FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

in

Electronics & Instrumentation Engineering

By

S.DINESH KUMAR
ROLL NO : 10407001

CHITTARANJAN SWAIN
ROLL NO : 10407019

N

FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & INSTRUMENTATION ENGINEERING

By

S.DINESH KUMAR
ROLL NO : 10407001

CHITTARANJAN SWAIN
ROLL NO : 10407019

UNDER THE GUIDANCE OF
PROF. S.MEHER

Department of Electronics & Communication Engineering
National Institute of Technology, Rourkela
Rourkela, Orissa – 769008
2008

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that the thesis titled, "FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS" submitted by S.DINESH KUMAR (Roll No : 10407001), CHITTARANJAN SWAIN (Roll No : 10407019) in partial fulfillment for the award of Bachelor of Technology degree in Electronics and Instrumentation Engineering ,National Institute of Technology, Rourkela(deemed university) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, this matter embodied in the thesis has not been submitted at any other university / institute for the award of any Degree or Diploma.

Professor S.Meher
Department of Electronics &Communication Engineering
National Institute of Technology
Rourkela – 769008.

ACKNOWLEDGEMENT

We take this opportunity as a privilege to thank all individuals without whose support and guidance we could not have completed our project in this stipulated period of time. First and foremost we would like to express our deepest gratitude to our Project Supervisor Professor.S.Meher, Department of Electronics and Communication Engineering, for his invaluable support, guidance, motivation and encouragement through out the period this work was carried out. We would also like to thank all professors and lecturers, and members of the department of Electronics and Communication Engineering for their generous help in various ways for the completion of this thesis. We also extend our thanks to our fellow students for their friendly co-operation.

S.Dinesh Kumar | Chittaranjan Swain

Roll No – 10407001 | Roll No – 10407019

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

CONTENTS

FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS

ABSTRACT

Face Recognition is the process of identification of a person by their facial image. This technique makes it possible to use the facial images of a person to authenticate him into a secure system, for criminal identification, for passport verification,... Face recognition approaches for still images can be broadly categorized into holistic methods and feature based methods . Holistic methods use the entire raw face image as an input, whereas feature based methods extract local facial features and use their geometric and appearance properties.

This paper describes how to build a simple, yet a complete face recognition system using Principal Component Analysis, a Holistic approach. This method applies linear projection to the original image space to achieve dimensionality reduction. The system functions by projecting face images onto a feature space that spans the significant variations among known face images. The significant features known as eigenfaces do not necessarily correspond to features such as ears, eyes and noses. It provides for the ability to learn and later recognize new faces in an unsupervised manner. This method is found to be fast, relatively simple, and works well in a constrained environment.

LIST OF FIGURES:

Chapter 1

Biometric Authentication Technology

1. BIOMETRIC AUTHENTICATION TECHNOLOGY :

Biometrics is automated method of identifying a person or verifying the identity of a person based on a physiological or behavioral characteristic. Examples of physiological characteristics include hand or finger images, facial characteristics.

Biometric authentication requires comparing a registered or enrolled biometric sample (biometric template or identifier) against a newly captured biometric sample (for example, captured image during a login). During Enrollment, as shown in the picture below, a sample of the biometric trait is captured, processed by a computer, and stored for later comparison.Biometric recognition can be used in Identification mode, where the biometric system identifies a person from the entire ***enrolled*** population by searching a database for a match based solely on the biometric. Sometime identification is called "one-to-many" matching.

A system can also be used in Verification mode, where the biometric system authenticates a person's claimed identity from their previously enrolled pattern. This is also called "one-to-one" matching. In most computer access or network access environments, verification mode would be used.

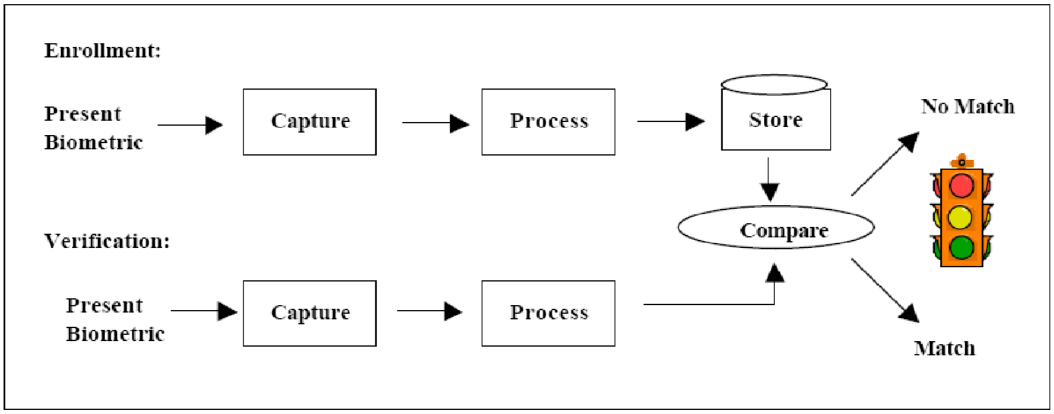


Fig 1.1 A simple process

TYPES OF BIOMETRICS : -

1.1 FACE RECOGNITION:

The identification of a person by their facial image can be done in a number of different ways such as by capturing an image of the face in the visible spectrum using an inexpensive camera or by using the infrared patterns of facial heat emission. Facial recognition in visible light typically model key features from the central portion of a facial image. Using a wide assortment of cameras, the visible light systems extract features from the captured image(s) that do not change over time while avoiding superficial features such as facial expressions or hair. Several approaches to modeling facial images in the visible spectrum are Principal Component Analysis, Local Feature Analysis, neural networks, elastic graph theory, and multi-resolution analysis.

Some of the challenges of facial recognition in the visual spectrum include reducing the impact of variable lighting and detecting a mask or photograph. Some facial recognition systems may require a stationary or posed user in order to capture the image, though many systems use a real-time process to detect a person's head and locate the face automatically. Major benefits of facial recognition are that it is non-intrusive,hands-free, continuous and accepted by most users.

1.2 FINGERPRINTS:

Fingerprints are unique for each finger of a person including identical twins. One of the most commercially available biometric technologies, fingerprint recognition devices for desktop and laptop access are now widely available from many different vendors at a low cost. With these devices, users no longer need to type passwords – instead, only a touch provides instant access. Fingerprint systems can also be used in identification mode. Several states check fingerprints for new applicants to social services benefits to ensure recipients do not fraudulently obtain benefits under fake names.

