truth	Trusting	Social Acceptance	Speed (sn)	Backup
Face Recognition	High	High	Very High	1.5	Human
Fingerprint Matching	Very High	Low	Low	6.0	-
Hand Geometry	High	Low	Low	5.0-15.0	-
Iris Scanning	Very High	Low	Very Low	5.0-15.0	-
Voice Recognition	Low	Low	Very High	10.0	-
Signature Matching	Low	Low	Low	3.0-5.0	Human

Table 1 Comparison of the Biometric Technologies

6. FACE RECOGNITION

Although you can choose from several general strategies for evaluating biometric systems, each type of biometric has its own unique properties. This uniqueness means that each biometric must be addressed individually when interpreting test results and selecting an appropriate biometric for a particular application.

In the 1990s, automatic-face-recognition technology moved from the laboratory to the commercial world largely because of the rapid development of the technology, and now many applications use face recognition. The history of Face Recognition is shown at Figure-4.

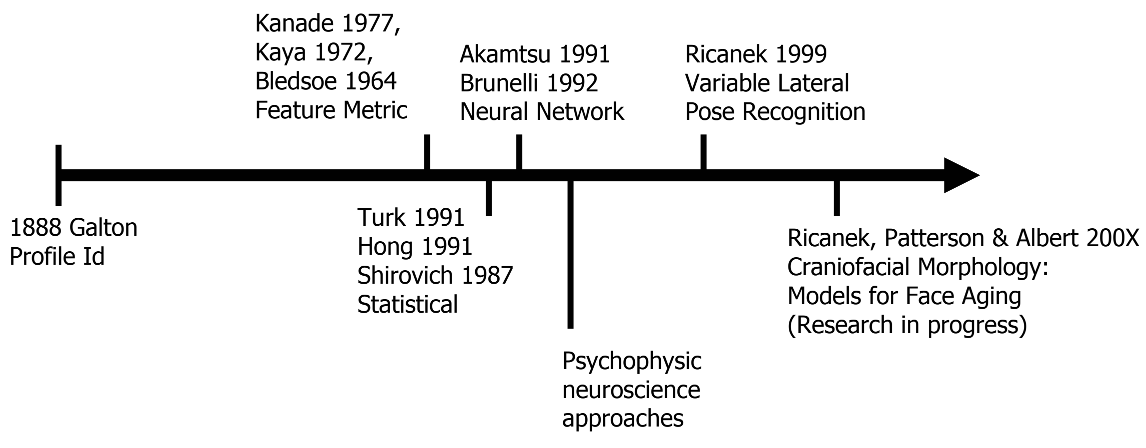


Figure-4 The history of Face Recognition

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 faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expression, aging, and distractions such as glasses, beards or changes in hair style.

Face recognition 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 vastly improve criminal identification.

There is a growing interest in face authentication, for use in such application areas as:

Table 2. Typical Applications of Face Recognition

Areas	Specific applications
Entertainment	Video game, virtual reality, training programs
	Human-robot-interaction, human-computer-interaction
Smart cards	Drivers' licenses, entitlement programs
	Immigration, national ID, passports, voter registration
	Welfare fraud
Information security	TV Parental control, personal device logon, desktop logon
	Application security, database security, file encryption
	Intranet security, internet access, medical records
	Secure trading terminals
Law enforcement and surveillance	Advanced video surveillance, CCTV control
	Portal control, postevent analysis
	Shoplifting, suspect tracking and investigation

There are many techniques used in recognition. We used Eigenfaces Technique in our project, so we will not give detailed information about the others.

Face Recognition can be classified into three different classes:

➔ **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, or invariant moments, are extracted to represent a face. Feature-based 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. These are the specific feature based approaches listed below;

-Facial Features
-Texture
-Skin Color
-Multiple Features
-Gabor Wavelet Decomposition

➔ **Template Based** (Photometric)

The simplest template-matching approaches represent a whole face using a single template, i.e., a 2-D array of intensity, 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 template-matching is its simplicity, however, it suffers from large memory requirements and an inefficient matching algorithm. These are the specific template based approaches listed below;

-Predefined Face Templates
-Deformable Templates

➔ **Appearance Based**

The idea of appearance-based approaches 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, fisherface space and illumination subspace have been proposed for dealing with recognition under varying illumination. These are the specific appearance based approaches listed below;

-Eigenface
-Distribution-based

-Neural Network
-Suppor Vector Machine
-Naive Bayes Classifier
-Hidden Markov Model
-Information Theorical Approach

We used Eigenface method in our simulation, so explaining this method is sufficient.

7. EIGENFACE METHOD

Eigenfaces approach is a principal component analysis method, 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. 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.

