# Fast Face Recognition Using Eigen Faces

Arun Vyas
M. Tech scholar
Marudhar Engineering College
Bikaner
*picker.arun@gmail.com*

Rajbala Tokas
Assistant Professor
Marudhar Engineering College
Bikaner
*rajtokas26@gmail.com*

***Abstract***— Face is a typical multidimensional structure and needs good computational analysis for recognition. Our approach signifies face recognition as a two-dimensional problem. In this approach, face recognization is done by Principal Component Analysis (PCA). Face images are faced onto a space that encodes best difference among known face images. The face space is created by eigenface methods which are eigenvectors of the set of faces, which may not link to general facial features such as eyes, nose, and lips. The eigenface method uses the PCA for recognition of the images. The system performs by facing pre-extracted face image onto a set of face space that shows significant difference among known face images. Face will be categorized as known or unknown face after imitating it with the present database.

***Keywords***— *Fast, face, recognition, eigenfaces, PCA, differential weights*

*****

## I. Introduction

The human face is an extremely complex and dynamic structure with characteristics that can quickly and significantly change with time. It is the primary focus of attention in social relationships and plays a major role in the transmission of identity and emotions. Therefore, face recognition is applied in many important areas such as security systems, identification of criminals, and verification of credit cards and so on. Unfortunately, many face features make development of facial recognition systems difficult.

This problem is solved by the method called Principal Component Analysis. PCA is a projection technique that finds a set of projection vectors designed such that the projected data retains the most information about the original data. The most representative vectors are eigenvectors corresponding to 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 [1].

## II. Face Recognition

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 [2]. 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 [3]. Namely, the images must be vertical frontal views of human faces.

## III. PCA Approach to Face Recognition

Principal component analysis transforms a set of data obtained from possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of components can be less than or equal to the number of original variables. The first principal component has the highest possible variance, and each of the succeeding components have the highest possible variance under the restriction that it has to be orthogonal to the previous component. We want to find the principal components, in this case eigenvectors of the covariance matrix of facial images. The first thing we need to do is to form a training data set. 2D image *Ii* can be represented as a 1D vector by concatenating rows [3]. Image is transformed into a vector of length *N* = mxn.

$$\boldsymbol{I} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{mxn} \xrightarrow{\text{CONCATENATION}} \begin{bmatrix} x_{11} \\ \vdots \\ x_{1n} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{mn} \end{bmatrix}_{1xN} = \boldsymbol{x}\,.$$

Let *M* such vectors ***xi*** (*i* = 1, 2... *M*) of length *N* form a matrix of learning images, ***X***. To ensure that the first principal component describes the direction of maximum variance, it is necessary to centre the matrix. First we determine the vector of mean values **Ψ**, and then subtract that vector from each image vector.

$$\boldsymbol{\Psi} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{x}_i \tag{1}$$

$$\phi_i = \boldsymbol{x}_i - \boldsymbol{\Psi} \tag{2}$$

Averaged vectors are arranged to form a new training matrix (size NxM)

$$A = [\Phi_1, \Phi_2, \Phi_3, \Phi_4...];$$

The next step is to calculate the covariance matrix ***C***, and find its eigenvectors ***ei*** and eigenvalues ***λi***:
Where $C = AA^T$

$$C^* e_i = \lambda_i e_i \quad (3)$$

Covariance matrix ***C*** has dimensions *N*x*N*. From that we get *N* eigen values and eigenvectors. For an image size of 128x128, we would have to calculate the matrix of dimensions 16.384x16.384 and find 16.384 eigenvectors. It is not very effective since we do not need most of these vectors. Rank of covariance matrix is limited by the number of images in learning set — if we have *M* images, we will have *M*–1 eigenvectors corresponding to non-zero eigenvalues [4].
One of the theorems in linear algebra states that the eigenvectors $e_i$ and eigenvalues $\lambda_i$ can be obtained by finding eigenvectors and eigenvalues of matrix $C=A^TA$ (dimensions *M*x*M*) [5]. If $v_i$ and $\mu_i$ are eigenvectors and eigen values of matrix $A^TA$, eigenvector associated with the highest eigenvalue reflects the highest variance, and the one associated with the lowest eigenvalue, the smallest variance. Eigenvalues decrease exponentially so that about 90% of the total variance is contained in the first 5% to 10% eigenvectors [3]. Therefore, the vectors should be sorted by eigenvalues so that the first vector corresponds to the highest eigenvalue. These vectors are then normalized. They form the new matrix ***E*** so that each vector $e_i$ is a column vector. The dimensions of this matrix are *NXD*, where *D* represents the desired number of eigenvectors. It is used for projection of data matrix ***A*** and calculation of $y_i$ vectors of matrix

$$Y = [y_1, y_2, y_3, ..... y_M]$$

The matrix Y is given as $Y = E^TA$

Each original image can be reconstructed by adding mean image **Ψ** to the weighted summation of all vectors $e_i$.
The last step is the recognition of faces. Image of the person we want to find in training set is transformed into a vector ***P***, reduced by the mean value **Ψ** and projected with a matrix of eigenvectors (eigenfaces):

$$\omega = E^T(P - \psi)$$

Classification is done by determining the distance, $\varepsilon_i$, between **ω** and each vector $y_i$ of matrix ***Y***. The most common is the Euclidean distance, but other measures may be used. This paper presents the results for the Euclidean distance.

