International Journal of Advanced Computer Research (ISSN (Print): 2249-7277 ISSN (Online): 2277-7970)
Volume-4 Number-4 Issue-17 December-2014

Study of Face Recognition Techniques

Sangeeta Kaushik[1*], R. B. Dubey[2] and Abhimanyu Madan[3]

Received: 10-November-2014; Revised: 18-December-2014; Accepted: 23-December-2014

Abstract

A study of both face recognition and detection techniques is carried out using the algorithms like Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Linear Discriminant Analysis (LDA) and Line Edge Map (LEM). These algorithms show different rates of accuracy under different conditions. The automatic recognition of human faces presents a challenge to the pattern recognition community. Typically, human faces are different in shapes with minor similarity from person to person. Furthermore, lighting condition changes, facial expressions and pose variations further complicate the face recognition task as one of the difficult problems in pattern analysis.

Keywords

Face recognition, face detection, Principal Component Analysis, Kernel Principal Component Analysis, Linear Discriminant Analysis and Line Edge Map.

1. Introduction

Face recognition is a very challenging task for the researches. Due to the difficulty of the face recognition task, several approaches have been used. The mainly used techniques are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Kernel Principal Component Analysis (KPCA), Line Edge Map (LEM) and Locality Preserving Projection (LPP). Face detection is an important part of face recognition as the first step of automatic face recognition.

*Author for correspondence

Sangeeta Kaushik, ECE dept., Hindu College of Engg., Sonepat, Haryana, India.
R. B. Dubey, ECE dept., Hindu College of Engg., Sonepat, Haryana, India.
Abhimanyu Madan, ECE dept., Hindu College of Engg., Sonepat, Haryana, India.

However, face detection is not straight forward because it has lots of variations of image appearance, such as pose variations, occlusion, image orientation, illuminating condition and facial expression.

A semi-automated system for face recognition required to locate features (such as eyes, ears, nose and mouth) on the photographs before it calculated distances and ratios to a common reference point which were then compared to reference data.
Goldstein, Harmon, and Lesk [1] used specific subjective markers such as hair color and lip thickness to automate the recognition. The only problem with both of these solutions was that the parameters were manually computed. Kirby and Sirovich [2] presented principle component analysis to the face recognition problem. Turk and Pentland [3] discovered that while using the eigen faces techniques, the residual error could be used to detect faces in images.

The edge map extraction from an input image and then matching it to a large template, with possible variations in position and size, was first introduced by Sakai et al. [4]. Kelly [5] introduced an improved edge detector, involving heuristic planning to extract an accurate outline of a person's head from various backgrounds. Govindaraju et al. [6] introduced a new method for cluttered image, quit similar to ones given by Yuille et al. [7].

Other techniques are based on hierarchical coarse-to-fine searches with template-based matching criteria [8-9]. The next step after a face has been located is computation of its features [10]. Brunelli and Poggio [11] introduced a comparision of feature-based methods to holistic approaches. An approach to face recognition uses the Karhunen-Loeve transform (KLT), which exhibits pattern recognition properties largely overlooked initially because of the complexity involved in computation. The KLT for face recognition introduced by Turk and Pentland [12] considered only a few KLT coefficients to represent faces and performed well for frontal mug shot images.

Akamatsu et al. [13] added operations to the KLT method to standardize faces with respect to position and size. A holistic approach to facial recognition was based on LDA Swets and Weng [14]; Belhumeur et al. [15]. Fisher's linear discriminant Duda and Hart [16] is used to obtain the most discriminating features of faces rather than the most expressive ones given by KLT alone Swets and Weng [14]. Other technique is based on the information theory which breaks down facial images into a small set of eigen faces was introduced by Sudhanshu et al. [17].

Artificial face recognition is one of the popular areas of research in image processing. It is different from other biometric recognition because faces are complex, multidimensional and almost all human faces have a similar construction [18]. A face image descriptor called Gabor ordinal measure is proposed by Z. Chai et al. [19]. The two discriminant analysis methods i. e; fractional step mixture subclass discriminant analysis and kernel mixture subclass discriminant analysis are proposed in [20]. A novel L1-norm discriminant analysis presented by W. Zheng et al [21] is a theoretical framework of Bayes error bound. This is based on the new discriminant criterion L1-LDA (linear discriminant analysis) method for linear feature extraction.

2. Proposed Methodology

2.1 Principle component analysis (PCA)

The Principal Component Analysis (PCA) is one of the most important techniques used in image recognition and compression. PCA is a mathematical tool which uses an orthogonal technique to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The purpose of using PCA for face recognition is to express the large 1-D vector of pixels from 2-D facial image into the compact principal components of the feature space. This is named as eigen space projection. Eigen space is calculated by finding the eigen vectors of the covariance matrix of facial images. Normally, it is very difficult to choose a suitable threshold [2].

2.1.1 Mathematical Analysis

Step 1: Get some data

A self-made data set is used which has got only two dimensions to show the PCA analysis at each step.

Step 2: Subtract the mean

For PCA to work properly, one has to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension.

Step 3: Calculate the covariance matrix.