7.1 Calculating Eigenfaces

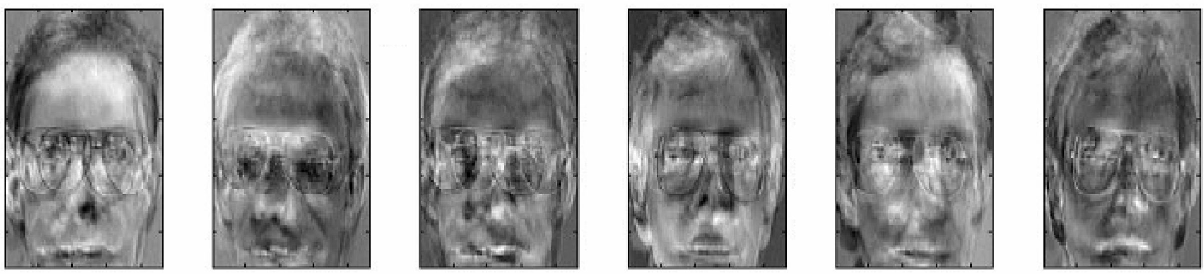
Let a face image I(x,y) be a two-dimensional N x N array of 8-bit intensity values. An image may also be considered as a vector of dimension N^2, so that a typical image of size 92 x 92 becomes a vector of dimension 8464, or equivalently a point in 8464-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis (or Karhunen-Loeve expansion) is to find the vectors that best account for the distribution of face images within the entire image space.

These vectors define the subspace of face images, which we call "face space". Each vector is of length N^2, describes an N x N image, and is a linear combination of the original face images.

Figure-5 Face images used in calculating eigen-faces

Figure-6 Average image



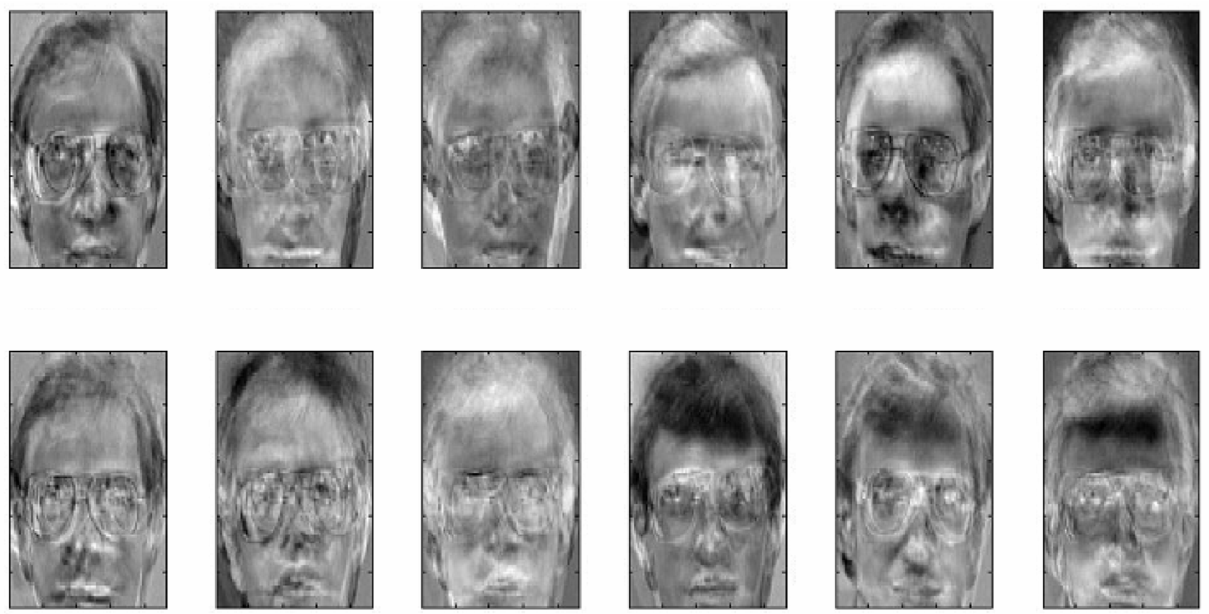


Figure-7 Eigen-faces calculated from sample images

Let the training set of face images be $\Gamma_1, \Gamma_2, ..., \Gamma_M$ (shown in Figure-5) then the average of the set is defined by

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

Each face differs from the average by the vector

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

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The kth vector, u_k, is chosen such that

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

is a maximum, subject to

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

The vectors u_k and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix

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

where the matrix $A = [\ \Phi_1\ \Phi_2\ ...\ \Phi_M\ ]$. The covariance matrix C, however is $N^2 x N^2$ real symmetric matrix, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space ($M<N^2$), there will be only M-1, rather than N^2 , meaningful eigenvectors. The remaining eigenvectors will have associated eigenvalues of zero. We can solve for the N^2 dimensional eigenvectors in this case by first solving the eigenvectors of an M x M matrix such as solving 18 x 18 matrix rather than a 8464 x 8464 matrix and then, taking appropriate linear combinations of the face images Φ_i .

Consider the eigenvectors v_i of A^TA such that

$$A^T A v_i = \mu_i v_i$$

Premultiplying both sides by A, we have

$$AA^T A v_i = \mu_i A v_i$$

from which we see that Av_i are the eigenvectors of $C = AA^T$.

Following these analysis, we construct the M x M matrix $L = A^TA$, where $L_{mn} = \Phi^T_m\ \Phi_n$, and find the M eigenvectors, v_l , of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces u_l .

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

7.2 Classifying a Face Image

A new face image (Γ) is transformed into its eigenface components (projected onto "face space") by a simple operation,

$$w_k = u_k^T (\Gamma - \Psi)$$

for k = 1,...,M. This describes a set of point by point image multiplications and summations, operations performed at approximately frame rate on current image processing hardware.