1.3 IRIS RECOGNITION:

This recognition method uses the iris of the eye, which is the colored area that surrounds the pupil. Iris patterns are thought unique. The iris patterns are obtained through a video-based image acquisition system. Iris scanning devices have been used in personal authentication applications for several years. Systems based on iris recognition have substantially decreased in price and this trend is expected to continue. The technology works well in both verification and identification modes (in systems performing one-to-many searches in a database). Current systems can be used even in the presence of eyeglasses and contact lenses. The technology is not intrusive. It does not require physical contact with a scanner. Iris recognition has been demonstrated to work with individuals from different ethnic groups and nationalities.

1.4 SIGNATURE VERIFICATION:

This technology uses the dynamic analysis of a signature to authenticate a person. The technology is based on measuring speed, pressure and angle used by the person when a signature is produced. One focus for this technology has been e-business applications and other applications where signature is an accepted method of personal authentication.

1.5 SPEAKER RECOGNITION:

Speaker recognition has a history dating back some four decades, where the outputs of several analog filters were averaged over time for matching. Speaker recognition uses the acoustic features of speech that have been found to differ between individuals. These acoustic patterns reflect both anatomy (e.g., size and shape of the throat and mouth) and learned behavioral patterns (e.g., voice pitch, speaking style). This incorporation of learned patterns into the voice templates (the latter called "voiceprints") has earned speaker recognition its classification as a "behavioral biometric." Speaker recognition systems employ three styles of spoken input: text-dependent, text-prompted and text independent. Most speaker verification applications use text-dependent input, which involves selection and enrollment of one or more voice passwords. Text-prompted input is used whenever there is concern of imposters. The various technologies used to process and store voiceprints include hidden Markov models;

pattern matching algorithms, neural networks, and matrix representation and decision trees. Some systems also use "anti-speaker" techniques, such as cohort models, and world models.

Ambient noise levels can impede both collections of the initial and subsequent voice samples. Performance degradation can result from changes in behavioral attributes of the voice and from enrollment using one telephone and verification on another telephone. Voice changes due to aging also need to be addressed by recognition systems. Many companies market speaker recognition engines, often as part of large voice processing, control and switching systems. Capture of the biometric is seen as non-invasive. The technology needs little additional hardware by using existing microphones and voice-transmission technology allowing recognition over long distances via ordinary telephones (wire line or wireless).

In this project we concentrate on face recognition approach out of these biometric approaches. For face recognition we use Principal Component Analysis or Karhunen-Loeve Transform. Description of that is given in following pages.

Chapter 2

Face Recognition using PCA

2. FACE RECOGNITION USING PCA

2.1 INTRODUCTION :

The face is our primary focus of attention in social intercourse, playing a major role in conveying identity and emotion. We can recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance 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 or changes in hairstyle or facial hair.

Computational models of face recognition, in particular, are interesting because they can contribute not only to theoretical insights but also to practical applications. Computers that recognize faces could be applied to a wide variety of problems, including criminal identification, security systems, image and film processing, and human computer interaction. Unfortunately, developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli.

The user should focus his attention toward developing a sort of early, pre attentive Pattern recognition capability that does not depend on having three-dimensional information or detailed geometry. He should develop a computational model of face recognition that is fast, reasonably simple, and accurate.

Automatically learning and later recognizing new faces is practical within this framework. Recognition under widely varying conditions is achieved by training on a limited number of characteristic views (e.g. a "straight on" view, a 45 degree view, and a profile view). The approach has advantages over other face recognition schemes in its speed and simplicity learning capacity.

Images of faces, represented as high-dimensional pixel arrays, often belong to a manifold of intrinsically low dimension. Face recognition, and computer vision research in general, has witnessed a growing interest in techniques that capitalize on this observation, and apply algebraic and statistical tools for extraction and analysis of the underlying manifold.

Eigenface is a face recognition approach that can locate and track a subject's head, and then recognize the person by comparing characteristics of the face to those of known individuals.

The computational approach taken in this system is motivated by both physiology and information theory, as well as by the practical requirements of near-real-time performance and accuracy. This approach treats the face recognition problem as an intrinsically two-dimensional (2-D) recognition problem rather then requiring recovery of three-dimensional geometry, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views.

2.2 FACE SPACE AND ITS DIMENSIONALITY :

Computer analysis of face images deals with a visual signal (light reflected of the surface of a face) that is registered by a digital sensor as an array of pixel values. The pixels may encode color or only intensity. After proper normalization and resizing to a fixed m-by-n size, the pixel array can be represented as a point (i.e. vector) in an mn-dimensional image space by simply writing its pixel values in a fixed (typically raster) order. A critical issue in the analysis of such multi-dimensional data is the dimensionality, the number of coordinate necessary to specify a data point.

2.3 IMAGE SPACE VS. FACE SPACE :

In order to specify an arbitrary image in the image space, one needs to specify every pixel value. Thus the “nominal" dimensionality of the space, dictated by the pixel representation, is mn - a very high number even for images of modest size However, much of the surface of a face is smooth and has regular texture.

Therefore, per-pixel sampling is in fact unnecessarily dense. The value of a pixel is typically highly correlated with the values of the surrounding pixels. Moreover, the appearance of faces is highly constrained; for example, any frontal view of a face is roughly symmetrical, has eyes on the sides, nose in the middle, etc. A vast proportion of the points in the image space does no represent physically possible faces.Thus, the natural constraints dictate that the face images will in fact be confined to a subspace, which is referred to as the face space.

2.4 THE PRINCIPAL MANIFOLD AND BASIS FUNCTIONS :

It is common to model the face space as a (possibly disconnected) principal manifold, embedded in the high-dimensional image space. Its intrinsic dimensionality is determined by the number of degrees of freedom within the face space, the goal of subspace analysis is to determine this number, and to extract the principal modes of the manifold. The principal modes are computed as functions of the pixel values and referred to as basis functions of the principal manifold.

To make these concepts concrete, consider a straight line in R3, passing through the origin and parallel to the vector a = [a1; a2; a3]T . Any point on the line can be described by 3 coordinates; nevertheless, the subspace that consists of all points on the line has a single degree of freedom, with the principal mode corresponding to translation along the direction of a. Consequently, representing the points in this subspace requires a single basis function:

$$\phi(x_1, x_2, x_3) = \sum_{j=1}^{3} a_j x_j .$$

The analogy here is between the line and the face space, and between R3 and the image space.

