



Eigenface and PCA Based Face Recognition System

Moumita Pal, Bikash Dey

Department of ECE, JIS College of Engineering, Kalyani,
West Bengal, India

Abstract— In this paper, a new technique of principal component analysis (PCA) is developed for image representation. The basic idea is to first map the input space into a feature space via nonlinear mapping and then compute the principal components in that feature space. This article deals with the feature of PCA as a mechanism for extracting facial features. The principal components are computed within the space spanned by high-order correlations of input pixels making up a facial image, thereby producing a good performance. Then recognizes the person by comparing characteristics of the face to those of known individuals is described. This approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. Face images are projected onto a feature space ('face space') that best encodes the variation among known face images. The face space is defined by the 'eigenfaces', which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner

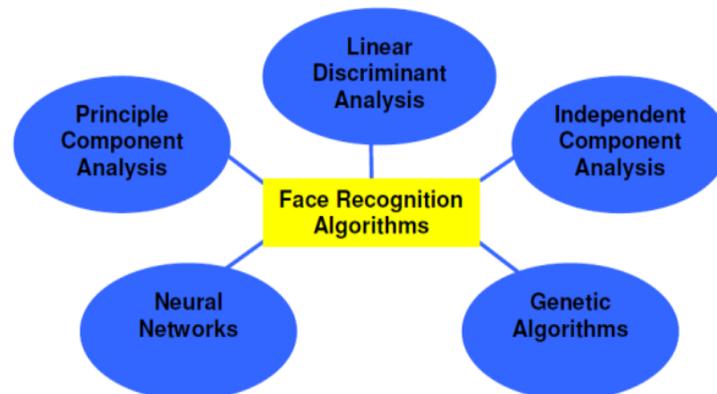
Keywords— Eigenface, face recognition, kernel principal component analysis, machine learning.

I. INTRODUCTION

The face is our primary focus of attention, playing major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable. Recognition of thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expression, aging, and distraction such as glasses, beards or changes in hair style. Face recognition has become an important issue in many applications such as security systems, credit card verification and criminal identification.

Developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional visual stimuli. Therefore, face recognition is a very high-level computer vision task, in which many early vision techniques can be involved. Lately, confront face recognition has pulled in much consideration also its exploration has quickly stretched by not just designers anyway likewise neurosciences, since it has numerous potential applications in machine vision correspondence and programmed access control framework. It has some broad applications such as Commercial and research application.

II. LITERATURE REVIEW & OVERVIEW OF PROPOSED METHODOLOGY



Principle Component Analysis, Linear Discriminate Analysis, Independent Component Analysis, Neural Networks, Genetic Algorithms are the various techniques used for face recognition. But in this paper Principal Component Analysis is considered.

2.1 Principal Component Analysis

Principle Component Analysis (PCA) is one of the most commonly used methodologies [4] in the Statistics and Data-Mining community. The Eigen Object Recognizer class applies PCA on each image, the results of which will be an

array of Eigen values that a Neural Network can be trained to recognize. PCA is a commonly used method of object recognition as its results, when used properly can be fairly accurate and resilient to noise. To perform PCA several steps are undertaken:

STAGE 1: Mean Subtraction

This data is fairly simple and makes the calculation of our covariance[2] matrix a little simpler now this is not the subtraction of the overall mean from each of our values as for covariance we need at least two dimensions of data. It is in fact the subtraction [6]of the mean of each row from each element in that row. (Alternatively the mean of each column from each element in the column however this would adjust the way we calculate the covariance matrix).

STAGE 2: Covariance Matrix

The basic Covariance equation for two dimensional data is:

$$cov(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)}$$

Which is similar to the formula for variance however, the change of x is in respect to the change in y rather than solely the change of x in respect to x. In this equation x represents the pixel value [3]and \bar{x} is the mean of all x values, and n the total number of values. The covariance matrix that is formed of the image data represents how much the dimensions vary from the mean with respect to each other. The definition of a covariance matrix is:

$$C^{n \times n} = (C_{ij}, C_{ij} = cov(Dim_i, Dim_j))$$

Now with larger matrices this can become more complicated and the use of computational algorithms essential.

