Different distance based PCA+LDA fusion Technique for Face recognition

Dhiren Pandit[1], Dr. Jayesh Dhodiya[2]
[1] *Assistant Professor, Science and Humanities Department, LDRP-ITR, Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat, India*
[1] *Research Scholar, Applied Mathematics and Humanities Department, SVNIT, Surat, Gujarat, India*
[2] *Assistant Professor, Applied Mathematics and Humanities Department, SVNIT, Surat, Gujarat, India*
[1]*dhirenpandit@gmail.com*
[2]*jdhodiya2002@yahoo.com*

***Abstract*—Since last few years, face Recognition has become one of the most challenging task in the pattern recognition field. The Face recognition plays very important role in many applications like video surveillance, retrieval of an identity from a database for criminal investigations and forensic applications. The face is considered as good biometric for many reasons: the acquisition process is nonintrusive and does not require collaboration of the subject to be recognized. The acquisition process of a face from a scene is simpler and cheaper than the acquisition of other biometrics as the iris and the fingerprint. On the other hand, many problems arise, because of the variability of many parameters like face expression, pose, scale, lighting, and other environmental parameters.**

Face recognition involved in application like problem of recognition of an identity in a scene. A system that automatically recognizes a face in a scene, first detects it and normalize it with respect to the pose, lighting and scale. Then, the system tries to associate the face to one or more faces stored in its database, and gives the set of faces that are considered as nearest to the detected face. This requires more computational resources and very robust algorithms for detection, normalization and recognition. In this paper we have implement different face recognition methods like Principle component analysis, Linear discriminant analysis and Fusion of PCA and LDA for face recognition. And better recognition rate is achieved by implementing different similarity measures between images.

I. Introduction

In recent years Many face recognition systems have been proposed. Each of them is based on a particular representation of a face. Mainly “appearance-based” approaches and “structural approaches” are used for face representation.
Methods of the first kind try to reduce the dimensionality of the original face space due to huge dimensionality of a face image and hence it may contain redundant or noisy information. A feature reduction is performed by applying some standard algorithms of pattern recognition. The most known approach is the PCA representation or “eigenface” approach, proposed by Turk and Pentland [9]: the face image is projected in a space in which the correlation among the components is zero. This space transformation is called “Karhunen-Loeve transform”. Another “appearance-based” approach is the LDA representation or “fisherface” approach, proposed by Kriegmann et al. [10]: the face image is projected in the Fisher space, in which the variability among the face-vectors of the same class is minimized and the variability among the face-vectors of different classes is maximized.

II. Principle Component Analysis for Face Recognition

Any particular face can be represented in terms of “eigenpictures”. Eigenpictures are eigenfunctions of the averaged covariance of the ensemble of faces. In other words, they showed that in principle, a collection of face images can be approximately represented by a small set of standard pictures with a small set of weights for each of the standard pictures.

A. Method of Principle component analysis

A face image, $I(x, y)$, is a two-dimensional N by N matrix of intensity values, which are usually quantized to 8- bit values. Each x and y pair denotes a position in the image. For the purpose of exposition, it is convenient to represent the matrix of intensity values as a vector, where each row is concatenated. Now, instead of having a matrix of dimension N by N, we have a vector of dimension N^2.
As an example, a typical image with size 220 by 220 pixels becomes a point in a 48400-dimensional space. To obtain the Eigenfaces for a training set, first determine the mean vector, deviation-from-mean vectors and the co-variance matrix for the particular training set. Let the images in the training set be represented by $\{T_1, T_2, T_3, \ldots, T_M\}$, where each T_n is a vector of N^2-dimension. The value M is the number of images in the training set. With this representation, the mean vector is:

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

The set of deviation-from-mean vectors, $\{\Phi_1 \Phi_2 \Phi_3 \ldots \Phi_M\}$ contains the individual difference of each training image from the mean vector. Kirby and Sirovich refer to these vectors as caricatures. They are simply defined as:

$$\Phi_i = T_i - \Psi \tag{2}$$

As described previously, the Eigenfaces are the set of principal components of the training set. To obtain the eigenface description of the training set, the training images are subjected to Principal Component Analysis (PCA), which seeks a set of vectors (the principal components)which significantly describes the variations of the data. Mathematically, the principal components of the training set are the eigenvectors of the covariance matrix of the training set [5]. The covariance matrix is given by:

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \tag{3}$$

It is clear from this matrix that we are interested in finding the set of vectors u_k and scalars λ_k that satisfy the relations

$$C u_k = \lambda_k u_k \tag{4}$$

$$u_1^T u_k = \begin{cases} 1 & \text{if } l = k, \\ 0 & \text{if } l \neq k. \end{cases} \tag{5}$$

It is clear from 5 that the vectors u_k are orthonormal. Another way of representing the covariance matrix is by writing

$$A = \{\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_M\} \tag{6}$$

$$C = \frac{1}{M} A A^T \tag{7}$$

A closer look at 7 reveals that matrix C has a dimension of N^2 by N^2, and determining N^2 eigenvectors and eigenvalues from a matrix this large (48400 by 48400 for example) is unwieldy. Furthermore, the purpose of employing PCA in the first place is to obtain a low dimensional representation that can briefly describe the training set, and using N^2 eigenvectors for that will defeat the purpose. In fact, if the number of data points in the image space for which we wish to find a compact representation is less than the dimension of the image space $(i.e. M << N^2)$, only $M-1$ eigenvectors will be meaningful. To circumnavigate the problem, Turk and Pentland proposed the following solution. Consider the eigenvectors v_i of $A^T A$ such that

$$A^T A v_i = \mu_i v_i \tag{8}$$

The scalars μ_i are the corresponding eigenvalues of vi. Multiplying $\frac{1}{M} A$ from the left for both sides of the equation yields

$$\frac{1}{M} A A^T A v_i = \frac{1}{M} \mu_i A v_i \tag{9}$$

$$C A v_i = \frac{1}{M} \mu_i A v_i \tag{10}$$

This implies that Av_i are the eigenvectors of the covariance matrix. With this treatment, we have effectively reduced the dimension of the matrix on which we have to work on from N^2 by N^2 to M by M.
Using this method, firstly construct the matrix $L = A^T A$ of M by M dimensions and find the M eigenvectors, v_i, of L. The first M eigenvectors of the covariance matrix can be obtained by finding Av_i, and the corresponding eigenvalues allow us to rank the eigenvectors according to their significance. As described in detail previously, these eigenvectors are termed Eigenfaces.
Each element of the training set $\{T_1, T_2, T_3, \ldots, T_M\}$ is projected onto "face space" by the following operation

$$\omega_k = (A v_k)^T (T_i - \Psi) ; 1 \leq k \leq M, 1 \leq i \leq M \tag{11}$$

Therefore, for each face image in the training set, we would have a set of M weights, $\Omega_i = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_M\}$, $1 \leq i \leq M$, which describes the contribution of each Eigenface to the face image.

B. *Classifying a Face Image*

With each training image represented by the set of weights, standard pattern recognition methods can be used to classify input images into known identity classes. For this case, the Euclidean distance was used as the measure for classification. Before the value can be calculated, the test image, T, has to be projected onto the face space as well, using equation 11 , yielding the set Ω_p. The test image is assigned to the class k which minimizes.
Since recognition is performed by projection first, any image similar-sized can be fed into the system. Images of individuals not previously seen in the training set, as well as non-face images, can be projected onto face space, yielding the set of weight Ω_p. Hence, a competent face recognition must be able differentiate between a face image and non-face image, and if a face image is received, whether it corresponds one or none of the individuals in the training set. For this purpose, the distance between the input image and face space, is proposed by Turk and Pentland to countercheck whether an input image is indeed a face image.

$$\varepsilon_F^2 = \|\Phi_P - \Phi_i\|^2 \tag{12}$$

with

$$\Phi_P = T_p - \Psi \tag{13}$$

$$\Phi_I = \sum_{i=1}^{M} \omega_i (A v_i) \tag{14}$$

The value of Φ is simply the reconstructed image of the projection of the input image onto the face space spanned by the eigenvectors.

For the system trained with the set in IndianFace Database [4]. These faces were carefully chosen to have neutral expression as well as the same lighting conditions. They were then manually centered and cropped to be of the same size. After a PCA was performed on these points, it was found that the first principal component was sufficient to capture the major variations among the points, i.e. all points can be discriminated based on their projections onto the first principal component.