2.5 PRINCIPAL COMPONENT ANALYSIS :

Principal Component Analysis (PCA) is a dimensionality reduction technique based on extracting the desired number of principal components of the multi-dimensional data. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables.The first principal component is the linear combination of the original dimensions that has the maximum variance; the n-th principal component is the linear combination with the highest variance, subject to being orthogonal to the n -1 first principal components.

An important, and largely unsolved problem in dimensionality reduction is the choice of k- the intrinsic dimensionality of the principal manifold. No analytical derivation of this number for a complex natural visual signal is available to date. To simplify this problem, it is common to assume that in the noisy embedding of the signal of interest (in our case, a point sampled from the facespace) in a high-dimensional space, the signal-to-noise ratio is high. Statistically,that means that the variance of the data along the principal modes of the manifold is high compared to the variance within the complementary space.

This assumption relates to the eigenspectrum - the set of the eigenvalues of the data covariance matrix. The i-th eigenvalue is equal to the variance along the i-th principal component; thus, a reasonable algorithm for detecting k is to search for the location along the decreasing eigenspectrum where the value of λi drops significantly

Since the basis vectors constructed by PCA had the same dimension as the input face images, they were named “Eigenfaces".

PCA is an information theory approach of coding and decoding face images may give insight into the information content of face images, emphasizing the significant local and global "features". Such features may or may not be directly related to face features such as eyes, nose, lips, and hair.

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of face is to somehow capture the variation in a collection of images, independent of any judgment of features, and use this information to encode and compare individual face images.

These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less of each eigenvector, so that we can display the eigenvector as a sort of ghostly face which we call an ***eigenface***.

Each individual face can be represented exactly in terms of a linear combination of the eigenfaces. Each face can also be approximated using only the "best" eigenfaces-those that have the largest eigenvalues and which therefore account for the most variance within the set of face images. The best M eigenfaces span an M-Dimensional subspace- "face space" – of all possible images.

This approach of face recognition involves the following initialization operations:

1. Acquire an initial set of face images (the training set).

2.Calculate the eigenfaces from the training set, keeping only the M images that correspond to the highest eigenvalues. These M images define the face space. As new faces are experienced; the eigenfaces can be up-dated or recalculated.

3.Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting his or her face images onto the "face space".

Having initialized the system, the following steps are then used to recognize new face images:

1. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.

2. Determine if the image is a face by checking to see if the image is sufficiently close to "face space".

3. If it is a face, classify the weight pattern as either a known person or as unknown.

4. (Optional) Update the eigenfaces and/or weight patterns.

5. (Optional) If the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into the known faces.

2.6 Calculating Eigenfaces

Images of faces, being similar in overall configuration, will not be randomly distributed in the huge space and thus can be distributed by a relatively low dimensional subspace. The main idea of principal component analysis 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 square, describes an N-by-N image, and is a linear combination of original face images, and because they are face-like in appearance, we refer then to as "eigenfaces".

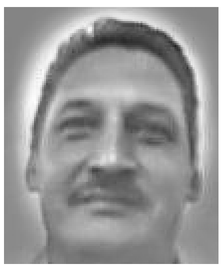
Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \ldots \Gamma_M$. The average face of the set is defined by

$$\mathbf{\Psi} = 1/M \sum_{n=1}^{M} \mathbf{\Gamma}_n$$

The Average Face $\mathbf{\Psi}$

Each face differs from the average by the vector $\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi}$.

Example of Γ



Example of Φ

An example training set is shown in Fig , with the average face Ψ shown.This set of very large vectors is then subject to principle component analysis, which seeks a set of M orthonormal vectors, U_n, which best describes the distribution of data. The *k*th vector, U_k, is chosen such that

$$\lambda_k = 1/M \sum_{n=1}^{M} (u^T{}_k \Phi_n)^2 \quad \text{………… ……………..(1)}$$

is a maximum, subject to

$$u_l\, u_k = \delta lk = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{if otherwise} \end{cases} \quad \text{………… ……….(2)}$$

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

$$C = 1/M \sum \Phi_n \Phi^T{}_n \qquad \text{………… …………….. (3)}$$

$$= AA^T$$

where the matrix $A = [\Phi_1\ \Phi_2\ \Phi_3 \ldots \Phi_M]$. The matrix C, however, is N^2 by N^2 , and determining the N square 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 then 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). Fortunately we can solve for the N2 dimensional eigenvectors in case by first solving for the eigenvectors of an M-by-M matrix – e.g., solving a 16 x 16 matrix rather than a 16,384 x 16,384 matrix and then taking appropriate

linear combinations of the face images. Consider the eigenvectors vi of A^TA such that

$$A^TA\, v_i = \mu_i v_i \quad \text{………… ……………..(4)}$$

Premultiplying both side by A we have

$$AA^TAv_i = \mu_i Av_i \quad \text{… ……………..(5)}$$

from which we see that Av_i are the eigenvectors of

$$C = AA^T\ .$$

Following this analysis, we can construct the M by 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 from the eigenfaces u_l.

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1,\ldots\ldots.,M \text{ …… …………….. (6)}$$

With this analysis the calculations are greatly reduced, from the order of the number of pixels in images (N2) to the order of the number image in the training set (M).The associated eigenvalues allow us to rank to eigenvectors according to their usefulness in characterizing the variation among the images.