The weights form a feature vector,

$$\Omega^T = [w_1 w_2 \dots w_{M'}]$$

that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The feature vector is then used in a standard pattern recognition algorithm to find which of a number of predefined face classes, if any, best describes the face.

$$\frac{\|\Omega - \Omega_k\|}{\|\Omega_k\|} \le \varepsilon_k$$

7.3 Experimental Results

In our simulation we used training set shown in Figure-5. We examined the epsilon vector to see the result for every input image. Figure-9 shows epsilon for an input image (Figure-8) from training set.

Figure-8 input image from training set

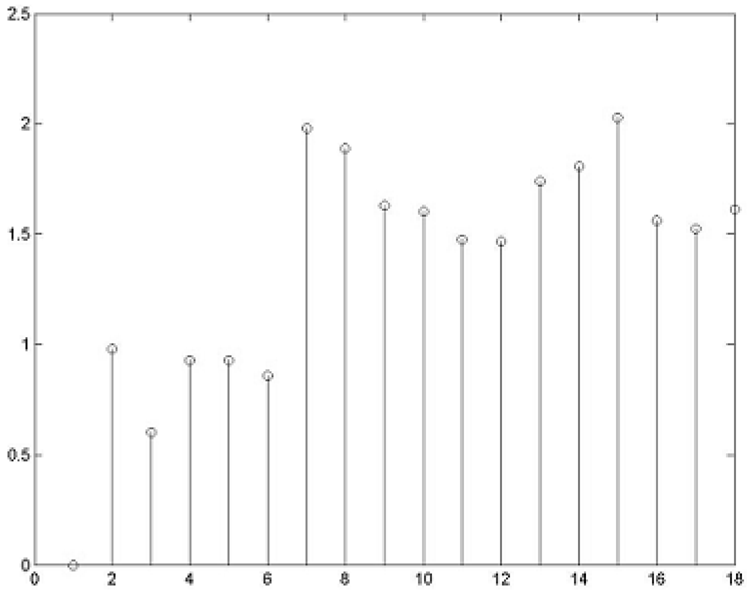


Figure-9 Epsilon vector for input image from training set

Epsilon vector's values change according to the image selected as an input. If image is a member of training set, one coefficient epsilon vector will be zero corresponding to the image's place.

We repeated the operation for the image above which has beard. The result is shown below.

Figure-10 Input image having beard

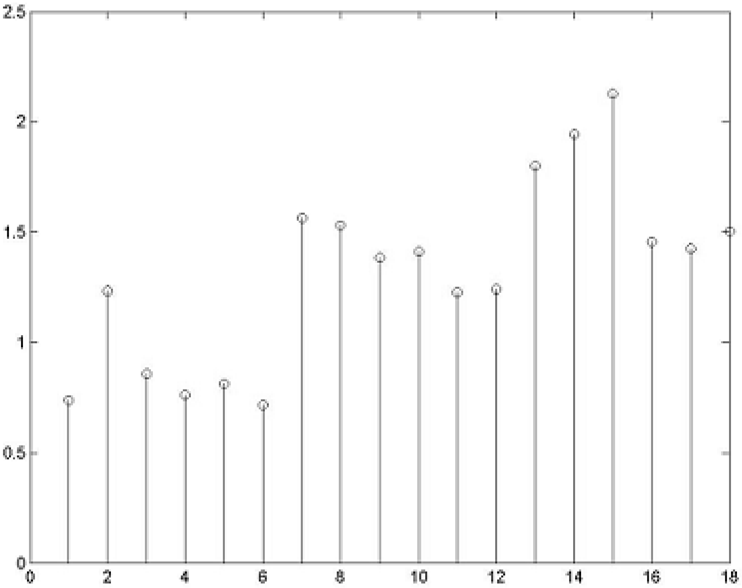


Figure-11 Epsilon vector for the input image

We repeated the operation for the image having sunglasses from training set. The result;

Figure-12 Input image having sunglasses

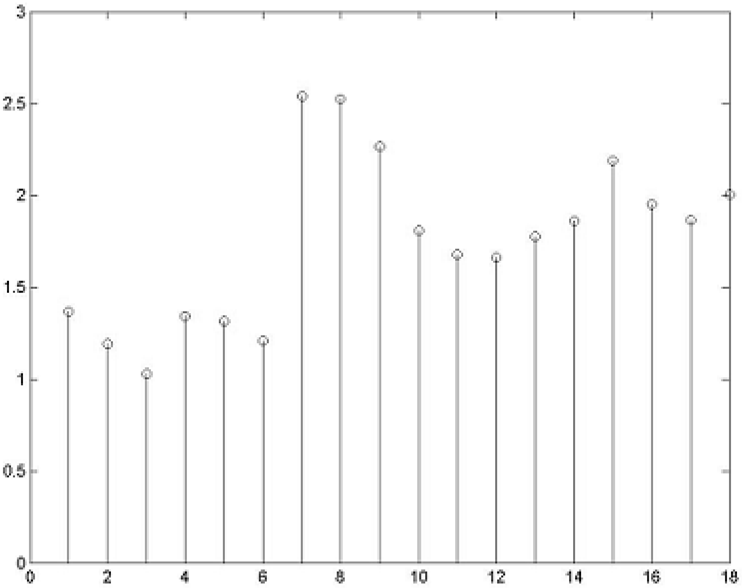


Figure-13 Epsilon vector for the input image

Now we will see the result for an image which the training set don't have. But there are three images of same person in the set.