From equations 12, 13 and 14, equation of the value of distance between input image and face space can be rewrite as,

$$\varepsilon_F^2 = |T_p - T_l|^2 \tag{15}$$

with

$$T_l = \left(\sum_{i=1}^{M} \omega_i (Av_i) + \Psi \right) \tag{16}$$

From equation 16, we can see that T_l is the reconstruction of the projection of T_P onto the first M' Eigenfaces. The first M' Eigenface were found to be able to account for more than 90 % of the variations in the training set, and the reconstruction is very good approximation of T_l if the image has a position in the image space close to the subspace defined by the Eigenfaces [15]. This means that as long as an input image lies near the subspace defined by the Eigenfaces, regardless of whether the position of the image in the image space R^L (with $L = N^2$) is close to the positions of the face images, T_l and will be fairly similar, and will have a small value. This causes face space distances defined by equation 12 to be close for face.
A better measure for face space distance would be to use deviation from mean directly. As it was reported in [9] that "Images of faces, being similar in overall configuration, will not be randomly distributed in the huge image space", It can be conjectured that face images are situated near the average face. Therefore, it can be simply used as a measure of face space distance:

$$\varepsilon_F^2 = \|T_P - \Psi\|^2 \tag{17}$$

Based on the training set in IndianFace Database, a threshold can be established to differentiate face images.

III. Linear Discriminant Analysis (Fisherface approach)

Fisher faces method [7] derives from Fishers linear disriminant analyis (FLD or LDA); it works on the same principle as the eigenfaces method.
For appearance-based face recognition, a 2Dface image is viewed as a vector with length N in the high dimensional image space. The training set contains M samples $\{x_i\}_{i=1}^{M}$ belonging to C individual classes $\{x_j\}_{j=1}^{C}$.

LDA tries to find a set of projecting vectors w best discriminating different classes. According to the Fisher criteria, it can be achieved by maximizing the ratio of determinant of the between-class scatter matrix S_b and the determinant of the within-class scatter matrix S_W.
The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible by finding direction along which the classes are best separated. In the Fisherface method [1], the face data is first projected to a PCA subspace spanned by M-C largest eigenfaces.

A. Training Phase

LDA finds the vectors in the underlying space that best discriminate among classes [1][7]. For all samples of all classes, the between class scatter matrix Sb and the within class scatter Sw are defined by

$$S_b = \sum_{i=1}^{C} q_i \left(\Psi_{ci} - \Psi\right)^T \left(\Psi_{ci} - \Psi\right) \tag{18}$$

$$S_w = \sum_{i=1}^{C} \sum_{\Gamma_k \in C_i} \left(\Gamma_k - \Psi_{ci}\right)^T * \left(\Gamma_k - \Psi_{ci}\right) \tag{19}$$

q_i is the number of training samples in class i, C the number of distinct class.
Ψ_{ci} is the mean vector of samples belonging to class i defined by the equation:

$$\Psi_{ci} = \frac{1}{q_i} \sum_{k=1}^{q_i} \Gamma_k \tag{20}$$

$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n$ is the mean of the set of training images. We used matrix of dimension $N * M$ in equation 18 and 19 to calculate the within class scatter of dimension $(M * M)$ that deals with covariance between individuals.

The within class scatter S_W represents how face images are distributed closely within classes and between class scatter matrix S_b how classes are separated from each other [2].

The goal of LDA is to maximize S_b while minimizing S_W; the images in the training set are divided into the corresponding classes. LDA finds a set of vectors w such that the fisher discriminant criterion is maximized.

$$w = argmax_T(J(T))$$

$$max(J(T)) = \frac{|T^T S_b T|}{|T^T S_w T|} T = w \tag{21}$$

w can be constructed by calculating the eigenvectors of the matrix $S_w^{-1} S_b$

$$w = eig(S_w^{-1} S_b) \tag{22}$$

When face images are projected into the discriminant vectors w, face images should be distributed closely within classes and should be separated between classes as much as possible. These eigenvectors are called the fisher faces [2]. Fisherface approach is similar to eigenface approach, which makes use of projection of training images into a subspace.

