ISSN22773061

Face recognition using Eigen face based technique utilizing the concept of principal component analysis

Mrs. Sunita Roy[1] and Prof. Samir K. Bandyopadhyay[2]

[1] Ph.D. scholar in the Dept. of Computer Science & Engineering,
University of Calcutta, Kolkata, India,
Email: sunitaroy07@gmail.com,

[2] Professor of the Dept. of Computer Science Engineering,
University of Calcutta, Kolkata, India,
Email: skb1@vsnl.com.

ABSTRACT

Face recognition has been an active research area since late 1980s [1]. It has numerous applications like security-based system, forensic identification, facial expression detection, gender classification etc. It has limited scope in biometric application but they are more useful for surveillance applications like activity tracking and recognition, abnormality detection. Face recognition system can be implemented using various types of methods but in this paper we use appearance-based approach. Eigenface approach is one of the earliest appearance-based face recognition methods, which was developed by M. Turk and A. Pentland [1] in 1991. In this approach we have to perform a lots of computations, which are not feasible with respect to time in many real time system. The concept of principal component analysis (PCA) is used in this approach to reduce the dimension and hence reducing the computation time. Principal component analysis [4] decomposes face images into a small set of characteristic feature images called eigenfaces. Recognition of face image can be done by projection the test image onto a low dimensional linear face space or eigenfaces. After that we compute the distance between the resultant position of the test face image in the face space and those of known face classes. Now we compare the distance against a threshold value, and recognize the face as a known or unknown face.

Keywords: Face recognition, Eigenface, principal component analysis (PCA), face space, appearance-based.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTER AND TECHNOLOGY

Vol 10, No. 8

editor@cirworld.com

www.cirworld.com, member.cirworld.com

1. INTRODUCTION

Among various biological features, face plays an important role to uniquely identify a person. Furthermore, not only the identity, human face can also be used for expression detection, age detection and many more applications. For a human being it is easy to recognize a face at a glance even after years though there are large changes in the visual appearance, due to expression, aging or may be the presence of some objects like glasses, beard etc. Hence developing an automated face recognition system is not an easy task. Now before going into the implementation details, we would like to discuss some important concept other than face recognition that may be considered as preprocessing tasks and without them the process of face recognition is impossible to implement. Whenever we consider an image, it may contain some noise, which has to be removed before going into the next phase. Also an image can be generated by various input sources, hence we need to resize the image into standard size so that the required computational time can be optimized. After that we have to apply the face detection process [9], which will give us a face region. Then we need to normalize the face image so that the head position (level of head) of the face is same as of the face images stored in the database. Therefore the success of all these processes will enhance the performance of the face recognition process [8].

Now come to the concept of face recognition process, which can be either an entry-level based recognition in which, we determine whether a given test face is present in our face database or not. Other type of recognition is classification-based recognition, which take an input face image and categorize according to the requirements. In our paper, we will discuss about the implementation details of the entry-level based recognition system.

Face recognition approaches can be classified as three categories: template matching-based, feature-based, and appearance-based [7].

In template-matching based technique, we represent the face as a template considering the relative distance between various facial features like distance between two eyes, distance between eye and nose, distance between nose and mouth. Instead of the whole face we can think each individual facial feature as a template like eye template, nose template and mouth template. Then we compare their property with the database image. Though these types of recognition methods are easy to implement but the memory requirement is very high. Feature-based approaches have smaller memory requirement and a higher recognition speed than template-based method. However, perfect extraction of various facial features is very difficult to implement. In the last approach, that is appearance-based approach, we project the face image onto a lower dimensional linear subspace. This subspace is constructed by principal component analysis on a set of training images, with eigenfaces as its eigenvectors. In this paper we will focus only on the appearance-based approach.

2. FACE RECOGNITION

One of the simplest and most effective ways to recognize a face is eigenface approach [1]-[3]. In this approach we use the concept of principal component analysis (PCA) to transform a face image onto a lower dimensional subspace that consists eigenfaces. PCA [5] [6] technique always finds out a set of projection vector such that the projected data retains the most important information about the original data. One common way to do this is to consider those eigenvectors corresponding to the highest eigenvalues of the covariance matrix. This method reduces the dimensionality of data space by projecting data from M-dimensional space to P-dimensional space, where $P << M$.