Figure-14 Input image

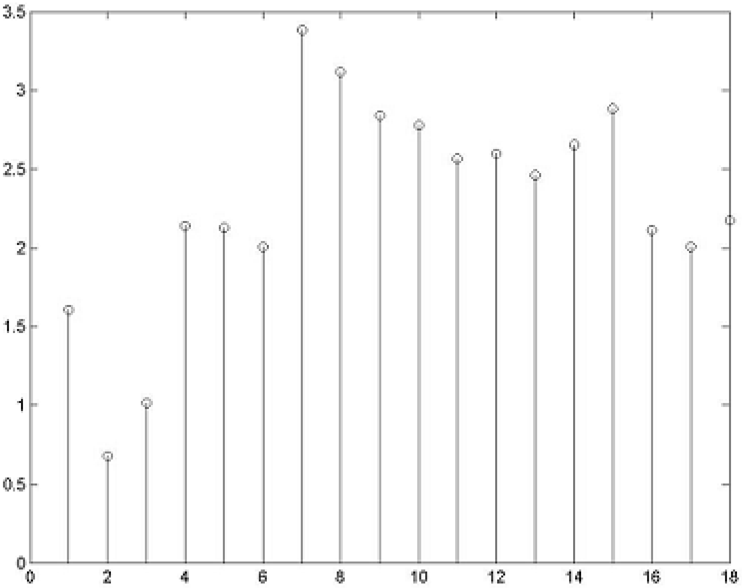


Figure-15 Epsilon vector for the input image

We chose the threshold value as 1. If minimum value of epsilon vector is less than 1, simulation shows the picture corresponding to the index of minimum. If it is greater than one, Matlab displays that "The person searched was not found."

After examining the results, we saw that there is a probability of taking wrong outcome.

8. A NEW APPROACH FOR FACE RECOGNITION

In the training set every image represents a (92x92)matrice, so we have 18 square matrices. If we generate orthonormal basis matrices for this training set, we can regenerate every elements of training set by linear combination of orthonormal basis matrices using required coefficients. So we can not form an image out of set exactly.

We formed our database from gray scaled images. One image consists of three matrices corresponding to Red, Green, and Blue. Because of the gray scale these three matrices have same values. So we express these three matrices as one matrice using one of them.

Suppose that we have M square matices (faces). We will make orthonormal basis matrices from these matrices.

The Gramm Schmidt Method is used to find orthonormal matrices.

Let $\{ F_1 , F_2 , \ldots, F_M \}$ be a non-orthonormal basis. We derive $\{ f_1 , f_2 , \ldots, f_M \}$ an orthonormal basis from it as follows:

$$E_1 = F_1 \Rightarrow f_1 = E_1 / \| E_1 \|$$
$$E_2 = F_2 - < F_2 , f_1 > f_1 \Rightarrow f_2 = E_2 / \| E_2 \|$$
$$E_3 = F_3 - < F_3 , f_2 > f_2 - < F_3 , f_1 > f_1 \Rightarrow f_3 = E_3 / \| E_3 \|$$
$$\vdots$$
$$E_M = F_M - \sum_{k=1}^{M-1} < F_M , f_k > f_k \Rightarrow f_M = E_M / \| E_M \|$$

where || || represents the oclid norm.

After these mathematical operations we have an orthonormal basis.
Now we can show any face as a linear combination of orthonormal-faces. The input face is A. The output face is A'.

A' is calculated by Projection Theorem.

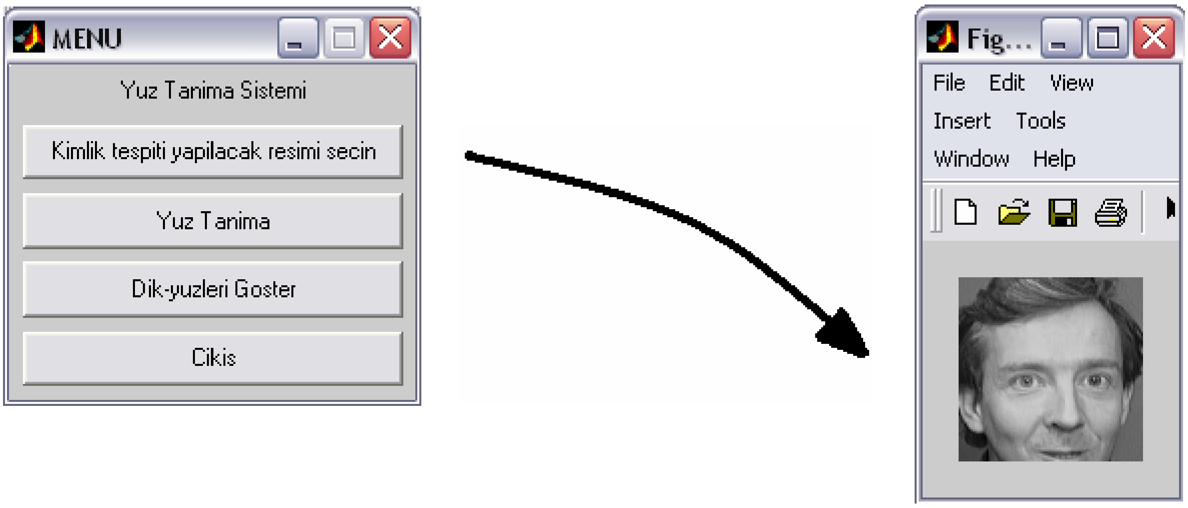
$$A' = \sum_{k=1}^{M} < A , f_k > f_k$$

where the symbol < , > represents the inner product.