B. Recognition phase:

Given a test image (Γ_x), where the mean image Ψ is subtracted $\Gamma_x - \Psi$ and the result ϕ_t is projected onto the face space and identified using the euclidean distance as a similarity measure. $g(\phi_t) = W\phi_t W^T$ The face which has the minimum distance with the projected face images is labeled with the identity of that image. The same procedure is established in PCA method for calculating the minimum distance to find the corresponding face class k that minimizes the Euclidean distance in fisher space.
Face recognition systems using LDA/FLD have also been very successful (Belhumeur et al [1]; Swets and Weng [6]); Zhao et al [13][14]. Zhao et al [13][14] describes the LDA approach for face recognition using the class probability: the face image is projected from the original vector space to a face subspace via Principal Component Analysis where the subspace dimension is carefully chosen, the LDA is used to obtain a linear classifier in the subspace. In addition, a weighed Euclidean distance metric is employed to improve the performance of the subspace LDA method.

Two or four training samples per person are available; LDA training is carried out via scatter matrix analysis [13]. For M class problem, the within and between-class matrices S_W and the S_b are computed as follows:

$S_W = \sum_{i=1}^{M} Pr(C_i)(m_i - m_o)\cdot(m_i - m_o)^T$ Where $Pr(C_i)$ is the prior class probability and usually replaced by $\frac{1}{M}$ in practice with the assumption of equal priors. Here S_W is the within-class matrix showing the average Scatter of the sample vector x of different classes C_i around their respective means m_i

IV. Fusion of PCA and LDA for face recognition

In this section we present methodology for fusing two appearance-based (or statistical) approaches to face recognition: the PCA representation ("eigenface" approach) and the LDA representation ("fisherface" approach). This composed of the following steps:

- representation of the face according to the PCA and the LDA approaches;
- The distance vectors d^{PCA} and d^{LDA} from all the N faces in the database are computed;
- For the final decision, these two vectors are combined according to a given combination rule. We propose algorithm for the fusion phase: the K-Nearest Neighbours.

A. Method of fusion of PCA and LDA

Let X be a d-dimensional feature vector. In our case, d is equal to the number of pixel of each face image. The high dimensionality of the related "image space" is a well-known problem for the design of a good verification algorithm. Therefore, methods for reducing the dimensionality of such image space are required. To this end, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are widely used. Principal Component Analysis [9] [12] is defined by the transformation:

$$y_i = W^T x_i \tag{23}$$

Where $x_i \in X \subseteq R^d$, $i = 1, 2, \dots, n$ (n samples). W is a d-dimensional transformation matrix whose columns are the eigenvectors related to the eigenvalues computed according to the formula:

$$\lambda e_i = S e_i \tag{24}$$

S is the scatter matrix (i.e., the covariance matrix):

$$S = \sum_{i=1}^{n} (x_i - m).(x_i - m)^t; m = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{25}$$

This transformation is called Karuhnen-Loeve transform. It defines the d-dimensional space in which the co-variance among the components is zero. In this way, it is possible to consider a small number of "principal" components exhibiting the highest variance. In the face space, the eigenvectors related to the most expressive features are called "eigenfaces". The Linear Discriminant Analysis is defined by the transformation:

$$y_i = W^T x_i \tag{26}$$

The columns of W are the eigenvectors of $S_W^{-1} S_b$, where S_W is the within-class scatter matrix, and S_b is the between-class scatter matrix. It is possible to show that this choice maximizes the ratio $\frac{det(S_b)}{det(S_W)}$. These matrices are computed as follows:

$$S_W = \sum_{j=1}^{c} \sum_{i=1}^{n_j} (x_i^j - m_j).(x_i^j - m_j)^T; m_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^j \tag{27}$$

Where x_i^j is the i-th pattern of j-th class and n_j is the number of patterns for the j-th class.

$$S_b = \sum_{j=1}^{c} (m_j - m)(m_j - m)^T; m = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{28}$$

The eigenvectors of LDA are called "fisherfaces". LDA transformation is strongly dependent on the number of classes (c), the number of samples (n), and the original space dimensionality (d). It is possible to show that there are almost $c - 1$ nonzero eigenvectors. $c - 1$ being the upper bound of the discriminant space dimensionality. We need $d + c$ samples at least to have a nonsingular S_W . It is impossible to guarantee this condition in many real applications. Consequently, an intermediate transformation is applied to reduce the dimensionality of the image space. To this end, we used the PCA transform.