Fig 2.1 Example of Face DataBase of 2 persons

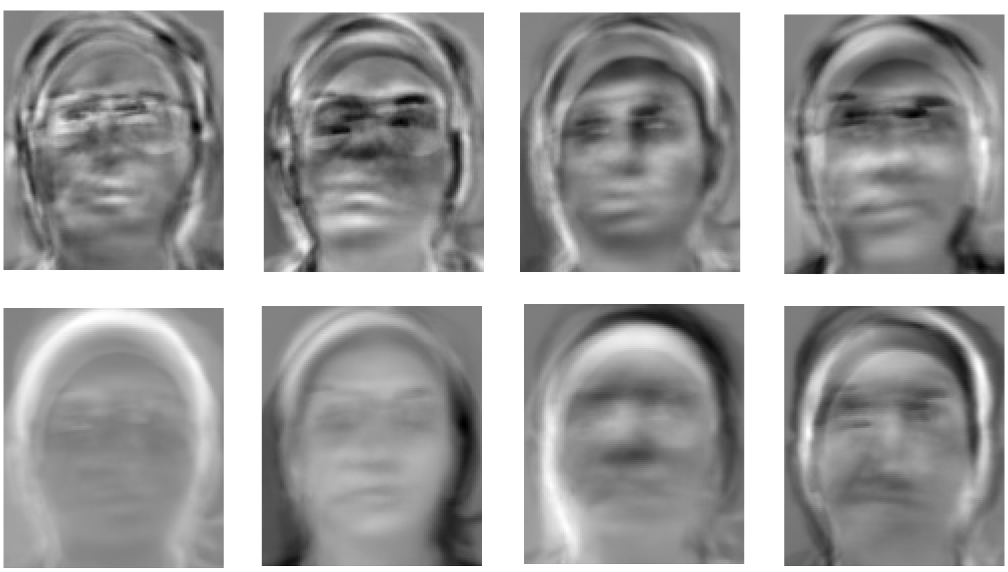


Fig 2.2 Example of eigen faces

2.7 USING EIGENFACES TO CLASSIFY A FACE IMAGE:

A new face image ($\mathbf{\Gamma}$) is transformed into its eigenface components (projected into "face space") by a simple operation,

$$\omega = u^T_k (\mathbf{\Gamma} - \mathbf{\Psi}) \quad \ldots\ldots\ldots\ldots\ldots..(7)$$

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 and its processing hardware.

The weights form a vector $\Omega^T = [\omega_1, \omega_2 \ldots.. \omega_{M'}]$ that describes the contribution of each eigenface in representing the input face image, treating the eigenface as a basis set for face images. The vector may be used in a standard pattern recognition algorithm to find which of a number of predefined classes, if any best describes the face. The simplest method for determining of an input face image is to find the face class k that minimizes the Euclidian distance

$$\varepsilon^2_k = ||(\Omega - \Omega_k)||^2 \quad \ldots\ldots\ldots\ldots\ldots..(8)$$

Where Ω_k is a vector describing the kth face class. The face classes Ω_I are calculated by averaging the results of the eigenface representation over a small number of face images of each individual. A face is classified as belonging to class k when minimum ε_k is below some chosen threshold θ_ε. Otherwise the face is classified as "unknown" and optionally creates a new face class.

Because creating the vector of weights is equivalent to projecting the original face image onto the low dimensional face space, many images will project onto a given pattern vector. The distance between the image and the face space is simply the squared distance between the mean-adjusted input image $\Phi = \Gamma - \Psi$ and $\Phi_f = \Sigma^{M'}_{i=1} \omega_i u_i$ its projection onto face space:

$$\varepsilon^2 = ||\Phi - \Phi_f||^2 \ldots\ldots \ldots\ldots\ldots\ldots..(9)$$

Thus there are four possibilities for an input image and pattern vector:

Near face space and near face class,

Near face space but not near a known face class,

Distant from face space and near a face class, and

Distant from face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is present. The last two cases indicate that the image is not a face image.

2.8 SUMMARY OF EIGENFACE RECOGNITION

To summarize the eigenfaces approach to face recognition involves the following steps:

- Collect a set of characteristic face images of the known individuals. This set should include a number of images of each person, with some variation in expression and in the lighting. (Say four images of ten people, so M = 40)
- Calculate the (40 x 40) matrix L, find its eigenvalues and eigenvectors, and choose the M' eigenvectors with the highest associated eigenvalues. (Let M' = 10 in example).
- Combine the normalized set of images according to equation (6) to produce the (M' = 10) eigenfaces u_k.
- For each known individual, calculate the class vector Ω_k by averaging pattern vector Ω[from Eq. (8)] calculated from the original (four) images of the individual. Choose a threshold θ_ε that defines the maximum allowable distance from any face class, and a threshold θ_ε that defines the maximum allowable distance from face space.
- For each new image to be identified, calculate its pattern vector Ω, the distance ε_I to each known class, and the distance ε to face space. If the minimum distance $\varepsilon_k < \theta_\varepsilon$ and the distance $\varepsilon < \theta_\varepsilon$, classify the input face as the individual associated with class vector Ω_k. If the minimum distance $\varepsilon_k = \theta_\varepsilon$ but distance $\varepsilon < \theta_\varepsilon$, then the image may be classify as "unknown", and optionally used to begin a new face class.
- If new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated. This gives the opportunity to modify the face space as the system encounters more instances of the known faces.

Chapter 3

Implementation

3.1 IMPLEMENTATION OF "FACE RECOGNITION" USING PCA IN MATLAB :

The entire sequence of training and testing is sequential and can be broadly classified as consisting of following two steps:

1. Database Preparation
2. Training
3. Testing

The steps are shown below.

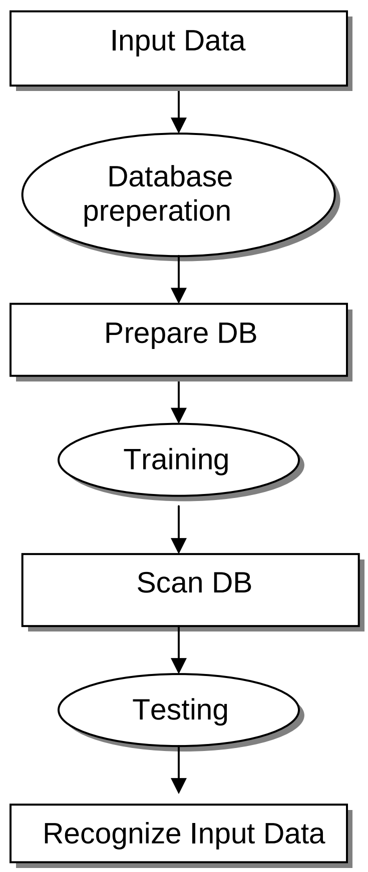


Fig 3.1 Flowchart indicating the sequence of implementation

3.1.1DATABASE PREPARATION :-

The database was obtained with 15 photographs of each persons at different viewing angels and different expressions. There are 28 persons in database1.The Database is kept in the train folder which contains subfolders for each person having all his/her photographs.

Database was also prepared for testing phase by taking 4-5 photographs of 10 persons in different expressions and viewing angles but in similar conditions (such as lighting, background, distance from camera etc.) using a low resolution camera.And these images were stored in test folder.