Now recognition can be done by projecting a new face onto the lower dimensional subspace of eigenfaces [10]. After that we compare its position in the subspace with the position of known individuals. Then we measure their distance, which can be Mahalanobis distance or Euclidean distance. In our approach we use Euclidean distance. If the Euclidean distance is greater than a threshold value, the person is unknown otherwise; it is a known person [1].

2.1 Eigenfaces for Face Recognition

In early 1990s, M. Turk and A. Pentland have realized that using some encoding and decoding techniques with the help of the significant local and global “features” can represent information content of face images. These features may or may not be related to our intuitive notion of face features such as the eyes, nose, lips, and hair. Hence in the face recognition process we first do some encoding on the test face and then compare the result with the other encoded information produced by the database images using the same encoding technique. A simple way to represent the face information is to use the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.

Principal component analysis finds the principal components of the distribution of faces in terms of eigenvectors of the covariance matrix of the set of face images. These eigenvectors can be thought of as a set of features, which together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face called an eigenface. Some of these faces are shown in Figure 4. Each face image in the training set can be represented exactly in terms of a linear combination of the eigenfaces. The number of possible eigenfaces is equal to the number of face images in the training set. However, we can consider less number of eigenfaces, which have the largest eigenvalues and labeled them as *‘best’* eigenfaces. Thus, we can reduce the computation time.

2.1.1 The eigenface approach for face recognition involves the following steps:

Step 1: Obtain face images $I_1, I_2, ..., I_M$ that form the training set.

Figure 1: Face images from training set.

MATLAB command: imread (' Filename . extension');

Step 2: Represent every image I_i as a vector Γ, where Γ is an N^2 x 1 vector, corresponding to an N x N face image I.

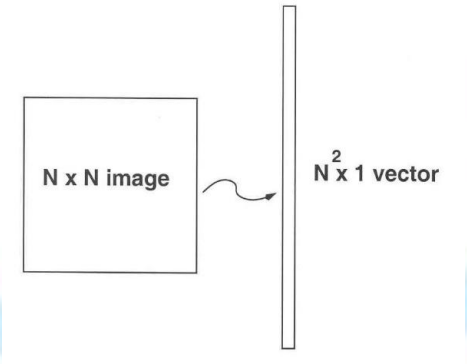

Figure 2: Representation of an N x N face image to an N^2 x 1 vector.

MATLAB command:m=1;

```
for j=1 : col
    for i=1 : row
        A(m,1) = I(i, j);
    end
end
```

Step 3: Compute the average face vector Ψ:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

MATLAB command: Let images are stored in A_1, A_2,…,A_n vectors, which are the columns of matrix 'Img' with dimension row*col by n and 'avgimg' is the average image.

```
sum=0;
for i=1 : n
      Img(:,i)=double(A(:,i));
      sum=sum + A;
end
      avgimg=sum/n;
      imshow(uint8(avgimg));
```

Figure 3: Average face image.

Step 4: Subtract the mean face from each face images in the training set:

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

Let 'O' is the matrix of dimension row*col by n where we store the result after the subtraction operation.

MATLAB command: for i=1 : n

```
            O(:,i)=Img(:,i) – avgimg;
      end
```

Step 5: Compute the covariance matrix C:

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

where the $A = [\Phi_1\Phi_2...\Phi_M]$ is N^2 by M matrix. Hence after the above operation the dimension of matrix C will be N^2 by N^2. But for a large value of N it is almost infeasible to calculate such amount of calculations. Hence we need a computationally feasible method to find these eigenvectors. To compute the eigenvectors u_i of AA^T we do the following steps:

Step 5.1: Compute the eigenvectors v_i of A^TA.

$A^TA\ v_i = \mu_i\ v_i$

Multiplying both sides by A we get

$AA^TA\ v_i = \mu_i A\ v_i \Rightarrow CA\ v_i = \mu_i A\ v_i \Rightarrow Cu_i = \mu_i u_i$ where $u_i = A\ v_i$

Thus, AA^T and A^TA have the same eigenvalues and their eigenvectors are related as follows: $u_i = A\ v_i$.

Step5.2: Compute *M* eigenvalues of AA^T.