STAGE 3: Eigenvectors and Eigen values

Eigen values are a product of multiplying matrices how-ever they are as special case. Eigen values are found by multiples of the covariance matrix by a vector in 2 dimensional spaces (i.e. a Eigenvector). This makes the covariance matrix the equivalent of a transformation matrix.

Eigenvectors are usually scaled to have a length of 1:

$$\begin{bmatrix} A \\ B \end{bmatrix} \text{ becomes } \begin{bmatrix} A / (\sqrt{A^2 + B^2}) \\ B / (\sqrt{A^2 + B^2}) \end{bmatrix}$$

The Eigen value is closely related to the Eigenvector used and is the value of which the original vector was scaled

STAGE 4: Feature Vectors

Once Eigenvectors are found from the covariance matrix, the next step is to order them by Eigen value, highest to lowest. This gives you the components in order of significance[9]. Here the data can be compressed and the weaker vectors are removed producing a lossy compression method, the data lost is deemed to be insignificant.

Resultant Eigenvalues= 0.63920.7691

STAGE 5: Transposition

The final stage in PCA is to take the transpose of the feature vector matrix[4] and multiply it on the left of the transposed adjusted data set (the adjusted data set is from Stage 1 where the mean was subtracted from the data). The Eigen Object Recognizer class performs all of this and then feeds the transposed [9]data as a training set into a Neural Network.

2.2 Eigen faces

The eigenface face recognition system can be divided into two main segments: creation of the eigenface basis and recognition, or detection, of a new face. A robust detection system can yield correct matches when the person is feeling happy or sad.

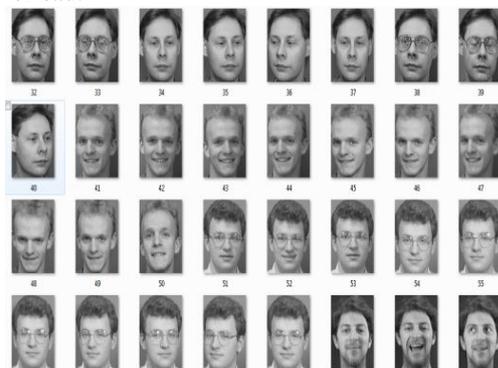


Fig 3.1: Example image from AT&T database.

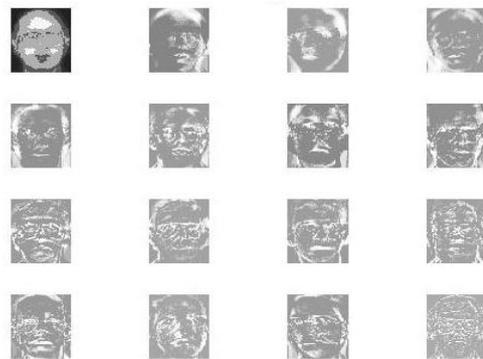


Fig 3.2: Top ten eigenfaces.

Next calculation of the average face in face space is needed. Here M is the number of faces in our set:

$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$, then computation of each face's differs from the average:

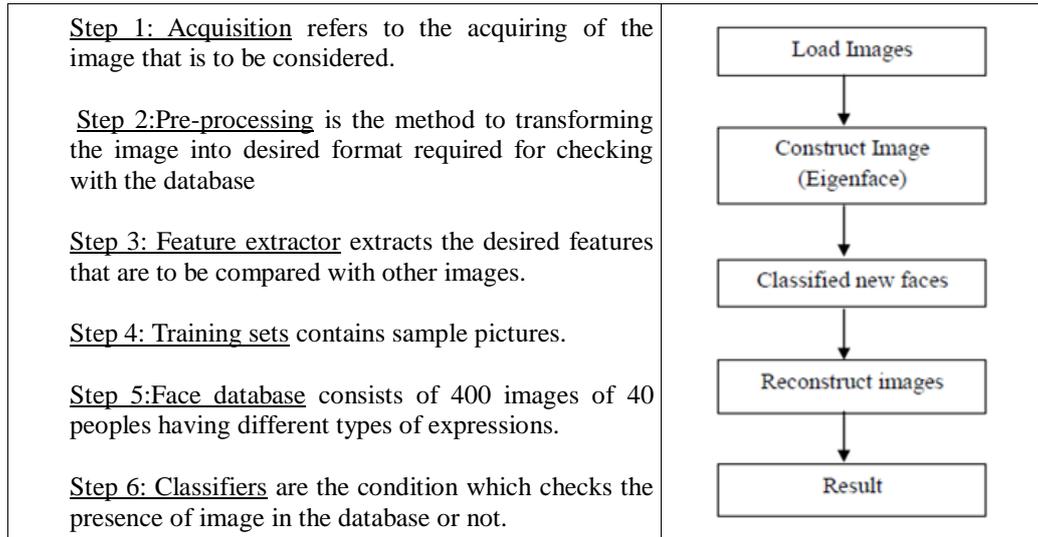
$\Phi_i = \Gamma_i - \Psi$. These differences are used to compute a covariance matrix (C)[5] for the dataset. The covariance between two sets of data reveals how much the sets correlate.