3.1.2 TRAINING :-

1. Select any one (.bmp) file from train database using open file dialog box.
2. By using that read all the faces of each person in train folder.
3. Normalize all the faces.
4. Find significant Eigenvectors of Reduced Covariance Matrix.
5. Hence calculate the Eigenvectors of Covariance Matrix.
6. Calculate Recognizing Pattern Vectors for each image and average RPV for each person
7. For each person calculate the maximum out of the distances of all his imageRPVs from average RPV of that person.

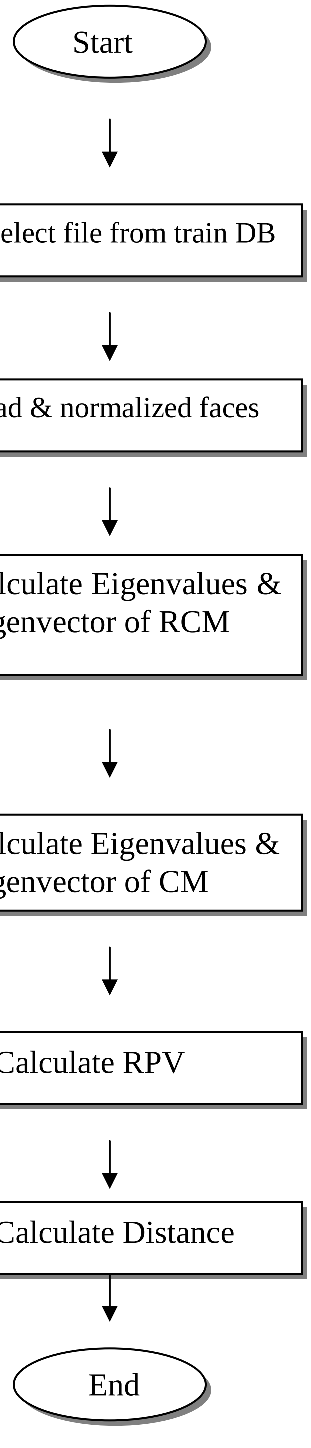


Fig 3.2 Flowchart for training

3.1.3 TESTING :-

Testing is carried out by following steps:

1. Select an image which is to be tested using open file dialog box.
2. Image is Read and normalize.
3. Calculate the RPV of image using Eigenvector of Covariance Matrix.
4. Find the distance of this input image RPV from average RPVs of all the persons.
5. Find the person from which the distance is minimum.
6. If this minimum distance is less than the maximum distance of that person calculated during training than the person is identified as this person.

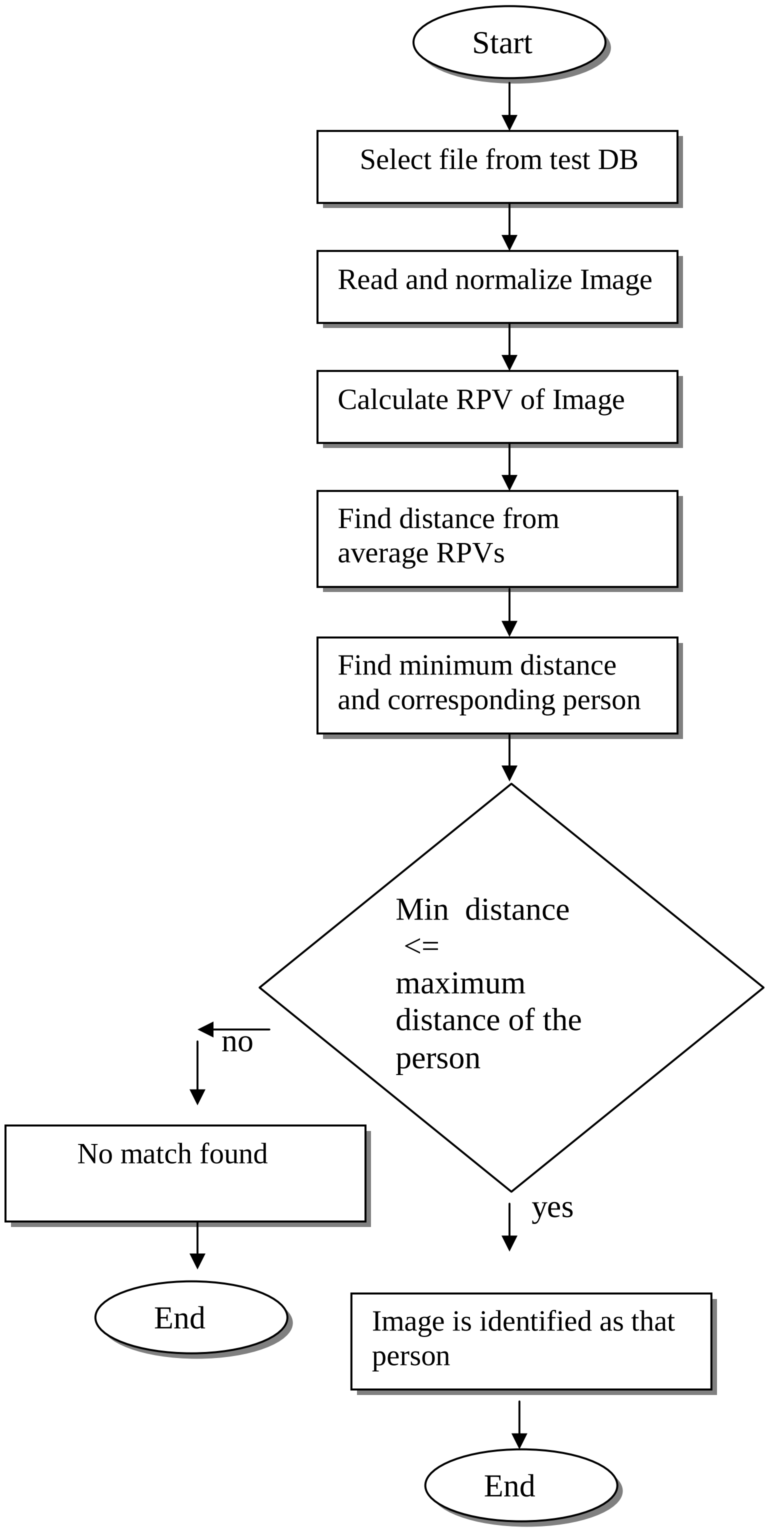


Fig 3.3 Flowchart fot testing

3.2 APPLICATION AREAS:

1. ACCESS CONTROL :

- ATM
- Airport

Fig. 4.1 A Door lock control system using face verification technology

2. ENTERTAINMENT :

- Video Game
- Virtual Reality
- Training Programs
- Human-Computer-Interaction
- Human-Robotics
- Family Photo Album

3. SMART CARDS:

- Drivers' Licenses
- Passports
- Voter Registrations
- Welfare Fraud
- Voter Registration

4. Information Security :

- TV Parental control
- Desktop Logon
- Personal Device Logon
- Database Security
- Medical Access
- Internet Access

5. Law Enforcement & Surveillance :

- Advanced Video surveillance
- Drug Trafficking
- Portal Control

6. Multimedia Management :

- Multimedia management is used in the face based database search.

