Face Recognition Using Principal Component Analysis Method

Liton Chandra Paul[1] , *Abdulla Al Sumam*[2]
Electronics and Telecommunication Engineering[1,2]
Rajshahi University of Engineering and Technology

Abstract— ***This paper mainly addresses the building of face recognition system by using Principal Component Analysis (PCA). PCA is a statistical approach used for reducing the number of variables in face recognition. In PCA, every image in the training set is represented as a linear combination of weighted eigenvectors called eigenfaces. These eigenvectors are obtained from covariance matrix of a training image set. The weights are found out after selecting a set of most relevant Eigenfaces. Recognition is performed by projecting a test image onto the subspace spanned by the eigenfaces and then classification is done by measuring minimum Euclidean distance. A number of experiments were done to evaluate the performance of the face recognition system. In this thesis, we used a training database of students of Electronics and Telecommunication Engineering department, Batch-2007, Rajshahi University of Engineering and Technology, Bangladesh.***

Index Terms—PCA, Eigenvalue, Eigenvector, Covariance, Euclidean distance, Eigenface.

I. INTRODUCTION

Over the last ten years or so, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but neuroscientists and psychologists also. It is the general opinion that advances in computer vision research will provide useful insights to neuroscientists and psychologists into how human brain works, and vice versa [1].The goal is to implement the system (model) for a particular face and distinguish it from a large number of stored faces with some real-time variations as well. It gives us efficient way to find the lower dimensional space. Further this algorithm can be extended to recognize the gender of a person or to interpret the facial expression of a person. Recognition could be carried out under widely varying conditions like frontal view, a 45° view, scaled frontal view, subjects with spectacles etc are tried, while the training data set covers limited views. The algorithm models the real-time varying lighting conditions as well. But this is out of scope of the current implementation. The aim of this research paper is to study and develop an efficient MATLAB program for face recognition using principal component analysis and to perform test for program optimization and accuracy. This approach is preferred due to its simplicity, speed and learning capability [2].

II. FACE RECOGNITION PROCESS

One of the simplest and most effective PCA approaches used in face recognition systems is the so-called eigenface approach. This approach transforms faces into a small set of essential characteristics, eigenfaces, which are the main components of the initial set of learning images (training set). Recognition is done by projecting a new image in the eigenface subspace, after which the person is classified by comparing its position in eigenface space with the position of known individuals [3]. The advantage of this approach over other face recognition systems is in its simplicity, speed and insensitivity to small or gradual changes on the face. The problem is limited to files that can be used to recognize the face. Namely, the images must be vertical frontal views of human faces. The whole recognition process involves two steps:

A. Initialization process
B. Recognition process

The Initialization process involves the following operations:

i. Acquire the initial set of face images called as training set.
ii. Calculate the Eigenfaces from the training set, keeping only the highest eigenvalues. These M images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.
iii. Calculate distribution in this M-dimensional space for each known person by projecting his or her face images onto this face-space.

These operations can be performed from time to time whenever there is a free excess operational capacity. This data can be cached which can be used in the further steps eliminating the overhead of re-initializing, decreasing execution time thereby increasing the performance of the entire system [4].

Having initialized the system, the next process involves the steps:

i. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the Eigenfaces.
ii. Determine if the image is a face at all (known or unknown) by checking to see if the image is sufficiently close to a “free space”.
iii. If it is a face, then classify the weight pattern as either a known person or as unknown.
iv. Update the eigenfaces or weights as either a known or unknown, if the same unknown person face is seen several times then calculate the characteristic weight

pattern and incorporate into known faces. The last step is not usually a requirement of every system and hence the steps are left optional and can be implemented as when the there is a requirement.

III. EIGENFACE ALGORITHM

Let a face image Γ(x, y) be a two dimensional M by N array of intensity values. In this thesis, I used a set of image by 200 × 149 pixels. An image may also be considered as a vector of dimension M × N, so that a typical image of size 200 × 149 becomes a vector of dimension 29,800 or equivalently a point in a 29,800 dimensional space.