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = \frac{1}{M} \sum_{n=1}^M \begin{pmatrix} \text{var}(p_1) & \dots & \text{cov}(p_1, p_N) \\ \vdots & \ddots & \vdots \\ \text{cov}(p_N, p_1) & \dots & \text{var}(p_N) \end{pmatrix}_n = AA^T$$

Where $A=[\Phi_1\Phi_2\dots\Phi_M]$ and p_i = pixel i of face n .

The eigenfaces are simply the eigenvectors of C . However, since C is of dimension N (the number of pixels in the images), solving for the eigenfaces gets ugly very quickly.

III. FLOWCHART OF A FACE RECOGNITION PROGRAM:



IV. PROPOSED METHODOLOGY

4.1 Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**. Basically it has two phases

4.1.1 Training Phase

1. Each face in the database is represented as a column in a matrix A . The values in each of these columns represent [7] the pixels of the image and range from 0 to 255 for an 8-bit grayscale image:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

2. Next, the matrix is normalized by subtracting from each column a column that represents the average face (the mean of all the faces):

$$\bar{A} = \begin{pmatrix} a_{11} - m_1 & \dots & a_{1n} - m_1 \\ \vdots & \ddots & \vdots \\ a_{m1} - m_m & \dots & a_{mn} - m_m \end{pmatrix}$$

3. Then computation done for the covariance matrix of A , which is $A \times A^T$, but since the operation is very mathematically intensive, a shortcut is taken:

$$L = A^T X A$$

4. To obtain U , the matrix of covariance eigenvectors, V is found, the matrix of eigenvectors of L , and calculate:

$$U = A \times V.$$

5. Each face is then projected to face space:

$$\Omega = U^T \times A$$

6. Next the threshold value for comparison is computed:

$$\theta = \frac{1}{2} \times \max \{ \|\Omega_i - \Omega_j\| \}, \text{ for } i, j = 1 \dots n.$$

4.2.2 Recognition Phase

1. We represent the target face as a column vector:

$$r = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix}$$

2. The target face is then normalized:

$$\bar{r} = \begin{pmatrix} r_1 - m_1 \\ \vdots \\ r_m - m_m \end{pmatrix}$$

3. Next, the face is project to face space: $\Omega = U^T \times r$

4. We then find the Euclidean distance between the target projection and each of the projections [10]in the database:

$$\varepsilon^2 = \|\Omega - \Omega_i\|^2 \text{ for } i = 1 \dots n$$

5. Finally, we decide if the face is known or not by selecting the smallest distance and comparing it to the threshold θ . If it is greater, then the face is new. Otherwise, the face is a match.

4.2.3 Eigenfaces

In Eigen face method follows two steps which are given below in a table format.

<u>Eigenfaces Initialization</u>	<u>Eigenfaces Recognition</u>
1. Acquire an initial set of face images (the training set) .	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. Calculate the Eigenfaces from the training set, keeping only the M images that correspond to the highest eigenvalues.	2. Determine if the image is a face at all by checking to see if the image is sufficiently close to "face space."
3. Calculate the corresponding distribution in M dimensional weight space for each known individual, by projecting their face images onto the "face space."	3.(Optional) Update the Eigenfaces and/or weight patterns.
	4. If it is a face, classify the weight pattern as either a known person or as unknown