However AA^T have N^2 eigenvalues and their eigenvectors, A^TA have *M* eigenvalues and their eigenvectors, where $M << N^2$. These *M* eigenvalues of A^TA corresponds to the *M best* eigenvalues of the AA^T. This is the main idea behind principal components analysis, where we consider only the best components from a domain of objects to optimize the computation time. In the above approach we have consider only *M largest* eigenvalues among N^2 eigenvalues.

Step 5.3: Compute *M* eigenvectors of A^TA.

MATLAB command: C=O′ * O

[V, D]=eig(C);

where V stores all the eigenvectors and D stores all the eigenvalues.

Figure 4: Eigenfaces of each image in the training set.

2.1.2 Representation of face images in the training set

After generating the eigenvectors we can represent each face image $\mathbf{\Phi_i}$ in the training set as a weighted sum of *K* eigenvectors among all these *M* eigenvectors.

Figure 5: Representation of each face image as a linear combination of eigenvectors.

$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

Hence each normalized training face $\mathbf{\Phi_i}$ is represented by a linear combination of weight vector and eigenvector, where the weight vector $\mathbf{\Omega_i}$ is as follows:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \dots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M$$

Let 'eigface' is a matrix of dimension row*col by n where each column represents an eigenface. Matrix 'w' contains the weights that result from the projection onto the facespace.

MATLAB command: for i=1 : n

```
        eigface(:,i)=O * V(:,i);
    end

    for j=1 : n
        for i=1 : n
            w(i,j)=(u(:,i))'*(Img(:,j)-avgimg);
        end
    end
```

2.1.3 Face recognition using Eigenfaces

In this stage a given face image Γ will be verified with the help of a training set. If the person is an authorized person, the module will show the recognized face and for an unauthorized person it will show an error message. The test face image should be in a normalized form (centered and of the same size like the training faces).

Step 1: Read the test face image and convert it into a vector.

Let test image is stored in the matrix 'I' and then we convert it into a vector 'testimage'.

MATLAB command: I=imread('filename.ext');

```
    I=rgb2gray(I);
    m=1;
    for i=1 : col
        for j=1 : row
            testimage(m,1)=I(j,i);
            m=m+1;
        end
    end
    testimage=double(testimage);
```

Step 2: The test face image Γ is transformed into its eigenface components (projected onto "face space").

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$$

Matrix 'wtest' contains the weights that result from the projection onto the facespace.

MATLAB command: for i = 1 : n

```
        wtest(i,:)= (u(:,i))' * (testimage(:,1)- avgimg);
    end
```

Step 3: Weight vector should be represented in the following form. Each weight describes the amount of contribution by eigenface in the test face image. Now we generate the image with the help of the weight vector.

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_K \end{bmatrix}$$

Let 'sum' and 'final' is a row*col by 1 vector. In vector 'final' we store the final image.

MATLAB command: for i=1 : n

```
sum(:,1) = sum(:,1)+u(:,i) * wtest(i);
end
final(:,1) = sum + avgimg;
```

'final_image' is a matrix of dimension row by col which store the image in matrix form.

```
a=1;b=1;
for i=1 : row*col
    final_image(a,b)=final(i,1);

    if (rem(i,row)==0)
        b=b+1;
        a=1;
    else
        a=a+1;
    end
end
```

Step 4: Now we find which face image in the training set best matches the test face image. We use euclidean distance to measure the distance between two images. If the distance (miminum euclidean distance) is greater than a theshold value, the images are of two different persons and if the distance is less than or equal to the threshold value, the images are of the same person. The euclidean distance between two images is determined by the following equation:

$$\varepsilon_k^2 = \left\| (\Omega - \Omega_k)^2 \right\|$$

where Ω_k is a vector describing the *k*th face class and Ω is a vector describing the test face class.

MATLAB command: for i=1 : n

```
a(:,1)=w(:,i);
b(:,1)=wtest(:,1);
sum=0;
aa=a(:,1);
bb=b(:,1);
for j=1:n
    term=aa(j,1)-bb(j,1);
    term1=term*term;
    sum=sum+term1;
end
eucl_dist=sqrt(sum)
eucl_dist_list(i)=eucl_dist;
end
```

Step 5: Now find the minimum and maximum euclidean distance to classify the test face as known or unknown face.

$$\varepsilon = \min \| \varepsilon_k \|$$

Also consider a threshold value θ.

$$\text{if } \varepsilon <= \theta$$

The person is known and return the k^{th} face.

else

print 'person is unknown'..