If ***A*** and ***B*** are two vectors of length ***D***, the Euclidean distance between them is determined as follows:

Euclidean distance:

$$d(A,B) = \sqrt{\sum_{i=1}^{D}(a_i - b_i)^2} = \|A - B\| \quad (4)$$

If the minimum distance between test face and training faces is higher than a threshold θ, the test face is considered to be unknown; otherwise it is known and belongs to the person in the database.

$$S = \text{argmin}_i [\varepsilon_i] \quad (5)$$

The program requires a minimum distance between the test image and images from the training base. Even if the person is not in the database, the face would be recognized. It is therefore necessary to set a threshold that will allow us to determine whether a person is in the database. There is no formula for determining the threshold. The most common way is to first calculate the minimum distance of each image from the training base from the other images and place that distance in a vector *rast*. Threshold is taken as 0.8 times of the maximum value of vector *rast* [5]:

$$\theta = 0.8 * \max(rast) \quad (6) .$$

## IV. Experimental Results

The experiment was conducted using database of 8 faces taken from the webcam of the user. The training database contains 10 images containing (8 known and 2 unknown). All photos are resized to dimensions 50x50 and the subject is photographed in an upright, frontal position. All images are grayscale (intensity levels of gray are taken as image features). Example of images from the training base is given in Fig. 1. Fig. 2 shows the average eigen faces. Each eigenvalue corresponds to a single eigenvector and tells us how much images from training bases vary from the mean image in that direction. It can be seen that about 10% of vectors have significant eigenvalues, while those for the remaining vectors are approximately equal to zero. We do not have to take into account eigenvectors that correspond to small eigenvalues because they do not carry important information about the image.
To recognize the face we calculate the distance of test image from each image from the training base. Minimum distance shows us which image from the database matches the test image best. Fig. 5 shows the Euclidean distance between the test image and all 8 images from database. This means that the test image completely matches the picture 4 from the training base (Fig. 1).

Fig 1. Training Images

ISSN: 2321-8169
3615 – 3618

Fig. 2 Average Face

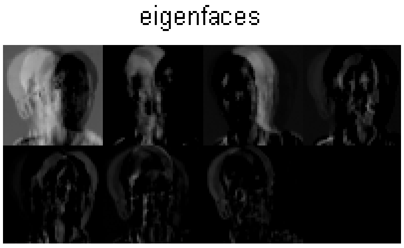


Fig 3 Eigen Faces

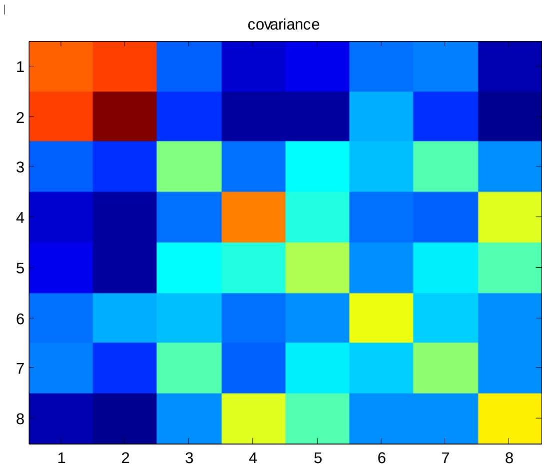


Fig. 4 Covariance Matrix 8x8

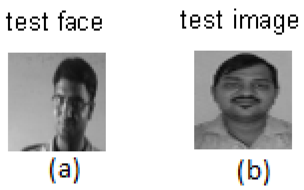


Fig. 5 Test images

Using these test images, the different weights were found to be as follows:

For Test Image 5(a)

1.0e+003 *

1.3399
0.9121
0.7084
1.4467
1.3148
1.0830
2.0306
1.9073

And the Euclidean distance = 4.4855e+003

For test image 5(b)

1.0e+003 *

0.9085
0.3573
0.5632
1.0584
1.0614
0.9596
1.5351
-0.0985

And the Euclidean distance = 4.5972e+003

ans = not in database

So we can see here that for test image 1, the Euclidean distance is lower than that of test image 2 which concludes us that test image 2 is not a part of the database or the face is not recognized.

## V. Conclusion

In this paper, a fast face recognition method using eigenfaces is proposed. We used database of face images which contains 8 images, two images are taken for testing purpose, and one is related to database and other a random image. From the results, it can be concluded that, for recognition, it is sufficient to take about 10% eigenfaces with the highest eigenvalues. It is also clear that the recognition rate increases with the number of training images.

## References

[1] Mamta Dhanda, Face Recognition Using Eigenvectors from Principal Component Analysis, in International Journal of Advanced Engineering Research and Studies, IJAERS/Vol. I/ Issue II/, pages 37-39, January-March, 2012.

[2] Vinay Hiremath, Ashwini Mayakar, Face Recognition Using Eigenface Approach.

[3] Marijeta Slavković1 and Dubravka Jevtić. Serbian Journal Of Electrical Engineering Vol. 9, No. 1, pages 121-130, February 2012

[4] Manal Abdullah, Majda Wazzan and Sahar Bo-Saeed. Optimizing Face Recognition Using PCA, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012

[5] M. A. Turk and A. P. Pentland. Face Recognition using Eigenfaces. in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 91, pages 586 – 591, 1991.

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

ISSN: 2321-8169
3615 – 3618


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

[8] R. Brunelli and T. Poggio, "Face recognition: Features versus templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Pg. 1042-1052, 1993

[9] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey", *Proc. of IEEE*, volume 83, pages 705-740, 1995

[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Pg. 711-720, 1997

[11] A. S. Georghiades, D. J. Kriegman, and P. N. Belhumeur, "Illumination cones for recognition under variable lighting: Faces", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 52-59, 1998.