Step 4: Calculate the eigen vectors and eigen values of the covariance matrix

Since the covariance matrix is square, we can calculate the eigen vectors and eigen values for this matrix. These are rather important, as they tell us useful information about our data. In the mean time, here are the eigen vectors and eigen values; It is important to notice that these eigen vectors are both unit eigen vectors. Their lengths are both 1. This is very important for PCA, but luckily, most maths packages, when asked for eigenvectors, will give unit eigen vectors.

Step 5: Choosing components and forming a feature vector

In general, once eigen vectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance.

Step 6: Deriving the new data set

This is the final step in PCA, and is also the easiest. Once we have chosen the components (eigen vectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed. It will give us the original data solely in terms of the vectors [2].

2.1.2 Eigen faces for recognition

Every face image is viewed as a vector. If image width and height are w and h pixels respectively, the numbers of the components of this vector will be w*h. Each pixel is coded by one vector component [15]. The rows of the image are placed each beside one another, as shown on Fig.1 (a).

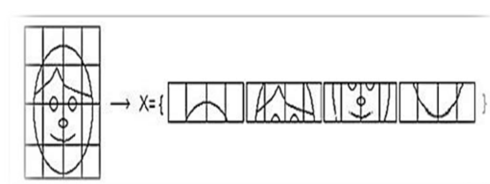

Figure 1(a): Formation of face vector from face image

2.1.3 Image space

This vector belong to a space, this space is the image space, the space of all images whose dimension is i by w*h pixels [15]. The basis of the image space is composed of the following vectors Fig. 1(b).

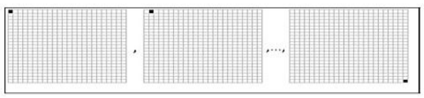

Figure 1(b): Image space

All the faces look like each other. They all have two eyes, a mouth, a nose, etc. located at the same place. Therefore, all the face vectors are located in a very narrow cluster in the image space, as shown in the Fig. 1(c).

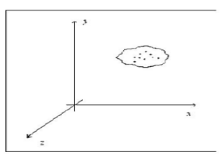

Figure 1(c): Image and face cluster

The full image space is not optimal space for face description. The basis vectors of this space are called principle components. Linear algebra is used to find principle components of the distribution of the faces, or eigenvectors of the covariance matrix of the set of face images. These eigen vectors can be thought as set of features which characterize the variation between face images [18]. Each of the images contributes more or less eigen vectors, some of these faces are shown in Fig. 1(d).

Figure 1(d): Shows schematically various steps of PCA

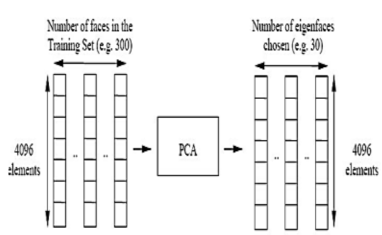

Figure 1(e): Eigen faces generation process

The number of eigen faces is equal to number of face images in the training set. However faces can also be approximated using only best eigen faces those have the largest eigen values and therefore account for the most variance within the set of face images [18]. In PCA technique [2]; images can be approximately reconstructed by storing small collection of weights for each of face and small set of standard pictures. Images in training set can be reconstructed by weighted sum of small collection of characteristic images. A flow chart for PCA analysis is shown in Fig. 1(g).

Fig. 1(f): A face developed in the face space

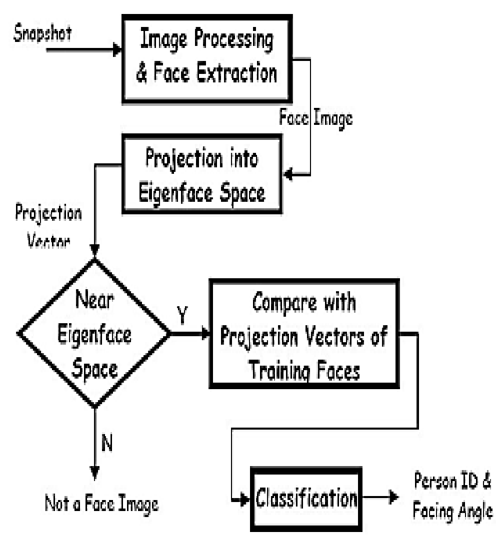

Figure 1(g): Flow chart for PCA

2.2 Line Edge Map (LEM)

The edge images of objects are used for object recognition and to achieve similar accuracy as gray-level images. Taka Acs [22] made use of edge maps to measure the similarity of face images. The faces were converted into binary edge maps by applying Sobel edge detection algorithm. The Hausdorff distance is used to find the similarity of the two point sets. The following modified Hausdorff distance

$$h(A,B) = \frac{1}{N_a}\sum_{a \epsilon A} \min_{b \epsilon B} \|a - b\| \qquad (1)$$

is used, as it is less sensitive to noise than the maximum or kth ranked Hausdorff distance formulations. 92 % accuracy is achieved by the experiments. Taka Acs [22] suggested that the process of face recognition might start at a much earlier stage and edge images can be used for the recognition of faces without the involvement of high-level cognitive functions.