We use maximum Euclidean distance for our threshold value and minimum Euclidean distance as our decision parameter.

MATLAB command: a=max(eucl_dist_list) * 0.23;

```
b=min(eucl_dist_list);
if(b>a)
    msgbox('UNKNOWN IMAGE','error','error');
else
    imshow(gray2rgb(uint8(finalimg)));
end
```

3. EXPERIMENTAL RESULTS

We use MATLAB GUI based interface to design our face recognition system. Here we use two interfaces. The first one is dynamic in nature and will show the test face and the recognized face. The second one will show the information, which are static in nature like the database images, the eigenfaces of the database images, the average image. In our system we use three types of images.

Case 1: If the person is known and the test face is exactly looks same as stored in the database.

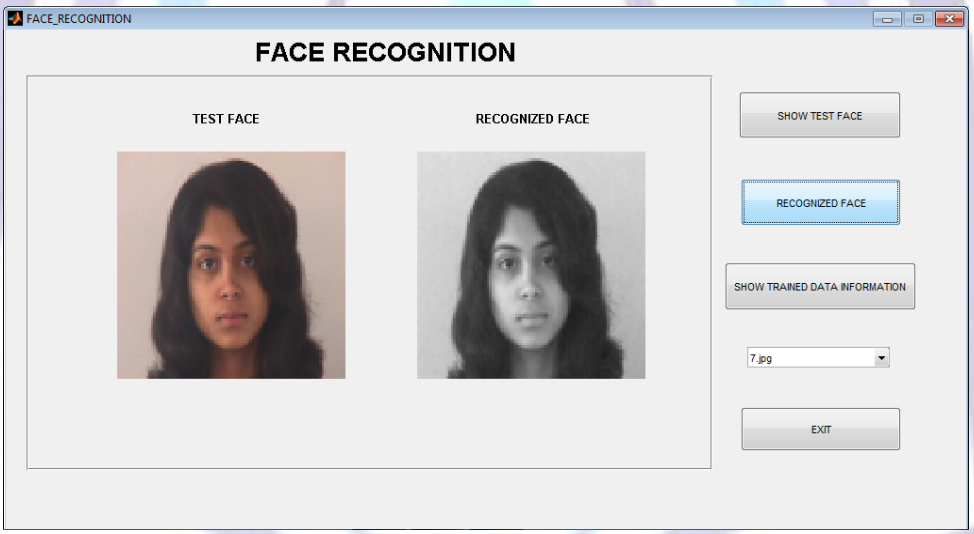

Figure 6: Recognized face for a known test face.

Case 2: If the person is unknown.

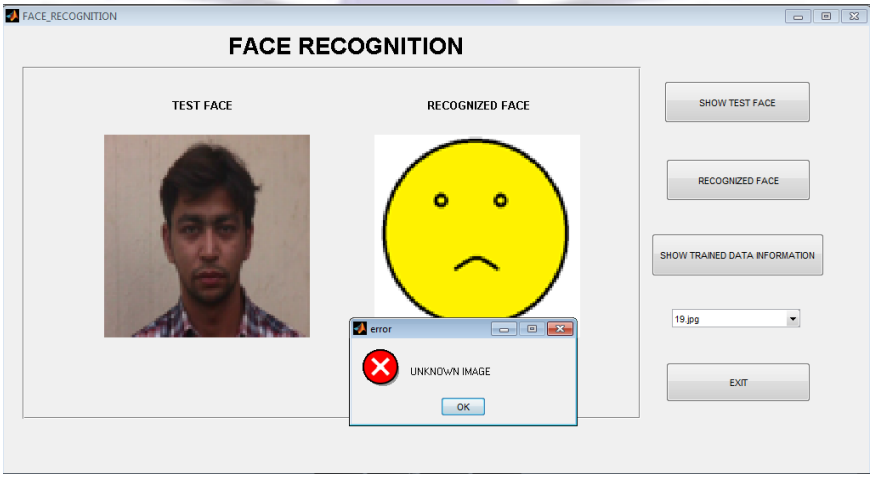

Figure 7: Recognized face with an error message for an unknown test face.