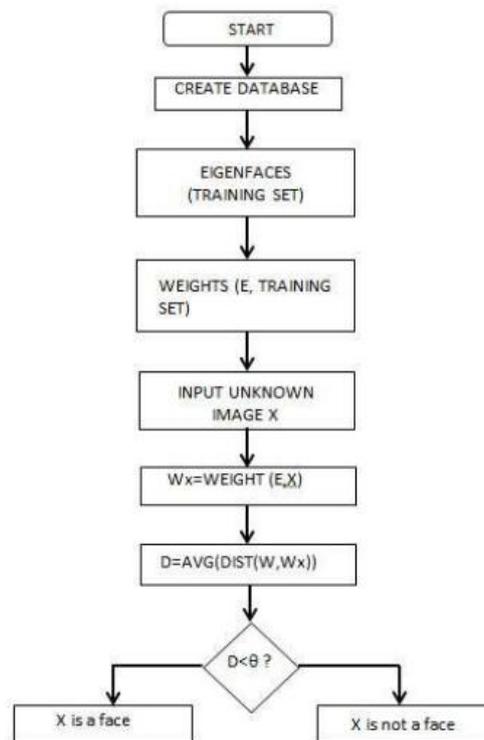


Fig 5.1 : Face recognition flowchart.

V. RESULTS AND ANALYSIS

Step I: Loading Database:

This step involves loading the database into the program in matrix v . The database is AT&T database which includes 400 images of different expression of 40 different peoples. The database is loaded in such a manner that once it is loaded it need not to be loaded twice thus saving a lot of time and space.

Step II: Initialization:

In this step we query an image from the database and use rest of the 399 images for training. We later use the selected picture to test the program.

Step III: Eigen vectors calculation:

Eigen vectors of correlation matrix are calculated here. At first we pick the eigen values of 400 eigenfaces, and then we pick the Eigen vectors of 10 largest eigen values. After this 20 different signatures are taken for each image.

Step IV: Recognition:

In this step mean of 400 images is taken and subsequently subtracted from each image. After that the eigen values of the selected image is compared with 399 different images and after that the image having the nearest Eigen values is considered to be the found image.

Step V: Result

This step is simply the result showing step. If the image match found then both the images(image that we selected for query and the matched image) will be displayed in a subplot. If no matching image is found then it will show a black screen. If matched image is found then another subplot is formed that gives the relevant information of the person from the available database in our system along with a guided speech system.

MATLAB Software has been extensively used for the study of **FACE RECOGNITION**.

Explanation of Face Recognition using Principal Component Analysis and Eigenface:

To explain Eigenfaces (Principal Component Analysis) in simple terms, Eigen faces figures out the main differences between all the training images, and then how to represent each training image using a combination of those differences. So for example, one of the training images might be made up of: (Average Face) + (13.5% of eigenface0) - (34.3% of eigenface1) + (4.7% of eigenface2) + ... + (0.0% of eigenface199).



Fig 7.1: Eigen Face image of first 31 images.