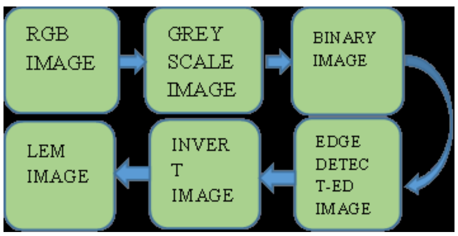

Figure 2: Block diagram of LEM

The Hausdorff distance uses only the spatial information of an edge map without considering the inherent local structural characteristics inside such a map. Bruneli and Poggio [11] suggested that successful object recognition approaches might need to combine aspects of feature-based approaches with template matching method. This is a valuable hint for us when proposing a LEM approach which extracts lines from a face edge map as features. A simple block diagram of LEM is shown in Fig. 2. This approach can be considered as a combination of template matching and geometrical feature matching.

2.2.1 Line Segment Hausdroff Distance (LHD)

LHD uses the additional structural attributes of line orientation, line-point association and hence have a better distinct power than LEM. LHD is used to measure the similarity of face LEMs. The LHD is basically used to compare shape based LEMs which is a distance between two line sets [4].

Given two LEMs $M^l = \{m_1^l, m_2^l, \ldots. m_p^l\}$ and $T^l = \{t_1^l, t_2^l, \ldots.. t_q^l\}$, LHD is built on the vector d(m_i^l, t_j^l) that represents the distance between two line segments m_i^l and t_j^l.The vector is defined as

$$\vec{d}\,(m_i^l, t_j^l) = \begin{bmatrix} d_\theta(m_i^l, t_j^l) \\ d_\parallel(m_i^l, t_j^l) \\ d_\perp(m_i^l, t_j^l) \end{bmatrix} \qquad (2)$$

The representation of above is shown in Fig.3.

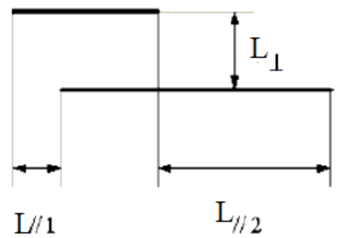

Figure 3: Representation of orientation distance, parallel distance and perpendicular distances

Where $d_\theta(m_i^l, t_j^l)$, $d_\parallel(m_i^l, t_j^l)$ and $d_\perp(m_i^l, t_j^l)$ are the orientation distance, parallel distance and perpendicular distance respectively. All these three entries are independent and defined as

$$d_\theta(m_i^l, t_j^l) = \mathrm{f}(\boldsymbol{\theta}(m_i^l, t_j^l\ ) \qquad (3)$$

$$d_\parallel(m_i^l, t_j^l) = \min(l_{\parallel 1}, l_{\parallel 2}) \qquad (4)$$

$$d_\perp(m_i^l, t_j^l) = l_\perp (5)$$

$\theta(m_i^l, t_j^l)$ computes the smallest intersecting angle between lines m_i^l and t_j^l.f($\boldsymbol{\theta}(m_i^l, t_j^l)$ is a nonlinear penalty function to map an angle to a scalar. It is necessary to ignore small angle variation but penalize heavily on large deviation. The quadratic function $f(x) = x^2/_W$ is used, where W is the weight to be determined by a training process [4]. The designs of the parallel and perpendicular displacements can be illustrated with a simplified example of two parallel lines, m_i^l andt_j^l. $d_\parallel(m_i^l, t_j^l)$ is defined as the minimum displacement to align either the left end points or the right end points of the lines. $d_\perp(m_i^l, t_j^l)$ is simply the vertical distance between the two lines. Normally, m_i^l and t_j^l would not be parallel, but one can rotate the shorter line with its midpoint as rotation center to the desirable orientation before computing $d_\parallel(m_i^l, t_j^l)$ and $d_\perp(m_i^l, t_j^l)$.

The shorter line is selected to rotate because this would cause less distortion to the original line pair. The effect of broken lines caused by segmentation error and alleviate the effect of adding, missing, and shifting of feature points caused by inconsistency of feature point detection, the parallel shifts $l_{\parallel 1} and l_{\parallel 2}$ are reset to zero if one line is within the range of the other. Finally, the distance between two line segments m_i^l and t_j^l is given by

$$d(m_i^l, t_j^l) = \sqrt{d_\theta(m_i^l, t_j^l)^2} + \sqrt{d_\parallel(m_i^l, t_j^l)^2} + \sqrt{d_\perp(m_i^l, t_j^l)^2} \qquad (6)$$

A primary line segment Hausdorff distance pLHD is defined as

$$H_{pLHD}(M^l, T^l) = \max(h(M^l, T^l), h(T^l, M^l)) \qquad (7)$$

Where

$$h(M^l, T^l) = \frac{1}{\sum_{m_i^l \epsilon M^l} l_{m_i^l}} \sum_{m_i^l \epsilon M^l} l_{m_i^l} \cdot \min_{t_j^l \epsilon T^l} d(m_i^l, t_j^l) \qquad (8)$$

and $l_{m_i^l}$ is the length of line segment m_i^l. In the calculation of $h(M^l, T^l)$ is the distance contribution from each line weighted by its length. In the definition of $d_\parallel(m_i^l, t_j^l)$, it can be found that the displacement distance depends on the smaller distance between the left/right end points of the two line segments to be matched, which means the measure only reflects the smallest shift of the two line end points [4].