Case 3: If the person is known but the image stored in the database is different from the test face image.

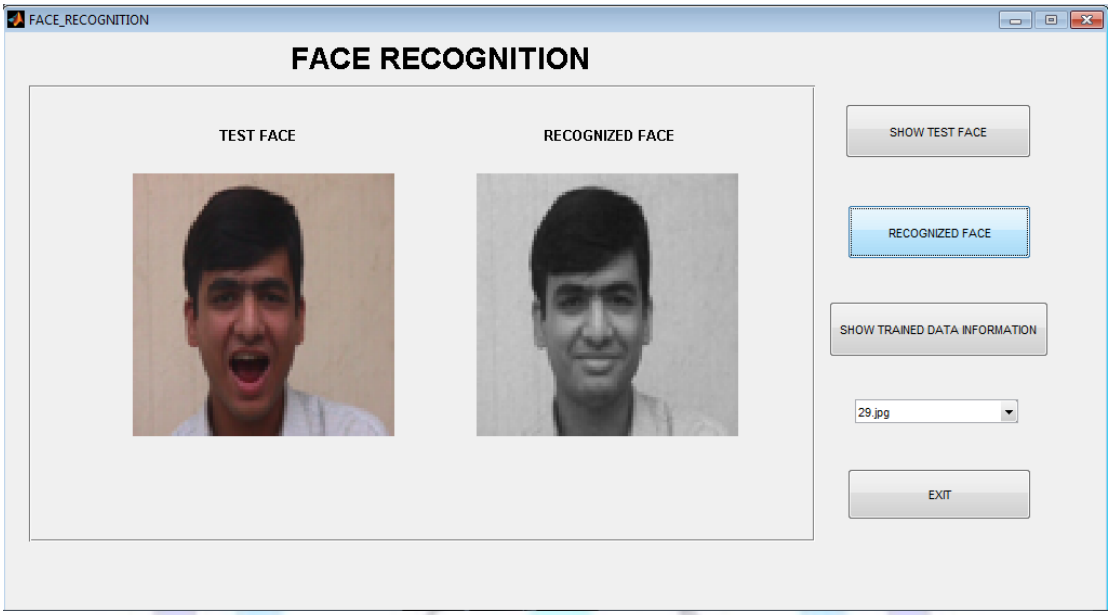

Figure 8: Recognized face for a morphed known test face.

In the second interface we show some static information but the output will vary only after image addition and deletion operation in the database.

Information 1: Database images

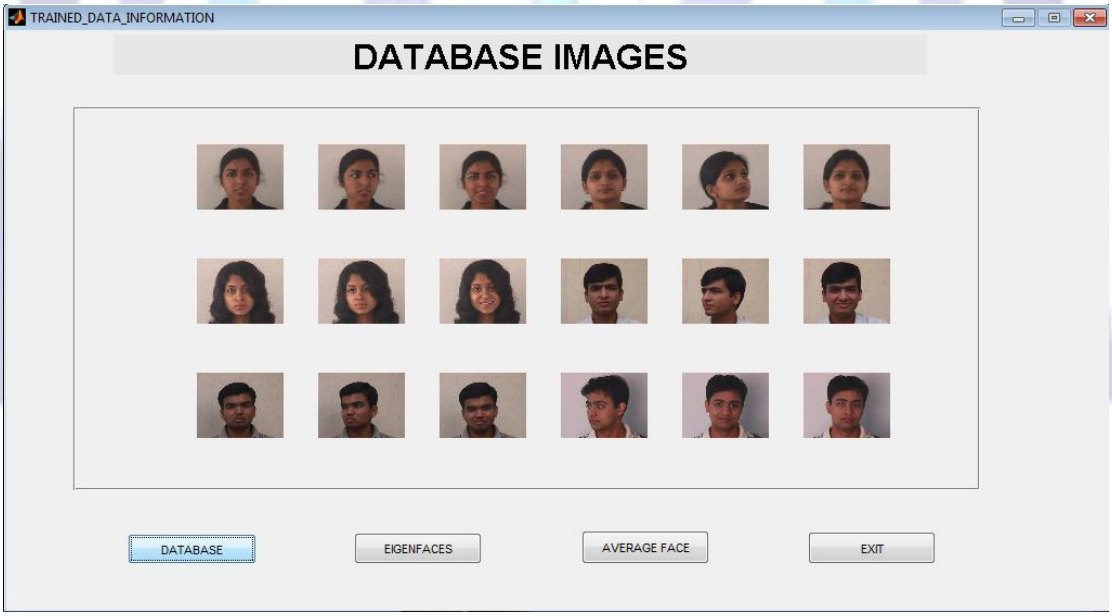

Figure 9: Database of training face images.