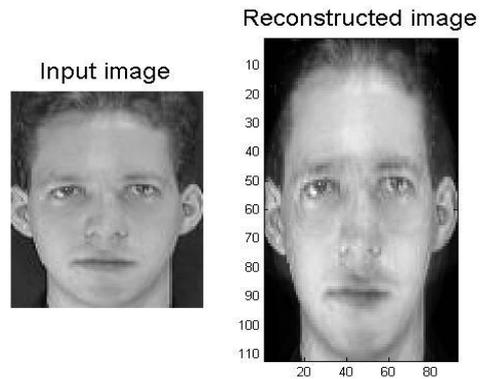


Fig 7.2: Reconstructed images

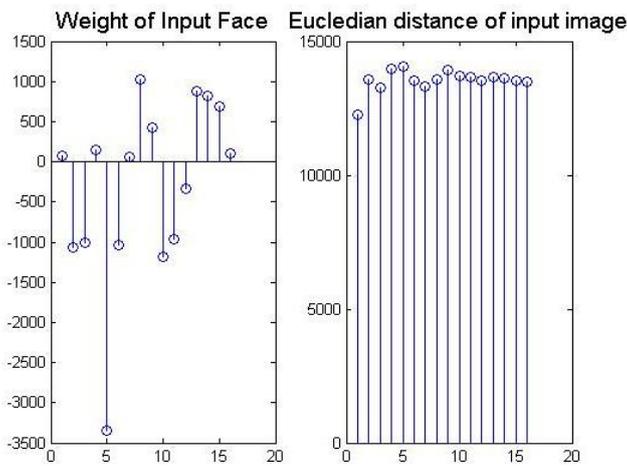


Fig 7.3: Weight and Euclidian distance of image

Pictorial Representation of project Execution

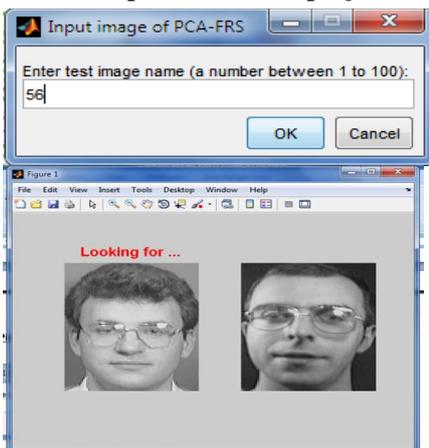


Fig 8.1: Image input prompt

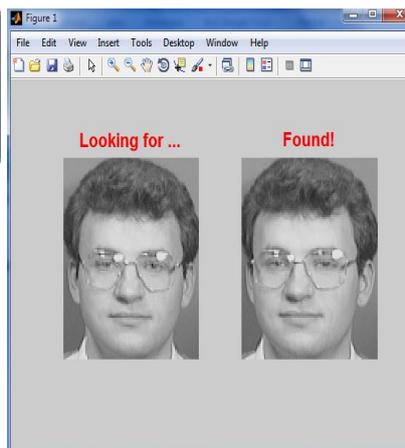


Fig 8.3: Image found confirmation
 Fig 8.2: Image checking window

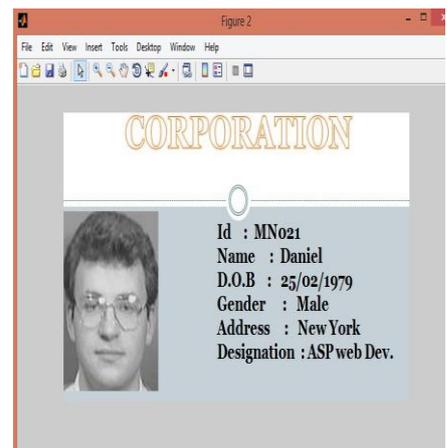


Fig 8.4: Relevant Information

REFERENCES

- [1] L. Sirovich, and M. Kirby, "Low-Dimensional Procedure for the characterization of Human Faces" IEEE Jour., pp. 519-524, 1987.
- [2] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," IEEE Transaction Pattern Analysis and Machine Intelligence, vol. 12, pp. 103-108, Jan. 1990.
- [3] M. Turk and A. Pentland, "Eigenfaces for recognition" J. of Cognitive Neuroscience, vol.3, no.1, pp. 71-86, 1991.
- [4] H. Murase and S. Nayar, "Visual learning and recognition of 3D objects from appearance," Int. Jour. of Computer Vision, pp. 5-24, 1995.
- [5] J. Lu, X. Yuan, and T. Yahagi, "A method of face recognition based on fuzzy c-means clustering and associated sub-NNs," IEEE Trans. on Neural Network, vol. 18, no. 1, pp. 150-160, Jan. 2007.
- [6] H. Zhao and P.C. Yuen, "Incremental linear discriminant analysis for face recognition," IEEE Transaction on System Man and Cybernetics B, vol. 38, no. 1, pp. 210-221, Feb. 2008.
- [7] Rabia Jafri and Hamid R. Arabnia, "A Survey of Face Recognition Technique," Jour. of Information Processing Systems, vol. 5, June 2009.
- [8] "Eigenfaces for recognition", M. Turk and A. Pentland, Journal of Cognitive Neuroscience, Vol.3, No.1, 1991.
- [9] J. Shermina and V. Vasudevan, "An Efficient Face Recognition System Based on the Hybridization of Invariant Pose and Illumination Process," European Journal of Scientific Research, 2011, pp. 225-243.
- [10] "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No.7, 1997, pp.711-720.