2.3 Linear Discriminant Analysis (LDA)

The basic idea of LDA is grouping of similar classes of data whereas PCA works directly on data. It seeks to find directions along which the classes are best separated. An example of fisher face is given below in Fig. 4.

Figure 4: Fisher faces

In facts, PCA is used as a basic step for dimensionality reduction and removal of the null spaces of the two scatter matrices. Then LDA is performed in the lower dimensional PCA subspace as it was done for example in Fisher faces. It is concluded that the discarded null spaces can contain significant discriminatory information. To avoid this direct LDA (D-LDA) methods have been introduced. The basic premise behind the D-LDA approaches is that the information residing in the null space of the within-class scatter matrix is more significant for discriminate tasks than the information out of the null space [20].

2.3.1 Mathematical Analysis

Following are the basic steps in LDA:

- Calculate with-in class scatter matrix, S_W:
  $$S_W = \sum_{i=1}^{c} \sum_{x_k \epsilon X_i} (x_k - \mu_i).(x_k - \mu_i)^T \qquad (9)$$

Where x_k is the ith sample of class i, μ_i is the mean of class i, and C is the number of classes.

- Calculate between class scatter matrix, S_B :
  $$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu).(\mu_i - \mu)^T \qquad (10)$$

  Where $\boldsymbol{\mu}$ represents the mean of all classes.

- Calculate the eigenvectors of projection matrix
  $$W = \text{eig}\ (S_W^{-1} . S_B) \qquad (11)$$

Compare the test image's projection matrix with the projection matrix of each training image by using a similarity measure. The result is the training image which is the closest to the test image.

LDA provides the 1D subspace where the Bayes classification error is smallest in the two-class homoscedastic problem [20]. One way to add flexibility to the kernel is to allow for each class to be subdivided into several subclasses.

2.3.2 With-in class & between class scatter matrices in LDA

Given a set of high-dimensional data grouped into classes, LDA aims to find an optimal transformation that tries to maximize the ratio

$$J_{LDA}(W) = arg_W max \frac{|W^T S_B W|}{|W^T S_W W|} \qquad (12)$$

Where S_B the between-class is scatter matrix and S_W is the within-class scatter matrix. Thus, by solving a generalized eigen values problem, the projection vector W can be found as the eigenvectors of $S_W^{-1} S_B$ corresponding to the largest eigenvalues. When the sample size is smaller than the dimensionality of samples, however, S_W becomes singular and we cannot compute $S_W^{-1}S_B$ directly. This is the major drawback of classical LDA for face recognition [20].

2.3.3 Limitations of LDA

LDA is one of the most popular techniques in statistical pattern recognition [23]. However, there are three major drawbacks restricting its use like, the so-called small sample size problem (SSS) [24]-[25], the (common) situation that real-world data have heteroscedastic class distributions, which violates the fundamental homoscedasticity assumption of LDA [25]–[28] and the instability of the LDA criterion in cases when the metric to be minimized and the metric to be maximized are in conflict [30]. Subclass discriminant analysis (SDA) [29] overcomes the above limitations.

Improvements in LDA can be roughly grouped into three categories: namely; the first category focuses on addressing the small sample size (SSS) problem, which always occurs when the data dimension exceeds the number of training samples. The second category of improvement focuses on the incremental versions of the LDA which are very suitable for online learning tasks. One of their main advantages is that the algorithm does not need to store the whole data matrix in the main memory. The third category of improvement is the tensor-based LDA which has shown fairly large improvement [31].

2.4 Kernel PCA (KPCA)

Kernel design is based on various methods. KPCA is performed using following steps [21]:

a) Kernel methods offer a modular framework.
b) In a first step, a dataset is processed into a kernel matrix. Data can be of various types, and also heterogeneous types as shown in Fig.6
c) In a second step, a variety of kernel algorithms can be used to analyze the data, using only the information contained in the kernel matrix as shown in Fig.5.

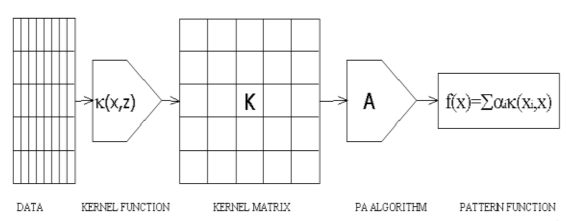

Figure 5: Steps of kernel PCA

2.4.1 Mathematical analysis

Given a set of m cantered (zero mean, unit variance) samples x_k

Where x_k=[$x_{k1}, x_{k2} \ldots . x_{kn}$]$^T R^n$, PCA aims to find the projection directions that maximizes the variance, C, which is equivalent to finding the eigen values from the covariance matrix

$$\lambda_W = C_W \quad (13)$$

For eigen values $\boldsymbol{\lambda} \geq 0$ and eigen vectors w $\boldsymbol{\epsilon} R^n$ [21]. In kernel PCA, each vector x is projected from the input space, R^n, to a high dimensional feature space, R^f, by a nonlinear mapping function:

$$R^n \longrightarrow R^f, f >> n \quad (14)$$

Note that the dimensionality of the feature space can be arbitrarily large [21]. In R^f, the corresponding eigen value problem is

$$\boldsymbol{\lambda} w^\phi = C^\phi w^\phi \quad (15)$$

Where $C^\phi$is a covariance matrix. All solutions w^ϕ with $\boldsymbol{\lambda} \neq 0$ lie in the span of (x_1),........, (x_m), and there exist coefficient α_i such that

$$w^\phi = \sum_{i=1}^{m} \alpha_i . \phi(x_i) \quad (16)$$

Denoting an m× m matrix K by

$$K_{ij} = k(x_i, x_j) = (x_i).(x_j) \quad (17)$$

the kernel PCA problem becomes

$$m\, \boldsymbol{\lambda} K \boldsymbol{\alpha} = K^2 . \boldsymbol{\alpha} \quad (18)$$

$$m\, \boldsymbol{\lambda} \alpha = K \boldsymbol{\alpha} \quad (19)$$

where $\boldsymbol{\alpha}$ denotes a column vector with entries α_1, α_2,.......,α_m. The above derivations assume that all the projected samples $\boldsymbol{\phi}$(x) are centered in R^f. We can now project the vectors in R^f to a lower dimensional space spanned by the eigenvectors w^ϕ.

Let x be a test sample whose projection is $\boldsymbol{\phi}$(x) in R^f, then the projection of $\boldsymbol{\phi}$(x) onto the eigenvectors w^ϕ is the nonlinear principals corresponding to $\boldsymbol{\phi}$:

$$w^\phi . \phi(x) = \sum_{i=1}^{m} \alpha_i (\phi(x_i) . \phi(x)) = \sum_{i=1}^{m} \alpha_i\, k(x_i, x) \quad (20)$$

We can extract the first q($1 \leq q \leq m$) non-linear principal components using the kernel function without the expensive operation that explicitly projects the samples to a high dimensional space R^f. The first q components correspond to the first q non-increasing eigen values [21]. For face recognition where each x encodes a face image, we call the extracted non-linear principal components kernel eigen values. PCA is a basis transformation to diagnose an estimate of the covariance matrix of the data

x_k, k = 1, 2, 3... $x_k \epsilon R^N$, (21)

$\sum_{k=1}^{l} x_k = 0$, defined as

$$C = \frac{1}{l}\sum_{j=1}^{l} x_j . x_j^T \quad (22)$$

The new coordinates in the eigenvector basis i.e, the orthogonal projection on to the eigen vectors are called principal components. To generalize this setting to a nonlinear one first map the data nonlinearly in to a feature space F by

$$R^N \rightarrow F, \qquad x \rightarrow X \quad (23)$$

2.4.2 Kernel PCA

Assume for the moment that our data mapped into feature space, (x_1),........(x_l), is centered

i.e. $\sum_{k=1}^{l} \phi(x_k) = 0$. (24)

To do PCA for the covariance matrix

$$\bar{C} = \frac{1}{l}\sum_{j=1}^{l} \phi(x_j).\phi(x_j)^T \quad (25)$$

Find eigen values $\boldsymbol{\lambda} \geq 0$ and eigen vectors V $\boldsymbol{\epsilon}$ F {0}, satisfying

$$\boldsymbol{\lambda} V = \bar{C} V. \quad (26)$$

Substituting the above equation, we note that all solutions V lies in the span of (x_1), (x_2).....(x_l). This implies that we may consider the equivalent system

$$((x_k).V) = (\boldsymbol{\phi}(x_k).\bar{C}V) \quad (27)$$

for all k = 1,2,.....$\boldsymbol{\ell}$.

and that there exist coefficients $\alpha_1, \alpha_2, \dots \dots \alpha_l$ such that

$$V = \sum_{i=1}^{l} \alpha_i . \boldsymbol{\phi}(\boldsymbol{x_i}) \quad (28)$$

substituting (26) and (27) into (28), and defining an $l \times l$ matrix K by

$$K_{ij} := (\boldsymbol{\phi}(x_i) . (x_j)) \quad (29)$$

we arrive at

$$\boldsymbol{\ell\lambda} K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha} \quad (30)$$

Where $\boldsymbol{\alpha}$ denotes the column vectors with entries $\alpha_1, \alpha_2 \dots \dots \alpha_l$.To find solutions of above equations, we solve the eigen value problem

$$\boldsymbol{\ell\lambda\alpha} = K\boldsymbol{\alpha} \quad (31)$$

for nonzero eigenvalues. Clearly, all solutions of above equations do satisfy equation (29). Moreover, it can be shown that any additional solutions of above equations do not make a difference in the Eqn. (30). We normalize the solutions α^kbelonging to nonzero eigen values by requiring that the corresponding vectors in F be normalized i.e; (V^k . V^k) = 1. By virtue of above equations this translates into

$$1 = \sum_{i,j=1}^{l} \alpha_i^k . \alpha_j^k \, (\phi(x_i).\phi(x_j)) = (\alpha^k.K.\alpha^k) = \lambda_k(\alpha^k.\alpha^k). \quad (32)$$