Information 2: Eigenfaces of the database images.

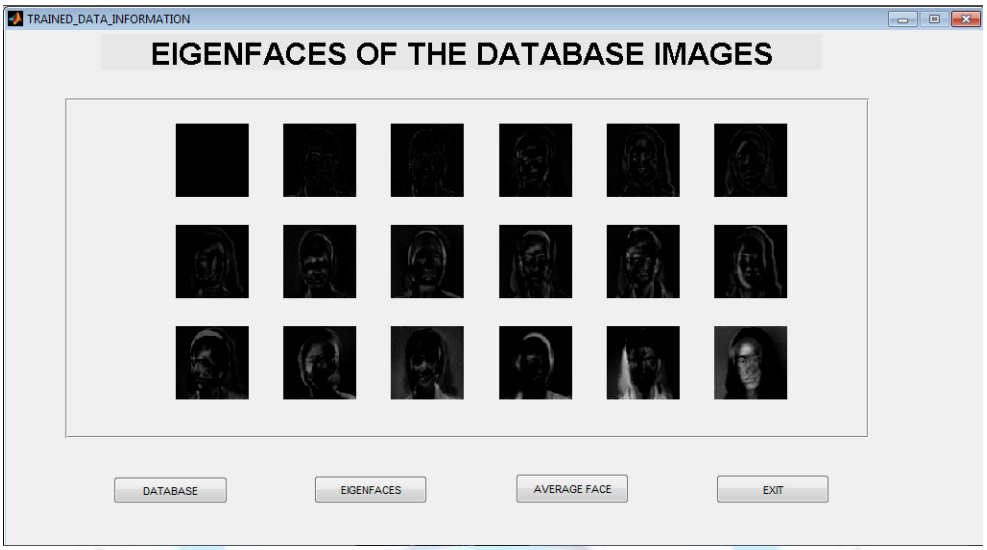

Figure 10: Eigenfaces.

Information 3: Average image.

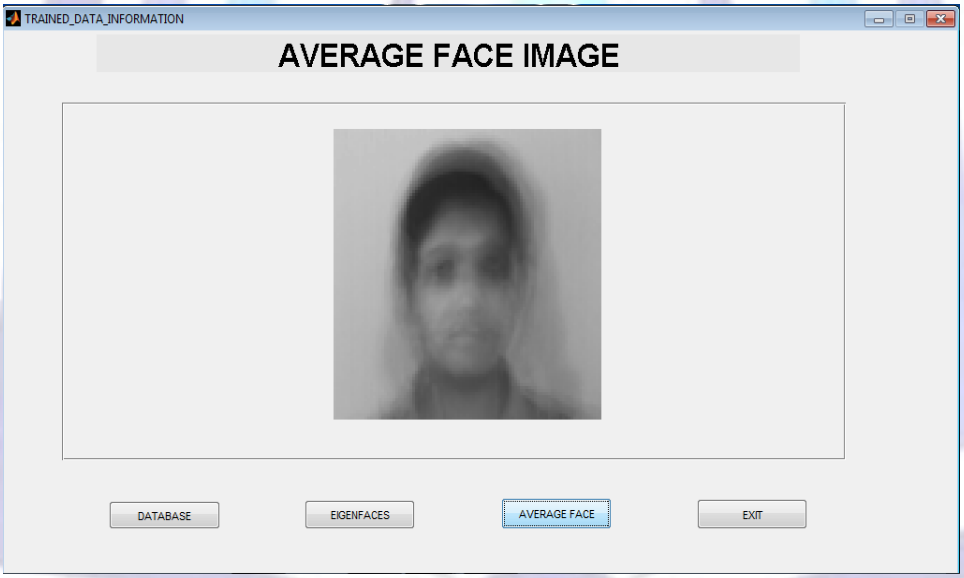

Figure 11: Average face image.