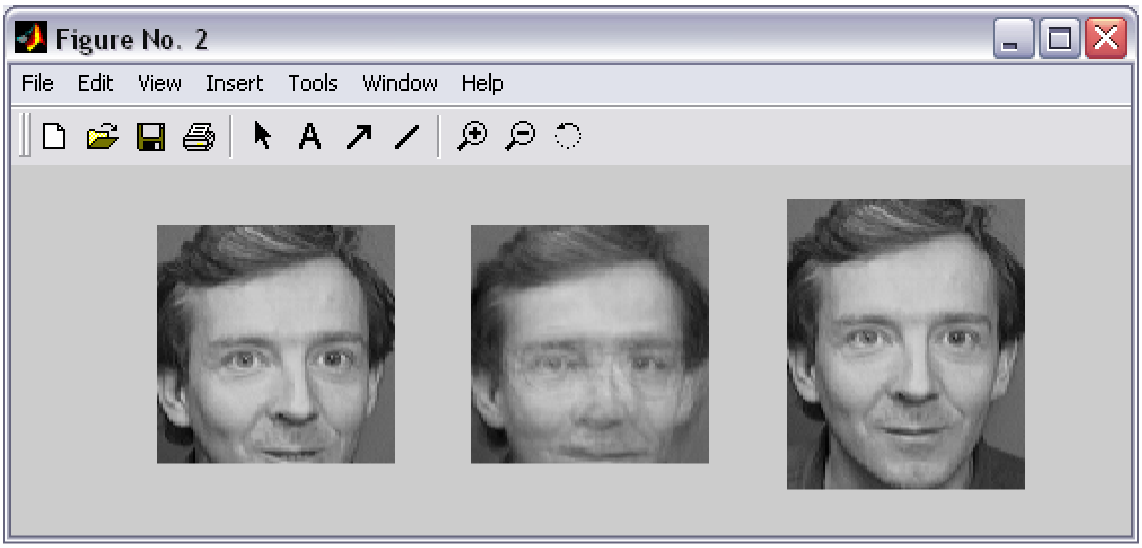
In our recognition problem the threshold was determined by experimental results. We used Oclid distance $\epsilon_i = \| A' - \Gamma_i \|$. We compare the distance with threshold value. If it is less than threshold, the person's picture will be shown otherwise Matlab displays that "The person searched was not found."

8.1 Experimental Results

First step choosing the person searched.

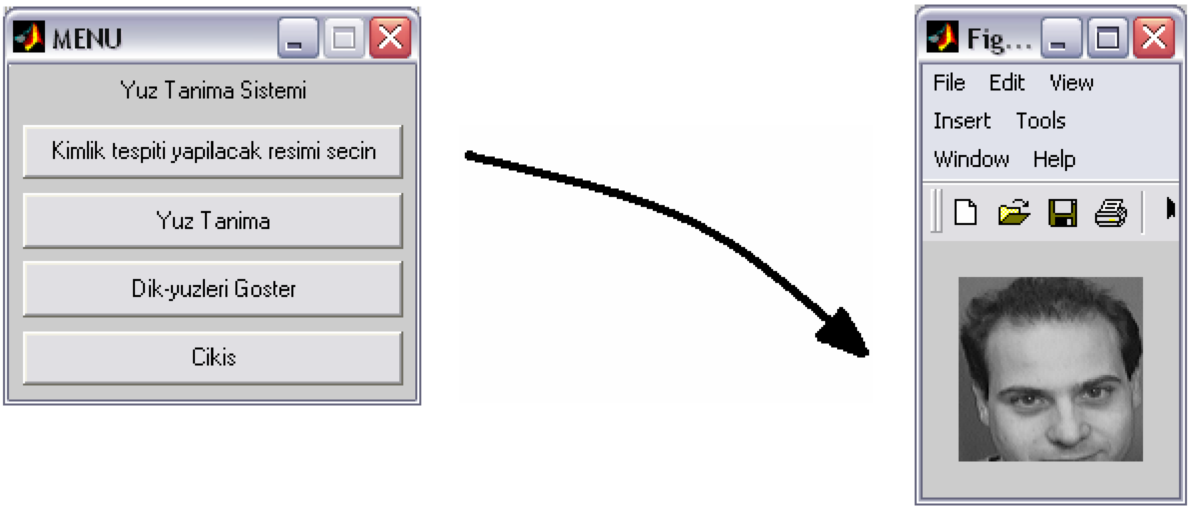


Second step is running the face recognition.