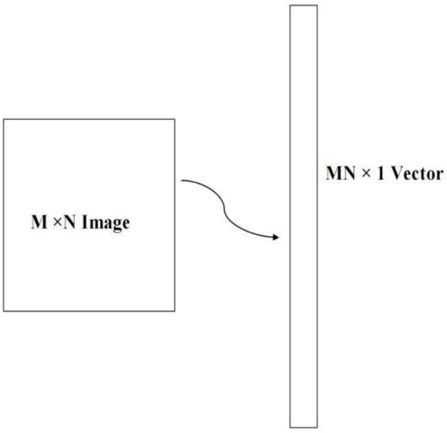


Fig-1:Conversion of M × N image into MN ×1 vector

Step1: prepare the training faces
Obtain face images $I_1, I_2, I_3, I_4, \ldots\ldots I_M$ (training faces). The face images must be centered and of the same size.

Step 2: Prepare the data set
Each face image I_i in the database is transformed into a vector and placed into a training set S.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \ldots \ldots .. \Gamma_M\}$$

In My example M = 34. Each image is transformed into a vector of size MN × 1 and placed into the set. For simplicity, the face images are assumed to be of size N × N resulting in a point in N^2dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Step 3: compute the average face vector
The average face vector (Ψ) has to be calculated by using the following formula:

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

Step 4: Subtract the average face vector
The average face vector Ψ is subtracted from the original faces Γ_i and the result stored in the variable Φ_i ,

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

Step 5: Calculate the covariance matrix
We obtain the covariance matrix C in the following manner,

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

$$= AA^T \qquad (N^2 \times N^2 \text{ matrix}$$ Where, A=$[\Phi_1, \Phi_2, \Phi_3, \Phi_4, \ldots \ldots .. \Phi_M$ $(N^2 \times M \text{ matrix})$

Step 6: Calculate the eigenvectors and eigenvalues of the covariance matrix
The covariance matrix C in step 5 has a dimensionality of $N^2 \times N^2$, so one would have N^2 eigenface and eigenvalues. For a 256 × 256 image that means that on 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. In general, PCA is used to describe a large dimensional space with a relative small set of vectors [3].
Compute the eigenvectors u_i of AA^T
The matrix AA^T is very large - - ► not practical!!!

Step 6.1: consider the matrix

$$L = A^T A \quad (\text{M} \times \text{M matrix})$$

Step 6.2: compute eigenvectors v_i of L= $A^T A$
$A^T A\, v_i = \mu_i v_i$
What is the relationship between u_i and v_i ?

$$A^T A\, v_i = \mu_i v_i$$
$$A\, A^T A\, v_i = \mu_i A v_i$$
$$CA\, v_i = \mu_i A v_i \quad [\text{ since } C = A\, A^T\ ]$$
$$C\, u_i = \mu_i A v_i$$

where, $u_i = A\, v_i$ Thus, $C = A\, A^T$ and $L = A^T A$ have the same eigenvalues and their eigenvectors are related as follows: $u_i = A\, v_i$
Note 1: $C = A\, A^T$ can have upto N^2 eigenvalues and eigenvectors.
Note 2: $L = A^T A$ can have upto M eigenvalues and eigenvectors.
Note 3: The M eigenvalues of $C = A\, A^T$ (along with their corresponding eigenvectors) correspond to the M largest eigenvalues of $L = A^T A$ (along with their corresponding eigenvectors).
Where v_i is an eigenvector of L $= A^T A$. From this simple proof we can see that $A\, v_i$ is an eigenvector of $C = A\, A^T$. The M eigenvectors of L $= A^T A$ are used to find the M eigenvectors u_i of C that form our eigenface basis:

$$u_i = \sum_{i=1}^{M} v_i\, \Phi_i$$

Where, u_i are the Eigenvectors i.e. Eigenfaces.

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues)
Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of Characteristic features of the faces.