Many works analysed the differences between these two techniques [10], but no work investigated the possibility of fusing them. Here it should be noted that LDA and PCA are not so correlated, as the LDA transformation applied to the principal components can generate a feature space significantly different from the PCA one. Therefore, the fusion of LDA and

PCA for face recognition and verification is worth of theoretical and experimental investigation. We propose following approach to fuse PCA and LDA face representations: the K-Nearest Neighbour approach (KNN).
First of all, we normalise the distance vectors d^{PCA} and d^{LDA} in order to reduce the range of these distances in the interval [0,1]. The second step is to compute a combined distance vector d that must contain both PCA and LDA informations. To this aim, we followed this way, we obtained the combined distance vector by computing the mean vector:

$$d = \left\{ \frac{d_1^{PCA} + d_1^{LDA}}{2}, \ldots, \frac{d_N^{PCA} + d_N^{LDA}}{2} \right\} \tag{29}$$

where N is the number of images in the face database. After computing and ordering the combined distance vector d, we follow the KNN decision: the most frequent identity among the first K components of d is selected. The combined distance vector follows 29, we call our algorithm "M-KNN" or "Mean-KNN".

V. Introduction to Different Distances

Distance measures are used to compute the difference between two vectors. The Face Identification Evaluation System includes many common distance measures that are used to compute the similarity between two images. Some of them are describe here. In the definitions of the distance measures in the following subsections, let u and v be vectors representing arbitrary images in PCA or LDA space.
The following are the different distances [11] to measure similarity between two images

1) CityBlock

$$D_{CityBlock} = \sum_i |u_i - v_i| \tag{30}$$

2) Euclidean

$$D_{Euclidean}(u,v) = \sqrt{\sum_i (u_i - v_i)^2} \tag{31}$$

3) Correlation

$$D_{Correlation}(u,v) = \frac{\sum_i (u_i - \bar{u})(v_i - \bar{v})}{(N-1)\sqrt{\frac{\sum_i (u_i-\bar{u})^2}{N-1}}\sqrt{\frac{\sum_i (v_i-\bar{v})^2}{N-1}}} \tag{32}$$

4) Covariance

$$D_{Covariance}(u,v) = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}} \tag{33}$$

5) Mahalinobis CityBlock

$$D_{Mahl1}(u,v) = \sum_i |m_i - n_i| \tag{34}$$

6) Mahalinobis Euclidean

$$D_{MahL2}(u,v) = \sqrt{\sum i(m_i - n_i)^2} \tag{35}$$

7) Cosine

$$D_{Cos}(u,v) = \frac{u.v}{\|u\|.\|v\|} \tag{36}$$

8) Mahalinobis Cosine

$$D_{MahCosine}(u,v) = \frac{m.n}{|m||n|} \tag{37}$$

9) Hellinger

$$D_{Hellinger}(u,v) = \sqrt{\sum_i \left(\sqrt{|u_i|} - \sqrt{|v_i|}\right)^2} \tag{38}$$

10) Canberra

$$D_{Canberra}(u,v) = \sum_i \frac{|u_i - v_i|}{|u_i + v_i|} \tag{39}$$

Among these all distances we have implemented following three distances for face recognition methods Principle component analysis(PCA), Linear discriminant Analysis(LDA) and Fusion of PCA and LDA.

A. Euclidean distance

Euclidean distance is usual distance between two vectors which can be measured using following

$$D_{Euclidean}(u,v) = \sqrt{\sum_i (u_i - v_i)^2} \tag{40}$$

B. Cosine similarity

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0 degree is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90 degree have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1]. These bounds apply for any number of dimensions, and Cosine similarity is most commonly used in high-dimensional positive spaces[8].

$$D_{Cos}(u,v) = \frac{u.v}{\|u\|.\|v\|} \tag{41}$$

C. Mahalinobis Cosine distance

The first step in computing Mahalinobis based distance measures is to understand the transformation between image space and Mahalinobis space. PCA is used to find both the basis vectors for this space and the sample variance along each dimension. The output of PCA are eigenvectors that give rotation into a space with zero sample covariance between dimensions, and a set of eigenvalues that are the sample variance along each of those dimension. Mahalinobis space is defined as a space where the sample variance along each dimension is one. Therefore, the transformation of a vector

from image space to feature space is performed by dividing each coefficient in the vector by its corresponding standard deviation. This transformation then yields a dimensionless feature space with unit variance in each dimension[8].
Here we will deal with the similarity between two vectors we will define two vectors u and v in the unscaled PCA space and corresponding vectors m and n in Mahalinobis space. First, we define $\lambda_i = \sigma_i^2$ where λ_i are the PCA eigenvalues, σ_i^2 is the variance along those dimensions and σ_i^2 is the variance along those dimensions and σ_i is the standard deviation. The relationship between the vectors are then defined as $m_i = \frac{u_i}{\sigma_i}$ and $n_i = \frac{v_i}{\sigma_i}$.