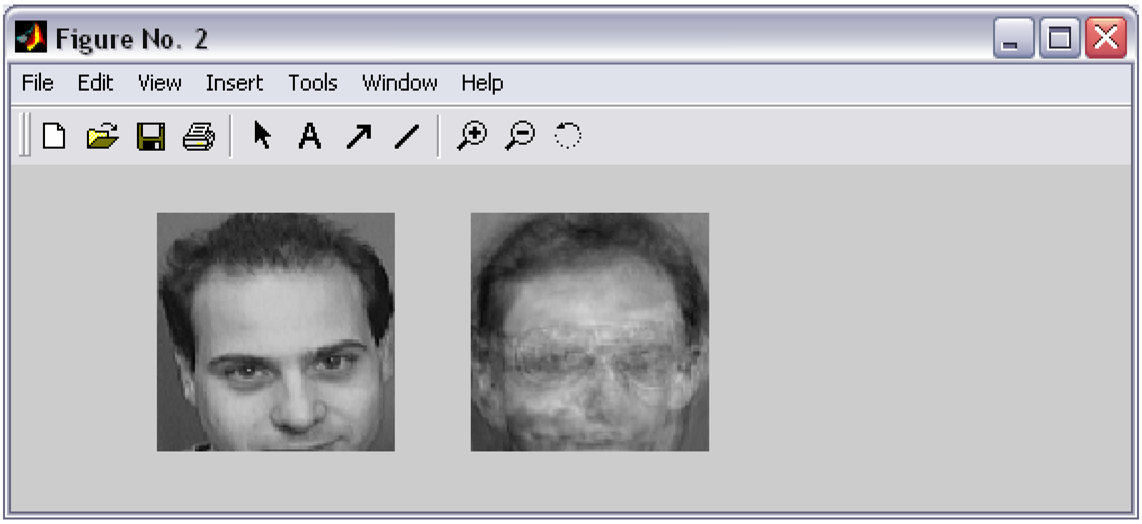


The input picture is not in the training set. But the person’s other pictures are in the set.

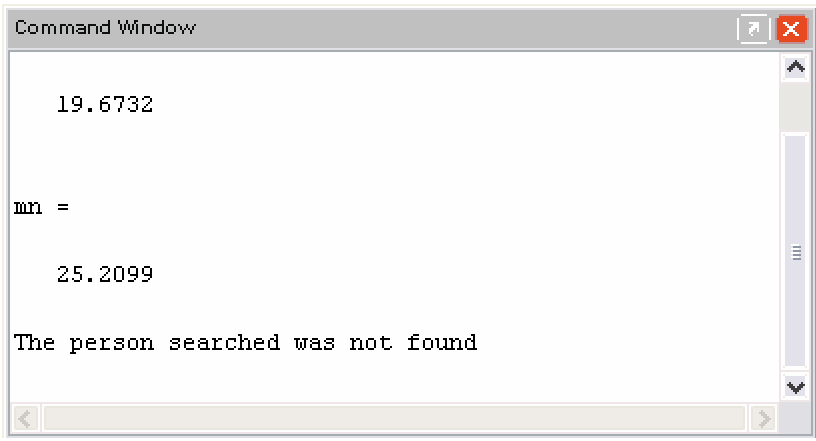
We run the simulation again to see the result for a picture out of the set.



Result of simulation;



The person was not found.