7. Some Commercial Applications :

- Motion Capture for movie special effects.
- Face Recognition biometric systems.
- Home Robotics.

Chapter 4

Experimental results and conclusion

4.1 EXPERIMENTAL RESULTS OBTAINED BY VARYING THE THRESHOLD:

Inorder to check the dependence of the results on the threshold value, 2 different databases were used. The rejection and acceptance plots for both databases were plotted for their respective training data.the si value was varied from 0 to 1 and the plots were made.

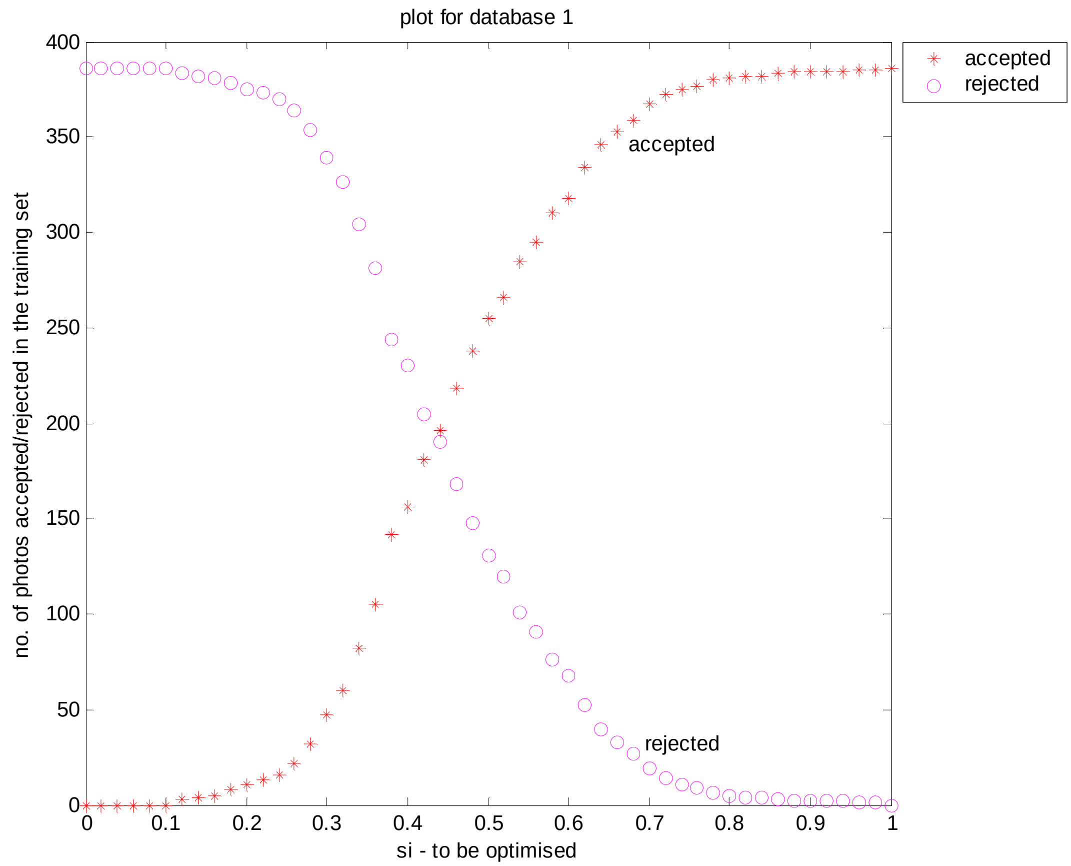


Fig 4.2 Plot for database 1

It was observed that a value of si above 0.75 resulted in very good acceptance, though the tradeoff being the acceptance of even unknown faces to the closest face.