Mahalinobis Cosine is the cosine of the angle between the images after they have been projected into the recognition space and have been further normalized by the variance estimates. So, for images u and v with corresponding projections m and n in Mahalinobis space, the Mahalinobis Cosine is:

$$S_{MahCosine}(u,v) = cos(\theta_{mn}) = \frac{|m||n|cos(\theta_{mn})}{|m||n|} = \frac{m.n}{|m||n|}$$

$$D_{MahCosine}(u,v) = -S_{MahCosine}(u,v) \quad (42)$$

VI. Algorithm for Fusion of PCA and LDA with different distances

In this fusion method on different databases first we employ principle component analysis to find eigenfaces and distance of eigenfaces with test image. After that we implement linear discriminant analysis on the same databases for the same test image and find distance of test images with other fisherfaces. Once both distances are obtained we will take average distance and then different measures are used as similarity measures to determine the closest match for the test image with the face in the trained database. The steps involved in implementing the algorithm are:

Step 1 Implement Principle component analysis method to find distance of eigenfaces with test image.

step 1.1 Represent the faces in the database in terms of the vector X as

$$X = \{X_1, X_2, \ldots, X_N\} \quad (43)$$

Where each X_i is a face vector of dimension N obtain from the $M \times N$ dimension face image.

step 1.2 From each of the face image vectors the average face is subtracted. The average face is given by

$$AvgFace = \frac{1}{N}\sum_{N} X_i \quad (44)$$

$$X^{'} = X - AvgFace \quad (45)$$

step 1.3 Classify the images based on the number of unique subjects involved. So the number of classes, C, will be the number of subjects who have been imaged.

step 1.4 Compute the eigenvectors of the scatter matrix. Retain only the K eigenvectors corresponding to the K largest eigenvalues

step 1.5 The face image vector X' is then projected onto the eigenvectors using $Y = W^T X'$. The values of Y are the feature vectors or weights of the images. Each of the face images can be represented in terms of these feature vectors. Second part is Recognition

step 1.6 For a test face image to be recognized, initially the normalization is performed by subtracting the average face from the image : $X' = X - AvgFace$

step 1.7 Then the face is projected on the basis vector : $Y = \sum_K W_i^T X' W_i$, where Y gives the weight of the test image

step 1.8 Compare the weight obtained Y with the values of weights recorded from the training phase. This comparison is performed using different distance metrics and find distance vector d^{PCA}

step 2 Implement Linear discriminant analysis method to measure distance of test image with fisherfaces. Steps are given in section 2.2. using which find distance vector d^{LDA} is computed.

step 3 To find a combined distance vector d that must contain both PCA and LDA informations, we compute the mean vector

$$d = \left\{ \frac{d_1^{PCA} + d_1^{LDA}}{2}, \ldots, \frac{d_N^{PCA} + d_N^{LDA}}{2} \right\}.$$

step 4 After computing and ordering the combined distance vector d, we follow the Nearest Neighbor decision: the most frequent identity among the first K components of d is selected.

Here we have implemented Principle component analysis, Linear discriminant analysis and fusion of PCA and LDA (PCA+LDA) using following distances.

- First we have used standard Euclidean distance to measure similarity between test image and images from train database.
- We have checked result for verification of face using Cosine similarity between test image and images from train database.
- Finally we have use Mahalinobis cosine distance to check similarity between test image and images from train database.

VII. Results

A. Data

Here we have used the standard computer vision data set, it contains frontal images of 395 individuals, and each person has 20 frontal images [16]. This data set contains images of people of various racial origins, mainly of first year undergraduate students, so the majority of individuals are between 18-20 years old but some older individuals are also present. Some individuals are wearing glasses and beards. The total number of images is 7900. In our experiments, ten face images are selected for training and reference, and five for testing.