For principal component extraction, we compute projections of the image of a test point $\boldsymbol{\phi}$(x) onto the eigenvectors V^k in F according to

$$V^k.\phi(x) = \sum_{i=1}^{l} \alpha_i^k \, (\boldsymbol{\phi}(x_i) . \boldsymbol{\phi}(x)) \quad (33)$$

Note that neither (32) nor (33) requires the (x_i) in explicit form - they are only needed in dot products. Therefore, we are able to use kernel functions for computing these dot products without actually performing the map [21]. For some choices of a kernel k(x; y), it can be shown by methods of functional analysis that there exists a map into some dot product space F such that k computes the dot product in F.

$$k(x,y) = (x . y)^d \quad (34)$$

radial basis function

$$k(x,y) = \exp(-\|(x-y)\|^2/2\sigma^2)) \quad (35)$$

and sigmoid kernels

$$k(x,y) = \tanh(k(x,y) + \boldsymbol{\theta}) \quad (36)$$

It can be shown that polynomial kernels of degree d correspond to a map $\boldsymbol{\phi}$ into a feature space which is spanned by all products of d entries of an input pattern e.g; for the case of N = 2, d = 2,

$$((x.y)^2) = (x_1^2, x_1x_2, x_2x_1, x_2^2)(y_1^2, y_1y_2, y_2y_1, y_2^2)^T \quad (37)$$

2.5 Comparison of PCA, KPCA and PCA & LDA

PCA and KPCA are important feature extraction techniques used for dimensionality reduction. It is very important to reduce the number of dimensions which increase the computation time heavily. PCA linearly converts the original inputs into new uncorrelated features. KPCA is a nonlinear PCA developed by using the kernel method.

3. Results

Here, four basic face recognition techniques are compared for different algorithms based on recognition rates (percentage) and elapsed time. The common training data set and test data set is used for comparison in all four used techniques. In face recognition part, we used training set of faces in which there were 20 faces in the training set having two expression of one face as shown in Fig. 7. There was a test database in which there were 10 faces as shown in Fig. 6. We took one test image from them as input image and apply face recognition algorithm using PCA on it. The result is equivalent face from the train database shown in Fig. 8. The same procedure is applicable to rest of three algorithms i. e; KPCA, LDA, and LEM for face recognition applying on the same database as shown in Fig. 6 and Fig. 7.

In this work, basically the comparison of various algorithms as stated above i. e; PCA, KPCA, LDA and LEM is done on the basis of recognition time used in face recognition procedure. All the algorithms were applied in MATLAB environment to compare recognition times and recognition rates as shown in Table-1.

Recognition Time: Elapsed Time is 24.13 seconds
Recognition Time: Elapsed Time is 20.33 seconds
Recognition Time: Elapsed Time is 33.80 seconds
Recognition Time: Elapsed Time is 5.07 seconds

From above discussions, it is clear that LEM has the lowest recognition time while LDA has the highest recognition time in seconds. Hence, from this we conclude that best algorithm applied for face recognition is LEM technique.

Table-1: Comparison of various methods on the basis of recognition rate and recognition time.

S. No.	Methods	Recognition Time (sec.)	Recognition Rate (%)
1.	PCA	24.13	98.99
2.	KPCA	20.33	94.99
3.	LDA	33.80	97.99
4.	LEM	5.07	96.00