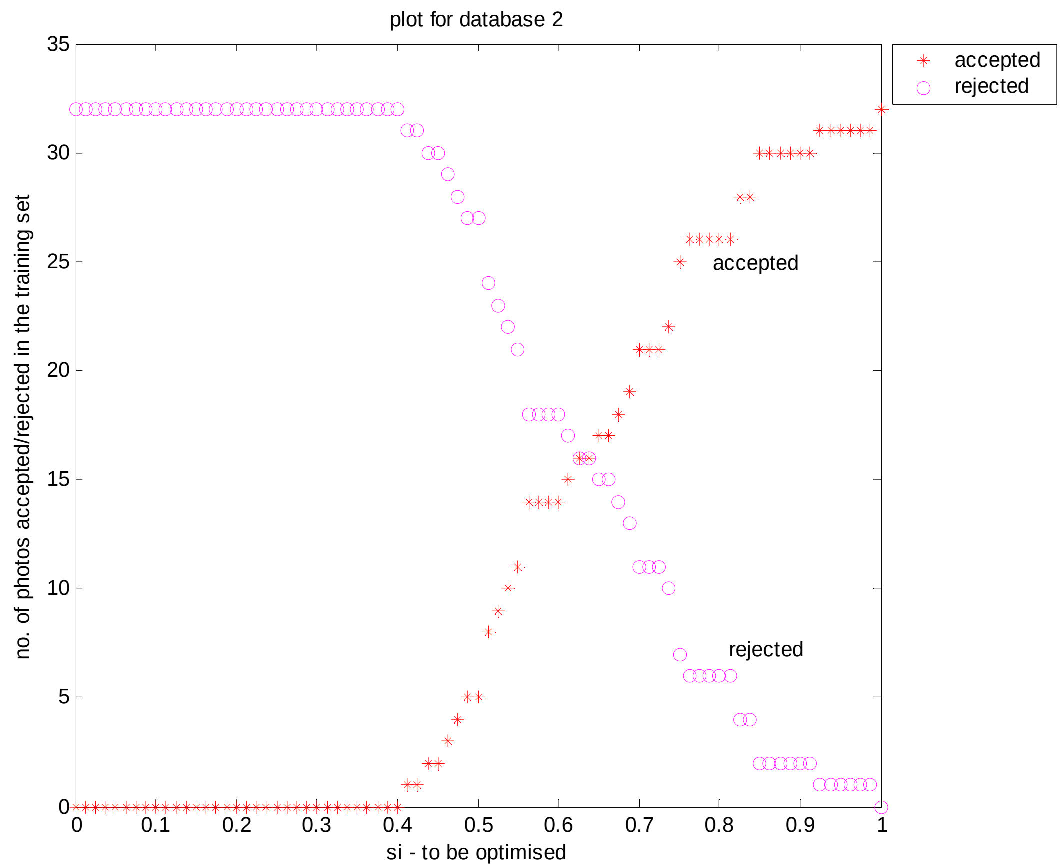


Fig 4.2 Plot for database 2

It was observed that a value of si above 0.85 resulted in very good acceptance, though the tradeoff being the acceptance of even unknown faces to the closest face.

The higher si value for db2 may be attributed to the lesser number of face samples in the database and to the quality of the face images.

4.2 LIMITATIONS OF THE ALGORITHM :

Success of a practical face recognition system with images grabbed live depends on its robustness against the inadvertent and inevitable data variations. Specifically, the important issues involved are

- Facial size normalization
- Non-frontal view of the face (3D pose, head movement)
- Tolerance to facial expression / appearance (including facial hair & specs)
- Invariance to lighting conditions (including indoor / outdoor)
- Facial occlusion (sunglasses, hat, scarf, etc.)
- Invariance to aging.

We tried to minimize the data variations by capturing the facial image subjected to the reasonably constrained environment –

- Frontal view geometry and
- Controlled lighting.

4.3 FUTURE SCOPE:

This project is based on eigenface approach that gives an accuracy maximum of about 92.5%. Adaptive algorithms may be used to obtain an optimum threshold value. There is scope for future betterment of the algorithm by using Neural Network technique that can give better results as compared to eigenface approach. With the help of neural network technique accuracy can be improved.

Instead of having a constant threshold, it could be made adaptive,depending upon the conditions and the database available, so as to maximise the accuracy.The whole software is dependent on the database and the database is dependent on resolution of camera. So if good resolution digital camera or good resolution analog camera is used , the results could be considerably improved.

4.4 CONCLUSION:

Many methods of making computers recognize faces were limited by the use of improvished face models and feature descriptions(matshing simple distances), assuming that a face is no more than the sum of its parts,the individual features. They tend to hide much of the pertinent information in the weights thar makes it difficult to modify and evaluate parts of the approach.

This particular method using Principal Component Analysis for face recognition was motivated by information theory, leading to basing face recognition on a small set of image features that best approximates the set of known face images,without regarding that they correspond to our intutive notions of facial parts and features.

The eigenface approach provides a practical solution that is well fitted for the problem of face recognition. It is fast, relatively simple, and works well in a constrained environment. Certain issues of robustness to changes in lighting, head size, and head orientation , the tradeoffs between the number of eigenfaces necessary for unambiguous classification are matter of concern.

REFERENCES:

1. Matthew Turk and Alex Pentland " Eigenfaces for Recognition". vision and Modeling Group , The Media Laboratory , Massachusetts institute of Technology.

2. Fernando L. Podio and Jeffrey S. Dunn[2] "Biometric Authentication Technology ".

3. Matthew Turk "A Random Walk through Eigenspace" IEICE Trans Dec 2001

4. Stan Z.Li & Anil K. Jain "Handbook of Face Recognition" Springer publications

5. http://www.face-rec.org

6. http://www.alglib.net

7. http://math.fullerton.edu/mathews/n2003/JacobiMethodProg.html

APPENDIX 1:

Matlab Code [Face Recognition using PCA]:

TRAINING:

```
clc;
clear all; % Remove items from workspace, freeing up system memory
close all;
global  db1;    %trained face database loaded in RAM
global  db_max;
db1=[];

person.name = [];   %name of persons in DB
person.path = []; %location = path name containing face data
person.desired =[];
person.obtained = [];

[f, p1]=uigetfile('*.*','Select any file from DB');

tic;  %start counting time to estimate how long the training takes
msg1 = '*                The system is learning                *';
msg_handle = msgbox( msg1, 'Please Wait                ...' );

%finding the extension of the selected file. Assumes that all files in the
%face data base will have the same file extension.
ext = find_file_ext( f );
```