IV. PROJECTION OF TRAINING SAMPLES INTO THE EIGENFACE SPACE

Next we have to project the training sample into the Eigenface space. The feature weight for the training images can be calculated by the following formula:

$$\omega_i = u_i^T(\Gamma_i - \Psi)$$

Where, u_i is the ith Eigenfaces and i=1, 2, 3 . . . . . .K. The weight is obtained as above form a vector as follows

$$\Omega^T{}_i = [\omega_1, \omega_2, \omega_3 \dots\dots\dots \omega_K]$$

V. TESTING SAMPLE CLASSIFICATIONS

a) Read the test image and separate face from it.
b) Calculate the feature vector of the test face.
The test image is transformed into its eigenface components. First we compare line of our input image with our mean image and multiply their difference with each eigenvectors [2]. Each value would represent a weight and would be saved on a vector Ω^T

$$\omega_{test} = u_i^T(\Gamma_{test} - \Psi)$$

Where, u_i is the ith Eigenfaces and i=1, 2, 3 . . . . . .K.

$$\Omega^T_{test} = [\omega_1, \omega_2, \omega_3 \dots\dots\dots \omega_K]$$

c) Compute the average distance (Euclidean distance) between test feature vector and all the training feature vectors.
Mathematically, recognition is finding the minimum Euclidean distance ε_k , between a testing point and a training point given in the following equation

$$\varepsilon_k = \sqrt{\|\Omega_{test} - \Omega_i\|^2}$$

Where, i = 1, 2, 3. . . . . . K. The Euclidean distance between two weight vectors thus provides a measurement of similarity between the corresponding images.
d) The face class with minimum Euclidian distance shows similarity to test image [5].

VI. SCHEMATIC DIAGRAM & FLOWCHART

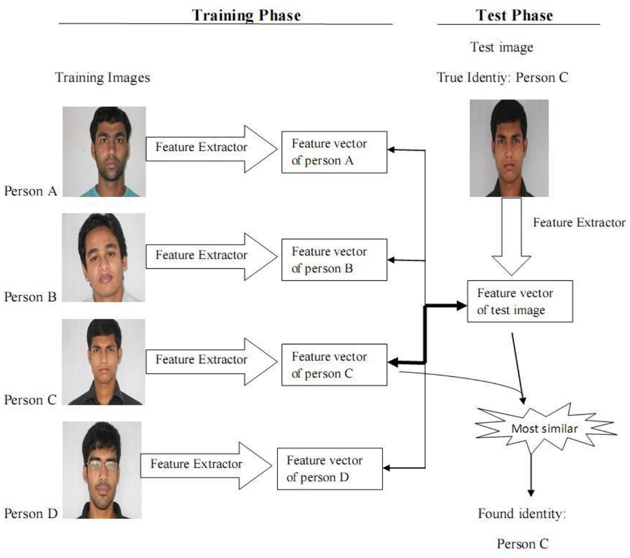
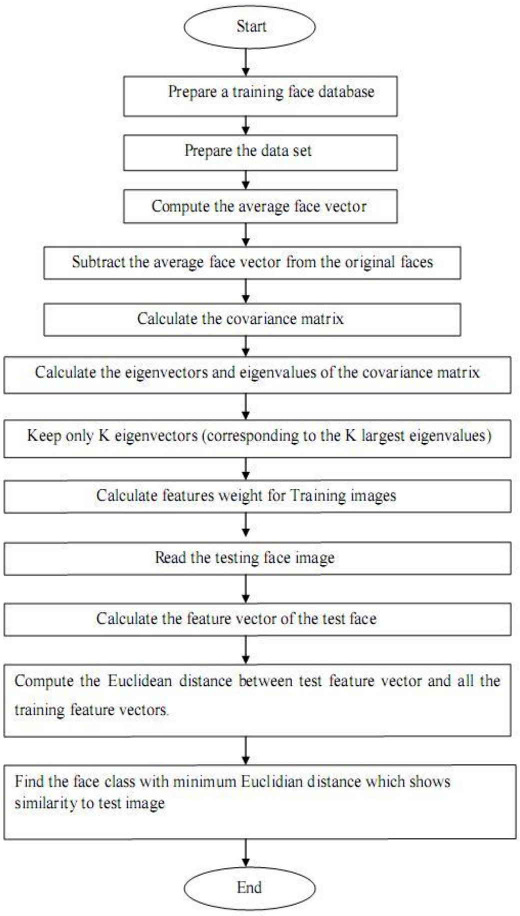


Fig-2: Schematic diagram of a face recognizer


Start
Prepare a training face database
Prepare the data set
Compute the average face vector
Subtract the average face vector from the original faces
Calculate the covariance matrix
Calculate the eigenvectors and eigenvalues of the covariance matrix
Keep only K eigenvectors (corresponding to the K largest eigenvalues)
Calculate features weight for Training images
Read the testing face image
Calculate the feature vector of the test face
Compute the Euclidean distance between test feature vector and all the training feature vectors.
Find the face class with minimum Euclidian distance which shows similarity to test image
End

VII. EXPERIMENTAL RESULT

This article represents some computational results of our program. In Experimental result-1 and Experimental result-2, both test image and equivalent image which is stored in database have same pose. But test image and equivalent image have different in pose which are shown in Experimental result-3 and Experimental result-4.