Figure 6: Test Database

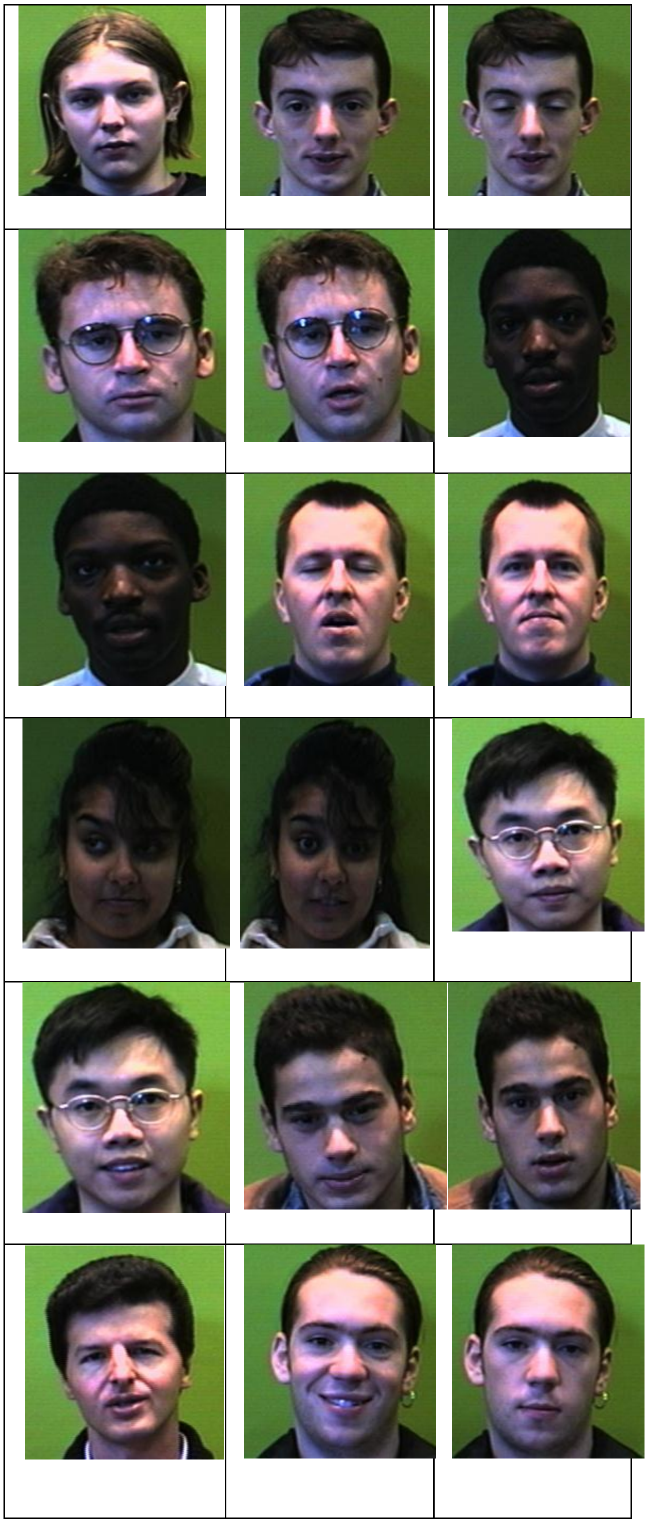

Figure 7: Training Database

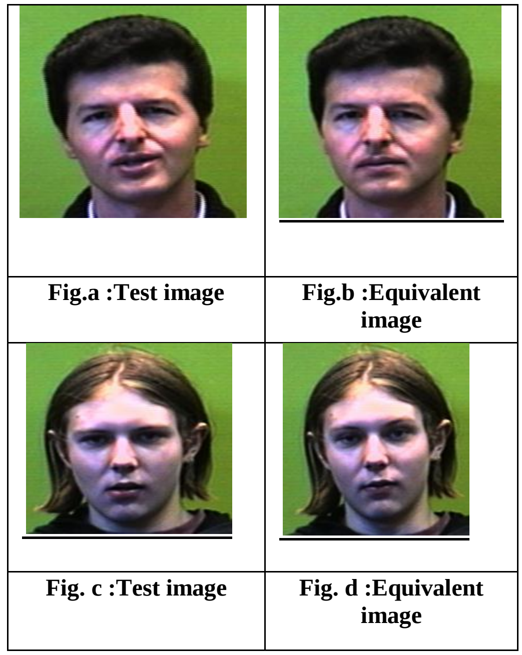

Fig.a :Test image **Fig.b :Equivalent image**

Fig. c :Test image **Fig. d :Equivalent image**

Figure 8: Resultant images

4. Conclusions

This work is showing a scenario of our future gadgets, where images can be recognized and matched for security purposes. We have developed a real-time face recognition system by combining face tracking and face technologies together. The system has provided a platform for developing new face recognition algorithms. During this study, some limitations are found only on theoretical basis. This approach was tested on a number of face images as shown in training images.

From the above proposed techniques, it is concluded that LEM is much better as compared to others techniques in error occurring conditions like lighting conditions, pose variations, facial expression. Our proposed LEM is exact in every conditions and achieving 96% accuracy which is much higher than other techniques. Also this is less time consuming, less memory occupying and faster technique. So overall all aspects it gives satisfactory results and due to its less memory occupation, it can be used in embedded system like mobile banking system.

References

[1] J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of human faces", Proceeding IEEE, May 1971 vol. 59, no. 5, pp. 748-760.

[2] L. Sirovich and M Kirby, "A low dimensional procedure for the characterization of human faces", JOSA A 4, no. 3 (1987): 519-524.

[3] M. A .Turk and A. P. Pentland, "Face recognition using eigen faces," Proc. IEEE, 1991, pp. 586-591.

[4] Sakai, T., Nagao, M., and Fujibayashi, S. "Line extraction and pattern recognition in a photograph", Pattern Recognition, vol.1, pp. 233–248, 1969.

[5] Kelly, M, "Visual identification of people by computer" Stanford AI Proj., Stanford, CA, Tech. Rep, 1970.

[6] Govindaraju, V., Srihari, S. and Sher, D. "A Computational model for face location", in Proc. 3rd Int. Conf. on Computer Vision 1990, pp. 718–721.

[7] Yuille, A., Cohen, D. and Hallinan, P., "Feature extraction from faces using deformable templates", in Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition, 1989, pp. 104–109.

[8] Burt P., "Multi resolution techniques for image representation, analysis, and 'smart' transmission", in SPIE Proc. 1199 Visual Communication and Image Proc. vol. 4, pp. 2-17.

[9] Craw, I., Tock, D. and Bennett, A. "Finding face features", in Proc. 2nd Europe. Conf. on Computer Vision 1992, pp. 92–96.