Number of test images	30
Number of correctly recognized images	29
Number of incorrectly recognized images	1
Success rate	96%

Table 1: Statistical record for our face recognition system.

4. CONCLUSION

In this paper we have introduced an eigenface-based face recognition approach. To recognize a face, we first transform all the database images onto lower-dimensional linear subspace called 'face space' or eigenfaces. After that we need to transform the test face onto the same 'face space'. Now we compute the distance between the relative position of these two images onto the space. The distance can be measured using different techniques, but here we use euclidean distance. The above approach is simple and ease to implement because we do not need any knowledge of geometry or

facial feature information. Furthermore we need to perform some preprocessing on the face images to transform it onto the 'face space'.

There are some limitations in our face recognition approach. First, the algorithm is sensitive to head scale. Second, it is applicable only to frontal view face. Third, this approach some time may fails in case of different background or natural scene. Forth, we consider all the eigenvectors instead of using only *K* best eigenvectors and this causes greater computation time.

5. REFERENCES

[1] M. Turk and A. Pentland ,"Eigenfaces for recognition", Journal of Cognitive Neuroscience, vol.3, No.1, 1991.

[2] M. Turk and A. Pentland, "Face recognition using eigenfaces", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages 586-591, 1991.

[3] A. Pentland and T. Choudhury, "Face recognition for smart environments", Computer, Vol.33 Iss.2, Feb. 2000.

[4] V. Perlibakas, "Face Recognition using Principal Component Analysis and Wavelet Packet Decomposition", Informatica, Vol. 15, No. 2, pp. 243 – 250, 2004.

[5] K. Kim, "Face Recognition using Principal Component Analysis", National Institute of Technology, Rourkela, India, 2008.

[6] V. Perlibakas, "Distance Measures for PCA-based Face Recognition", Pattern Recognition Letters, Vol. 25, No. 6, pp. 711 – 724, April 2004.

[7] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, pp. 1042-1052. 1993.

[8] R. J. Baron,"Mechanisms of human facial recognition.." International Journal of Man Machine Studies, Vol. 15, pp. 137-178. 1981.

[9] L. S. Balasuriya, "Frontal View Human Face Detection and Recognition," B.Sc.(Hons) Thesis, Department of Statistics and Computer Science, University of Colombo. 2000.

[10] A.J. O'Toole, H. Abdi, K.A. Deffenbacher, and D. Valentin, "A low-dimension representation of faces in the higher dimensions of the space", Journal of the Optical Society of America A, Vol. 10, pp. 405-411, 1993.

6. AUTHORS' BIOGRAPHY

Mrs. Sunita Roy

She received her B.Sc. degree in Computer Science from Barrackpore Rastraguru Surendranath College, 2006 from University of Calcutta. She completed her M.Sc. in Computer and Information Science from University College of Science and Technology, University of Calcutta in 2008. She did her M.Tech. in Computer Science and Engineering from University of Calcutta in 2010. She is the author of five International Journals. Her fields of Specialization are Face detection and recognition, Image Processing, Data Structure, Programming Language etc.

Prof. Samir Kumar Bandyopadhyay

He is the Professor of the department of Computer Science & Engineering, University of Calcutta, India. He received his PhD award in Computer Science & Engineering, 1989 from University of Calcutta, His thesis title is "Diagnosis of Cardiac Diseases using Syntactic Pattern Recognition Approach". He completed his M.Tech in Radio-Physics & Electronics, 1979 from University of Calcutta, BE in Electronics TeleCommunication,1975 from BE College, Shibpur, University of Calcutta. Chairman, Science & Engineering Research Support Society (SERSC, Indian Part), Fellow of Computer Society of India, Sectional President of ICT of Indian Science Congress Association, 2008-2009, Senior Member of IEEE, Member of ACM, Fellow of Institution of Engineers (India), Fellow of Institution of Electronics & Tele-Communication Engineering, India, Reviewer of International Journals IEEE Trans on Neural Networks, ACM, Springer Publications. Published Books like Data structure Using C, Addison Wesley, 2003, C Language, Pear-son Publication, 2010. He is the author of more than hundred publications in National and International Journal and Conference.