Fig. 1. Some examples of faces from database

	PCA	LDA	PCA+LDA
Euclidean Distance	46(92%)	46(92%)	46(92%)
Cos Distance	48(96%)	46(92%)	**50(100%)**
MahCos Distance	48(96%)	46(92%)	**50(100%)**

TABLE I. RECOGNITION IN PERCENTAGE

B. Training

To train above algorithms, we used M=100 images of C=10 classes (different persons). Each class contains 10 frontal images. For the ten classes; the images were taken at different times, varying the lighting, facial expressions and facial details (glasses/no glasses). Some examples are shown in following figure.

The following table shows the percentage accuracy of our approaches on face94 data set.

VIII. CONCLUSION

The fusion of two approaches, namely PCA and LDA, for face representation and recognition have been investigated. Reported results confirm the benefits in fusing them. We combined PCA and LDA with the KNN-based combination rule. Reported results are strongly dependent on the data set.

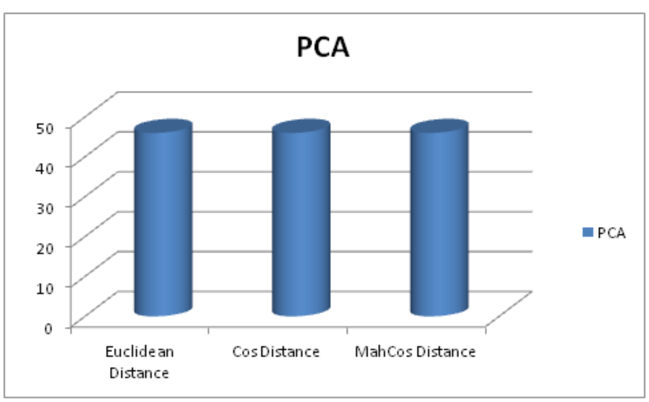

Fig. 2. PCA algorithm with Different Distances

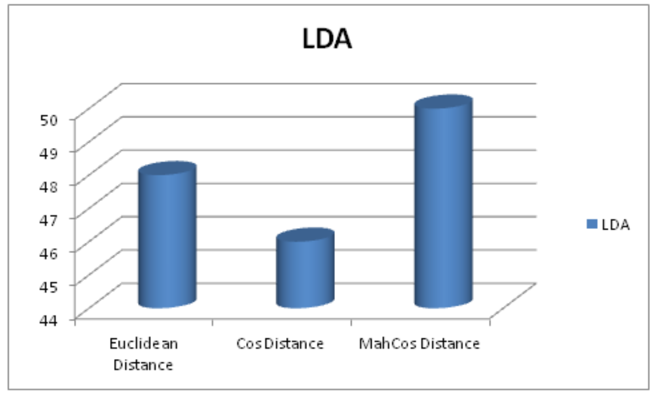

Fig. 3. LDA algorithm with Different Distances

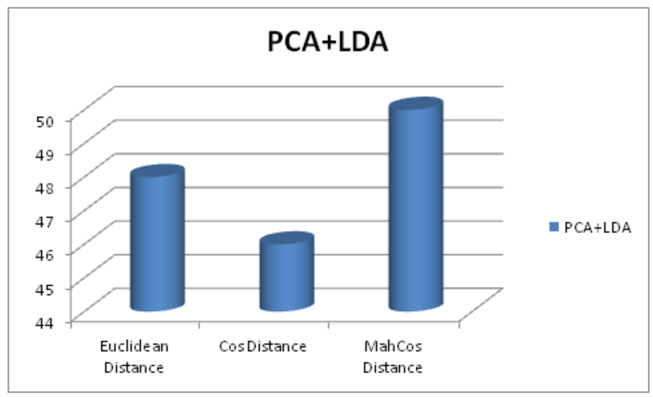

Fig. 4. PCA+LDA algorithm with Different Distances

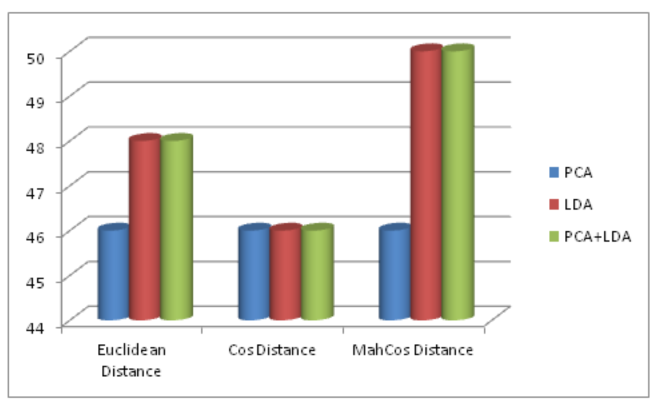

Fig. 5. PCA, LDA and PCA+LDA with different Distances

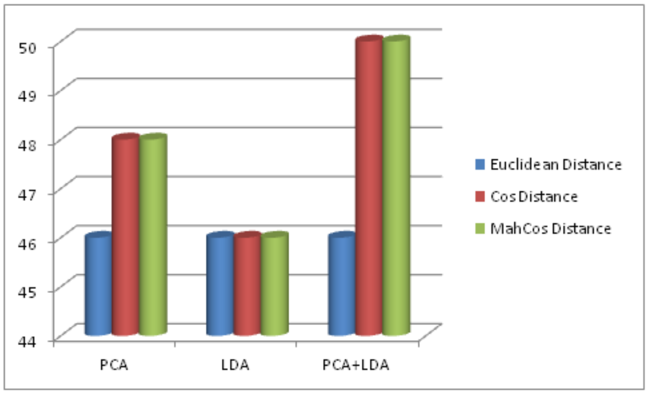

Fig. 6. Euc, Cos, MahCos Distances for PCA, LDA and PCA+LDA

Along with fusion we tried to apply this fusion technique using three different similarity measures namely Euclidean, Cosine and Mahalinobis Cosine and we conclude that Mahalinobis cosine and cosine works well for this fusion technique.

On the basis of the reported results it is worth devoting further theoretical and experimental investigations to understand the behavior of PCA and LDA in order to combine them.

References

[1] Belhumeur P., Hespanha J., Kriegman D., “Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, (July 1997) 711-720, 0162-8828.