9. CODES USED IN SIMULATION

9.1 Eigen-Face recognition technique

```
clc;clear;close all

for i=1:1:6
   for j=1:1:3
exampleImage(:,:,:,i,j)=
double(imread(strcat('C:\MATLAB6p5\work\s',num2str(i),'\',num2str(j),'.bmp'),'bmp'));
   end
end

m=1;
for i=1:6
   for j=1:3
      Red(:,:,m)=exampleImage(1:92,:,1,i,j);
      m=m+1;
   end
end



figure
subplot(361)
imagesc(Red(:,:,1)),colormap('gray')
subplot(362)
imagesc(Red(:,:,4)),colormap('gray')
subplot(363)
imagesc(Red(:,:,7)),colormap('gray')
subplot(364)
imagesc(Red(:,:,10)),colormap('gray')
subplot(365)
imagesc(Red(:,:,13)),colormap('gray')
subplot(366)
imagesc(Red(:,:,16)),colormap('gray')
```

```
subplot(367)
imagesc(Red(:,:,2)),colormap('gray')
subplot(368)
imagesc(Red(:,:,5)),colormap('gray')
subplot(369)
imagesc(Red(:,:,8)),colormap('gray')
subplot(3,6,10)
imagesc(Red(:,:,11)),colormap('gray')
subplot(3,6,11)
imagesc(Red(:,:,14)),colormap('gray')
subplot(3,6,12)
imagesc(Red(:,:,17)),colormap('gray')
subplot(3,6,13)
imagesc(Red(:,:,3)),colormap('gray')
subplot(3,6,14)
imagesc(Red(:,:,6)),colormap('gray')
subplot(3,6,15)
imagesc(Red(:,:,9)),colormap('gray')
subplot(3,6,16)
imagesc(Red(:,:,12)),colormap('gray')
subplot(3,6,17)
imagesc(Red(:,:,15)),colormap('gray')
subplot(3,6,18)
imagesc(Red(:,:,18)),colormap('gray')


ort=zeros(92,92);
for m=1:18
   ort=ort+Red(:,:,m);
end
ort=ort/m;

figure
imagesc(ort(:,:)),colormap('gray');
```

```
Red1=[];
for m=1:18
   Red1(:,:,m)=Red(:,:,m)-ort;
end



A=[];
for m=1:18
   for k=1:92
      for i=1:92
      t=92*k-92;
      A(i+t,m)=Red1(i,k,m);
      end
   end
end

L=A'*A;
[U,Lmd]=eig(L);

% L_l=U*Lmd*inv(U);
% U_C=A*U;

for l=1:18
   eigen(:,:,l)=zeros(92,92);
   for n=1:18
   eigen(:,:,l)=eigen(:,:,l)+U(l,n)*Red1(:,:,n);
   end
end



%xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

orgIm1=double(imread('C:\MATLAB6p5\work\s1\5.bmp'));
orgIm=orgIm1(1:92,:,:);
orgIm1X=uint8(orgIm1);

orgIm_R(:,:)=orgIm(:,:,1);
```

```
w=[];
for k=1:18
    w(:,:,k)=eigen(:,:,k).' * (orgIm_R - ort);
end


omega=[];
for m=1:18
    for k=1:92
        for i=1:92
        t=92*m-92;
        omega(i+t,k)=w(i,k,m);
        end
    end
end


W=[];
for m=1:18
    for k=1:18
    W(:,:,k,m)=eigen(:,:,k).' * Red1(:,:,m);
end
end



Omg=[];
for m=1:18
    for k=1:18
        for j=1:92
            for i=1:92
            t=92*k-92;
            Omg(i+t,j,m)=W(i,j,k,m);
            end
        end
    end
end
```

```
eps=[];
for m=1:18
    eps(m)=norm(omega-Omg(:,:,m))/norm(Omg(:,:,m));
end

figure
stem(eps);

   mn=min(eps);
if mn<1
   KN=find(eps==mn);
   bolum = int8(KN / 3);
   bolum=double(bolum);
   kalan= KN - (bolum * 3);
   bolum=bolum + 1;

     if kalan==0
         kalan = kalan+3;
         bolum = bolum-1;
     end

     disp(strcat('The person searched is s',num2str(bolum),'\',num2str(kalan)));
     im=imread(strcat('C:\MATLAB6p5\work\s',num2str(bolum),'\',num2str(kalan),'.bmp'),'bmp');

  else
     disp('The person searched was not found')
end



figure
subplot(121)
imshow(orgIm1X,256),title('input image')
subplot(122)
imshow(im,256), title('output image')
```

```
figure
subplot(361)
imagesc(eigen(:,:,1)),colormap('gray')
subplot(362)
imagesc(eigen(:,:,2)),colormap('gray')
subplot(363)
imagesc(eigen(:,:,3)),colormap('gray')
subplot(364)
imagesc(eigen(:,:,4)),colormap('gray')
subplot(365)
imagesc(eigen(:,:,5)),colormap('gray')
subplot(366)
imagesc(eigen(:,:,6)),colormap('gray')
subplot(367)
imagesc(eigen(:,:,7)),colormap('gray')
subplot(368)
imagesc(eigen(:,:,8)),colormap('gray')
subplot(369)
imagesc(eigen(:,:,9)),colormap('gray')
subplot(3,6,10)
imagesc(eigen(:,:,10)),colormap('gray')
subplot(3,6,11)
imagesc(eigen(:,:,11)),colormap('gray')
subplot(3,6,12)
imagesc(eigen(:,:,12)),colormap('gray')
subplot(3,6,13)
imagesc(eigen(:,:,13)),colormap('gray')
subplot(3,6,14)
imagesc(eigen(:,:,14)),colormap('gray')
subplot(3,6,15)
imagesc(eigen(:,:,15)),colormap('gray')
subplot(3,6,16)
imagesc(eigen(:,:,16)),colormap('gray')
subplot(3,6,17)
imagesc(eigen(:,:,17)),colormap('gray')
subplot(3,6,18)
imagesc(eigen(:,:,18)),colormap('gray')
```