Experimental Result-1(having same pose)

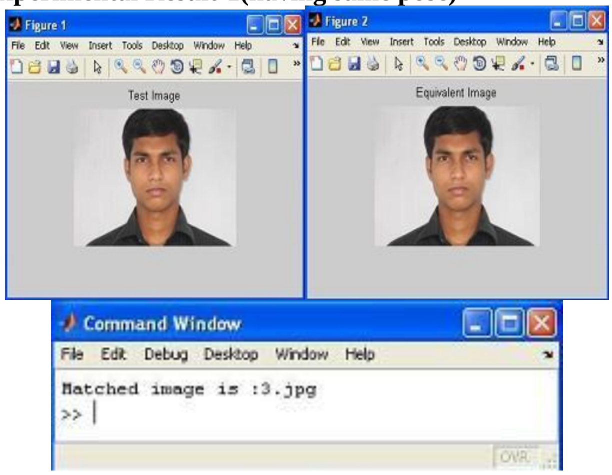
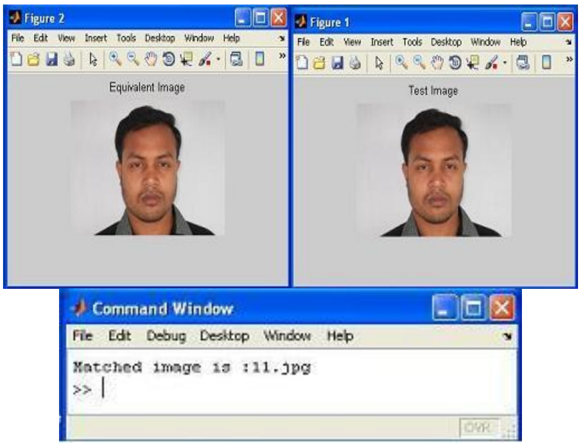


Figure 1
File Edit View Insert Tools Desktop Window Help
Test Image
Figure 2
File Edit View Insert Tools Desktop Window Help
Equivalent Image
Command Window
File Edit Debug Desktop Window Help
OVR

```
Matched image is :3.jpg
>> |
```

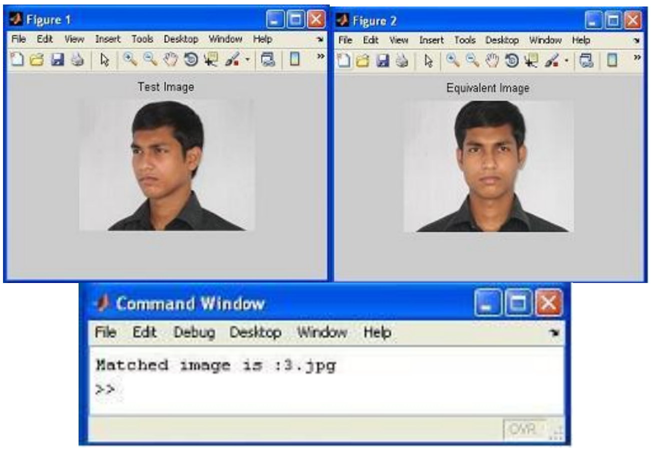
Experimental Result-2(having same pose)



```
Matched image is :11.jpg
>> |
```