[2] Eleyan A. Demirel H. “Face Recognition System based on PCA and Feedforward Neural Networks”, Proceedings of Computational Intelligence and Bioinspired Systems, pp. 935-942, 978-3-540-26208-4, Spain, June 2005, Springer-Verlag, Barcelona.

[3] Hu C., Chang Y., Feris R., Turk M.: Manifold based analysis of facial expression. Computer Vision and Pattern Recognition Workshop (2004)

[4] IIT Kanpur Database,http://viswww.cs.umass.edu/ vidit/IndianFaceDatabase/(7)

[5] Jie Wang, K.N.Plataniotis, A.N. Venetsanopoulos, Selecting discriminant eigenfeaces for face recognition, Pattern Recognition Letters, 26, 2005, 1470-1482.

[6] John J. Weng and Daniel L. Swets, “Face Recognition”, Book Biometrics, page 65-86, editeur springer US, ISBN 978-0-387-28539-9, 1999

[7] Kresimir delac, Mislav Grgic, Sonja Grgic, “independant comparative Study of PCA, ICA and LDA on the FERET data Set”, 2006 wiley periodicals,Inc. vol15,p252-260

[8] Kuldeep Singh Sodhi and Madan Lal, Face Recognition Using PCA, LDA and Various Distance Classifiers,Journal of Global Research in Computer Science, Volume 4, No. 3(2013)

[9] M. Turk, and A. Pentland, Eigenfaces for Face Recognition, Journal of Cognitive Neuroscience, vol. 3 num.1 (1991) 71-86.

[10] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE Trans. on PAMI, vol.19 num.7 (1997) 711-720.

[11] Philip Miller and Jamie Lyle, The Effect of Distance Measures on the Recognition Rates of PCA and LDA Based Facial Recognition, Digitial Image Processing Clemson Universtiy

[12] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, John Wiley & Sons, USA (2001).

[13] W. Zhao,R.Chellappa and P.J.Phillips , “Subspace Linear Discriminant Analysis for Face Recognition”, April 1999. Tech. rep. CAR-TR-914, Center for Automation Research, University of Maryland, College Park, MD.

[14] W.Zhao, R.Chellappa, J.Phillips and A.Rosenfeld , “Face recognition in still and video images:A litterature survey”, ACM comput surv 35(2003),399-458

[15] Y.V.Lata,ChandraKiran haradwajTungathurthi, H.RamMohanRao, Dr.AGovardhan, Dr.L.P.Reddy,Facial Recognition using Eigenfaces by PCA, International Journal of Recent Trends in Engineering, Vol.1,No.1,May2009 Face recognition by LDA