9.2 Orthonormal-Face recognition technique

```
clear;
clc;

%Yuzlerin matrislere donusturulmesi

X=6;
Y=9;
%X kisi sayisi, Y ise bir kisiye ait resim sayisi


for i=1:X
   for j=1:Y
   exampleImage(:,:,:,i,j)=
double(imread(strcat('C:\MATLAB6p5\work\s',num2str(i),'\',num2str(j),'.bmp'),'bmp'));
   end
end


%Matrisler gri ton oldugu icin renklerden (R,G,B) birini seciyoruz

m=1;
for i=1:X
   for j=1:Y
      Red(:,:,m)=exampleImage(1:92,:,1,i,j);
      m=m+1;
   end
end
```

```
%Gramm Schmidt teoremi ile dik-yuzlerin bulunmasi

for i=1:(X*Y)
    total=zeros([92 92]);



    if i >= 2
        for k=1:(i-1)
            total = total + trace( Red(:,:,i).' * ort_Red(:,:,k)) * ort_Red(:,:,k);
        end

    else
    end
    ort_Red(:,:,i) = Red(:,:,i) - total;
    ort_Red(:,:,i) = ort_Red(:,:,i) / sqrt(trace( ort_Red(:,:,i).' * ort_Red(:,:,i)));

end



%Menunun olusturulmasi

secenek=0;
olasilik=4;



while secenek~=olasilik,
secenek=menu('Yuz Tanima Sistemi','Kimlik tespiti yapilacak resimi secin','Yuz Tanima','Dik-yuzleri Goster','Cikis');
    %----------------
    if secenek==1,
        clc;
        [ad,yol]=uigetfile('*.*','Kimlik tespiti yapilacak resimi secin');
        if ad~=0
            orgIm1=double(imread(strcat(yol,ad)));
            orgIm=orgIm1(1:92,:,:);
            orgImX=uint8(orgIm);
            imshow(orgImX);
```

```
   else
      warndlg('Resim secilemedi.',' Dikkat ')
   end
end


%---------------
if secenek==2

  %resim gri tonlu oldugu icin (R,G,B) degerlerinden biri secilir

   orgIm_R(:,:)=orgIm(:,:,1);
   orgIm_R=double(orgIm_R);


  %resimin dik-yuzler uzerine izdusumleri hesaplaniyor

   for i=1:(X*Y)
      r(i)=trace( orgIm_R(:,:).' * ort_Red(:,:,i));
   end


  %dik-yuzlerden elde edilen yeni resimin olusturulmasi

    newIm_R= zeros([92 92]);


  for i=1:(X*Y)
       newIm_R(:,:)= newIm_R(:,:) + r(i)*ort_Red(:,:,i);
  end


  newIm(:,:,1)= newIm_R(:,:);
  newIm(:,:,2)= newIm_R(:,:);
  newIm(:,:,3)= newIm_R(:,:);

  newImX=uint8(newIm);
```

```
    figure
    subplot(131)
    imshow(orgImX,256)

    subplot(132)
    imshow(newImX,256)



    farkIm=orgIm - newIm;
    farkImX= uint8(farkIm);

%=============================================================

%Thresholdun belirtilmesi

    fark1=abs(farkIm(:,:,:));
    tot=0;
    for i=1:92
       for j=1:92
          tot=tot + fark1(i,j,1);
       end
    end

    tot=tot / (92*92)

    for k=1:(X*Y)
       tot1(k)=0;
       FARK(:,:,k)= newIm(:,:,1)-Red(:,:,k);
       FARK=abs(FARK);
       for i=1:92
          for j=1:92
             tot1(k)=tot1(k) + FARK(i,j,k);
          end
       end
    end
```

```
tot1 = tot1 / (92*92);
   mn = min(tot1)


   if sqrt(abs(mn^2 - tot^2))<=12.92
     KN=find(tot1(:)==mn)
     bolum = int8(KN / 9);
     bolum=double(bolum);
     kalan= KN - (bolum * 9);
     bolum=bolum + 1;

     if kalan==0
        kalan = kalan+9;
        bolum = bolum-1;
     end

     disp(strcat('The person searched is s',num2str(bolum),'\',num2str(kalan)));
     im=imread(strcat('C:\MATLAB6p5\work\s',num2str(bolum),'\',num2str(kalan),'.bmp'),'bmp');

     subplot(133)
     imshow(im,256)

   else
     disp('The person searched was not found')
   end
 end


 %----------------

 if secenek==3,

   %Dik-yuzlerin bir kismi gosteriliyor

   figure
   subplot(331)
   imagesc(ort_Red(:,:,1)),colormap('gray')
```

```
    subplot(332)
    imagesc(ort_Green(:,:,2)),colormap('gray')
    subplot(333)
    imagesc(ort_Blue(:,:,3)),colormap('gray')
    subplot(334)
    imagesc(ort_Red(:,:,4)),colormap('gray')
    subplot(335)
    imagesc(ort_Green(:,:,5)),colormap('gray')
    subplot(336)
    imagesc(ort_Blue(:,:,6)),colormap('gray')
    subplot(337)
    imagesc(ort_Red(:,:,7)),colormap('gray')
    subplot(338)
    imagesc(ort_Green(:,:,8)),colormap('gray')
    subplot(339)
    imagesc(ort_Blue(:,:,9)),colormap('gray')
  end


  %----------------
end
```

REFERENCES

[1] Rafael C. GONZALEZ
Digital Image Processing 2nd Edition

[2] Aydın AKAN
Optimization Methods

[3] Matthew A. TURK and Alex P. PENTLAND
Face Recognition Using Eigenfaces

[4] www-users.cs.york.ac.uk

[5] www.uk.research.att.com/facedatabase.html

[6] Matthew A. TURK and Alex P. PENTLAND.
Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[7] Feret Dataset. National Institute of Standards & Technology.

[8] L. Sirovich and M. Kirby
Low-dimensional procedure for the characterization of human faces

[9] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng.
Discriminant Analysis of Principal components for Face Recognition.
In *Face Recognition: From Theory to Applications*. Springer-Verlag, 1998.

[10] Evaluation of Face Recognition Algorithms. Colorado State University.
http://www.cs.colostate.edu/evalfacerec/data/normalization.html

[11] Peter Belhumeur et. Al. Eigenfaces vs Fisherfaces: Recognition Using Class Specific
Linear Projection. Vol 19. no 7. July 1997