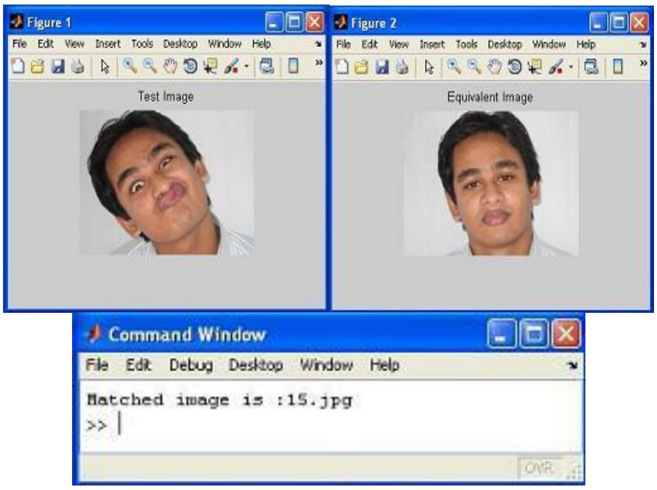
Fig-3: The Recognition of Individual Images having Same Pose

Experimental Result-3 (having pose variation)



```
Matched image is :3.jpg
>>
```

Experimental Result-4 (having pose variation)



```
Matched image is :15.jpg
>> |
```

Figure-4: The Recognition of Individual Images having Pose variation

VIII. CONCLUSION

In this thesis we implemented the face recognition system using Principal Component Analysis and Eigenface approach. The system successfully recognized the human faces and worked better in different conditions of face orientation. In this research, Principal component analysis approach to the face recognition problem was studied and a face recognition system based on the eigenfaces approach was proposed. The algorithm has been tested for the image database ETE-07 series, RUET and implemented using MATLAB. The algorithm developed in a generalized one which works well with any type of images. The tests conducted on Bitmap images, PNG images and JPEG images of various subjects in different poses showed that this method gave very good classification of faces though it has limitations over the variations in size of image. The eigenface approach thus provides a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple and has been shown to work well in constrained environment.

IX. FUTURE PLAN

In this thesis paper, we worked with some still pictures but we will try to develop a system using video camera that will work with real time face recognition. Here we used 36 face images of 18 persons of ETE-07 series, RUET but in future we would like to work with huge database. We want to overcome the problem of different size face image recognition. We will compare the performance analysis of PCA based method with all others existing face recognition methods.

REFERENCES

[1]. http://www.face-rec.org

[2]. Shemi P M, Ali M A, A Principal Component Analysis Method for Recognition of Human Faces: Eigenfaces Approach, International Journal of Electronics Communication and Computer Technology (IJECCT),Volume 2 Issue 3 (May 2012).

[3]. M. Turk, A. Pentland: Face Recognition using Eigenfaces, Conference on Computer Vision and Pattern Recognition, 3 – 6 June 1991, Maui, HI , USA, pp. 586 – 591.

[4]. Prof. Y. Vijaya Lata , Chandra Kiran Bharadwaj Tungathurthi , H. Ram Mohan Rao , Dr. A. Govardhan , Dr. L. P. Reddy,
Facial Recognition using Eigenfaces by PCA, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engg&Tech, Jawaharlal Nehru Tech. University.

[5]. Parvinder S. Sandhu, Iqbaldeep Kaur, Amit Verma, Samriti Jindal, Inderpreet Kaur, Shilpi Kumari, Face Recognition Using Eigen face Coefficients and Principal Component Analysis, International Journal on Electrical and Electronics Engineering 3:8 2009.

AUTHORS PROFILE

First Author: Liton Chandra Paul completed his B.Sc. in Electronics & Telecommunication Engineering (ETE) from Rajshahi University of Engineering & Technology (RUET), Bangladesh, in 2012. Currently, he is a lecturer of Electronics and Communication Engineering department at the University of Information Technology & Sciences (UITS), Bangladesh. He has two international conference paper and one national conference paper. His research interests include Mobile communication, Wireless & Satellite Communication, Image Processing, Digital Signal Processing, Digital Communication, Data Communication.

Second Author: **Abdulla Al Suman** received his B.Sc. in Electronics & Telecommunication Engineering (ETE) from Rajshahi University of Engineering & Technology (RUET), Bangladesh, in 2010. He is pursuing his M.Sc. in EEE from RUET. Currently, he is a lecturer of ETE at the RUET. He has 2 int'l journal and 2 int'l conference paper. His research interests include Wireless Communication, Image Processing, Digital Signal Processing, Information Theory, and Digital Communication.