ACCESSING THE WHOLE DATABASE:

```
%finding directories (=persons) in the DB
```

```
op = pwd;   %store old path
cd( p1 );   %p1 contains the paths of the clicked file in the face DB
cd ..       %go a level higher = root of the face DB
p2 = pwd;   %p2 stores the root of the DB
cd( op );   %restore the original path
d = dir( p2 ); %take directory of the face DB to know the persons in the DB
[s x] = size( d );
j = 1;      %j counts the no of persons (= no folders at this level)
for i = 3:s %1st 2 folders are . & .. i.e. current & parent directory
  if( d(i).isdir == 1 )
    name = d(i).name;
    person(j).name = name;
    l = size( name );
    person(j).name_length = l( 2 );
    name1 = strcat( p2, '\' );
    name1 = strcat( name1, name );
    name1 = strcat( name1, '\' );
    person(j).path = name1;
    j = j + 1;
  end
end
persons = j - 1;

disp( 'Scanned face DB' );

%read faces from disk DB to DB in RAM i.e. fdb
fdb = [];
nof = 0;  %total no of faces in DB = sum of all the faces of all the persons
          %nof = no of columns in fdb after reading the whole DB
for i = 1:persons %there r 'persons' no of persons in the DB
  p2 = [ person(i).path ext ];
```

```
  d = dir( p2 );   %d = n by 1 struct; n = no of faces
  faces = size( d );  %faces = no of faces of ith person in DB
  for j = 1:faces
    p2 = [ person(i).path   d(j).name ];
    im = imread( p2 );
    s = size(im);
    l = length( s );
    if( l == 3 ) %3=>colour picture
      for x = 1:s(1,1)  % along x axis of the image
        for y = 1:s(1,2)% along y axis of the image
          c(x,y) = (double(im(x,y,1))+double(im(x,y,2))+double(im(x,y,3)))/3;
        end             % c = colour intensity at x,y
      end
    else
      c = im;
    end
    t = double( ( c(:) ) );

    person(i).fd{j} = t;
    fdb = [fdb t];
    nof = nof + 1;
  end
  person(i).faces = faces;  %faces = no of faces of ith person in DB
end
dfdb = [];
average_face = mean(fdb, 2);

for i = 1:nof
%  dfdb = [ dfdb double(fdb(:,i:i))-average_face ];
  dfdb = [ dfdb ( fdb( :, i:i ) ) - average_face ];
end
```

```
%clear fdb;
dfdb = dfdb / sqrt( nof );

%actual correlation matrix = cm = dfdb * dfdb' -> very big matrix
%calculate dfdb' * dfdb = reduced cm -> much smaller matrix
%reduced cm = rcm -> nof by nof matrix
rcm = dfdb' * dfdb;  %rcm = reduced correlation matrix
[rvector, rvalue] = eig( rcm ); %eigen vector & eigen value of rcm
%clear rcm;

disp( 'Calculated Eigen Vectors & Eigen Values of Reduced CM' );

%normalise eigen vectors of rcm
for i = 1:persons
  t1 = rvector( :, i );
  s1 = sqrt( sum( t1.^2 ) );
  rvector( :, i ) = rvector( :, i )./s1;
end

%calculate eigen vector & eigen value of cm = evector & evalue
evector = dfdb * rvector;
evalue = diag( rvalue );

%clear rvector rvalue;
disp( 'Calculated Eigen Vectors & Eigen Values of CM' );




[sorted_evalue, index] = sort( evalue ); %sorted in acending order
sorted_evalue = flipud( sorted_evalue );  %rearranged in decending order
```

```
index = flipud( index );  %rearranged corresponding indies also

%Now rearrange eigenvectors in the order of rearranged eigen values
evector( :, 1:nof ) = evector ( :, index );



smallest_ev1p = 0.01 * sorted_evalue(1);% to obtain the effective eigen values
for i = 1:nof
  if( sorted_evalue(i) < smallest_ev1p )
    break;
  end
end
index = i;

%keep only as many eigen vectors as obtained by above cutoff criterion
evector = evector( :, 1:index );
evalue = sorted_evalue( 1:index );

%normalise eigen vectors of cm
for i = 1:index
  t1 = evector( :, i );
  s1 = sqrt( sum( t1.^2 ) );
    evector( :, i ) = evector( :, i )./s1;
end
```

CALCULATING RPV:

```
s = size( average_face );
for i = 1:persons
  p2 = [ person(i).path ext ];
  d = dir( p2 );
```

```
  sum_rpv = zeros( index, 1 );  %rpv = recognising pattern vector

  for j = 1:person(i).faces
    t = person(i).fd{j} - average_face;
    rpv = evector' * t;      %rpv = recognising pattern vector
    rpv = rpv / s(1,1);   %so that nos. don't grow too much
    person(i).rpv{j} = rpv;
    person(i).ido{j} = i;
    sum_rpv = sum_rpv + rpv;
  end
  person(i).average_rpv{1} = sum_rpv / person(i).faces(1,1);
  distance = zeros( person(i).faces(1, 1), 1);
  for j = 1:person(i).faces
     p2 = [ person(i).path   d(j).name ];
     distance( j, 1 ) = norm(person(i).rpv{j} -  person(i).average_rpv{1} );
  end
  [min_dist, index_dist ] = min( distance );
  [max_dist , index_dist ] = max (distance);
  person(i).min_dist = min_dist;
  person(i).max_dist = max_dist;
end

for k= 1:persons
   minmum(k) = person(k).min_dist;
   maxmum(k) = person(k).max_dist;
end
db_min=min(minmum);
db_max=max(maxmum);



disp( 'Calculated Recognising Pattern Vectors' );
```

```
db1.person = person;
db1.persons = persons;
db1.average_face = average_face;
db1.evector = evector;
save('Face_DB.mat', 'db1', '-mat');%saving in excel format



close( msg_handle );  %close dlg box showing "Learning..."
t = toc;  %stop counting time to estimate how long the training takes
t = etms( t );  %BPD's fn calcutales elasped time in min & sec
m = int2str( t(1,1) ); %minutes
s = int2str( t(1,2) );  %seconds
ms = 'Training Completed Successfully in  ';
ms = [ms   m '   Minutes and   ' s  '  Seconds.'];
msgbox( ms, 'Completed' );
```

TESTING:

```
%  test M-file read an image ,if it is colored then take the average of the
%  three bands Red, Blue n Green & then change that in a column matrix.If it
%  is Black & white then changes directly in column matrix.
%  It finds the rpv & with the help of rpv it finds Distance and minimum
%  distance.

global  db1;        %trained face database loaded in RAM

load Face_DB.mat;
%select the desired file to be tested
[f, p1]=uigetfile('*.*','Select a file for Testing');
```

```
file_name = strcat( p1, f );



% Read the image.
im = imread( file_name );
  s = size(im);      % returns the sizes of each dimension of im.
  l = length( s );   % returns the size of the longest dimension of s.
  if( l == 3 )       % 3=>colour picture.
    for x = 1:s(1,1)  % along x axis of the image.
      for y = 1:s(1,2)% along y axis of the image.
          c(x,y) = (double(im(x,y,1))+double(im(x,y,2))+double(im(x,y,3)))/3;
      end           % c = colour intensity at x,y
    end
  else
    c = im;
  end
  t = ( c(:) );      % convert matrix 'c' in column matrix.

s = size( db1.average_face );
f = double(t) - db1.average_face;  %f = difference of the given face from average.
x_rpv = (db1.evector)' * f;        %recognising pattern vector.
x_rpv = x_rpv / s(1,1);            %so that nos. don't grow too much.
distance = zeros( db1.persons, 1 ); % for dist of given face from faces in DB.
for i = 1:db1.persons
 distance( i, 1 ) = norm( x_rpv - db1.person(i).average_rpv{1} );
end
[ min_dist, index ] = min( distance );  % minimum distance in all dstances.

r.index = index;
r.min_dist = min_dist;
r.rpv = x_rpv;
```

```
if(r.min_dist > db_max)
    title = ' UNKNOWN FACE';
    msg = [title '  - not in database '  ];
    h = message_window( title, msg );
else
    title = 'Closest Match';
    msg = [title '  =  '  db1.person(r.index).name];
    h = message_window( title, msg );
end
```