[10] Harmon, L. D. and Hunt, W. F. , "Automatic recognition of human face profiles", Computer Graphics and Image Proc., vol. 6 pp. 135– 156, 1977.

[11] Brunelli, R. and Poggio, T. "Face recognition: Features versus templates", IEEE Trans. on Patt. Anal. and Mach. Intel., vol. 15, no.10, pp. 1042–1052, 1993.

[12] Turk, M. and Pentland, A. "Face recognition using eigen faces", in Proc. Int. Conf. on Comp. Vision and Patt.Recog 1991 (CVPR'91), pp. 586–591.

[13] Akamatsu, S. , Sasaki, T. , Fukamachi, H. and Suenaga, Y, "A robust face identification scheme—KL expansion of an invariant feature space", in SPIE Proc., Intell. Robots and Computer Vision X Algorithms and Techn., 1991 vol. 1607, pp. 71–84.

[14] Swets, D. and Weng J., "Using discriminant eigen features for image retrieval", IEEE Trans. on Pattern. Anal. and Mach. Intell., vol. 18, no.8, pp. 831–836, 1996.

[15] Belhumeur, P. N., Hespanha, J. P. and Kriegman, D. J. "Eigen faces vs. fisher faces: Recognition using classspecific linear projection", IEEE Trans. on Patt. Anal. and Mach. Intel., vol. 19, no.7, pp. 711– 720, 1977.

[16] Duda, R. O. and Hart, P. E., "Pattern classification and scene analysis", Wiley: New York, 1973.

[17] Sudhanshu Tripathi, Leena Singh and Himakshi Arora, "Face recognition machine vision system using eigen faces", International Journal of Recent Trends in Engineering, vol. 2, pp. 2, 2009.

[18] S. Ravi, Sadique Nayeem, "A study on face recognition technique based on eigenface", International Journal of Applied Information Systems, vol. 5, no. 4, 2013.

[19] Z. Chai, Z. Sun, H. Mendez-Vazquez, R. He and T. Tan, "Gabor Ordinal, measures for face recognition", IEEE Trans. on Information Forensic and Security, vol. 9, no. 1, pp. 1-26, 2014.

[20] N. Gkalelis, V. Mezaris and t. Stathaki, "Mixture subclass discriminant analysis link to restricted Gausian model and other generalizations", IEEE Trans. on Neural Network and Learning Systems, vol. 24, no. 1, pp. 8-21, 2013.

[21] W. Zheng, Z. Lin and H. Wang, "L1-norm kernel discriminant analysis via Bayes error bound optimization for robust feature extraction", IEEE Trans. on Neural Network and Learning Systems, pp. 1-13, 2013.

[22] B. TakaÂcs, "Comparing face images using the modified hausdorff distance", Pattern Recognition, vol. 31, pp. 1873-1881, 1998.

[23] K. Fukunaga, "Introduction to statistical pattern recognition", 2nd edition,. San Diego, CA: Academic, 1990.

[24] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigen faces vs. fisher faces: recognition using class specific linear projection," IEEE Trans. Pattern Anal. and Mach. Intell., vol. 19, no. 7, pp. 711–720, Jul. 1997.

[25] J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition," Pattern Recognition. Letter, vol. 26, no. 2, pp. 181–191, Jan. 2005.

[26] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," J. Royal Statist. Soc. B, vol. 58, no. 1, pp. 155–176, Jul. 1996.

[27] A. K. Jain and S. Z. Li, "Handbook of Face Recognition", New York, NY, USA: Springer-Verlag, 2005.

[28] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in Proc. ECCV, 2004, pp. 469–481.

[29] M. Zhu and A. Martinez, "Subclass discriminant analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[30] A. M. Martinez and M. Zhu, "Where are linear feature extraction methods applicable?," IEEE Trans. Pattern Anal. and Mach. Intell., vol. 27, no. 12, pp. 1934–1944, Dec. 2005.
[31] Zizhu Fan, Yong Xu and David Zhang, "Local linear discriminant analysis framework using sample neighbours" IEEE Transactions on Neural Networks, vol. 22, no. 7, July 2011.

Abhimanyu Madan was born on 14th April 1990. He received his B. Tech. degree in Electronics and Communication Engineering from M.D.U., Rohtak in 2011 and pursuing M. Tech. in Electronics and Communication Engineering from DCRUST. Currently, He is working as Tech. Sale Executive in PC Yantra Computer Solutions, Pvt. Ltd., Saket New Delhi.
Email: sangeeta.kaushik1@gmail.com.

Rash Bihari Dubey was born in India on 10th November 1961. He received the M. Sc. degree in Physics with specialization in Electronics in 1984 from Agra University Agra, India, the M. Tech. degree in Instrumentation from R.E.C. Kurukshetra, India in 1989 and the Ph.D. degree in Electronics Enggg., from M. D. University, Rohtak, India in 2011. He is at present Professor and Head in the Department of Electronics and Communication Engineering at Hindu College of Engineering, Sonepat, India. He has well over 52 publications in both conferences and journals to his credit His research interest are in the areas of Medical Imaging, Digital Signal Processing, Digital Image Processing, Biomedical Signal Analysis, and Industrial Real Time Applications.