Performance Metrics for Eigen and Fisher Feature Based Face Recognition Algorithms

Harihara Santosh Dadi, Krishna Mohan P.G.

Abstract
Three performance curves for evaluation of face recognition algorithms are introduced in this paper. Cumulative Match Score Curves (CMC) is the curve between the rank and face recognition rate. Expected Performance Curves (EPC) is the graph between the alpha and error rate. Receiver Operating Characteristics (ROC) is the graph between false acceptance rate and verification rate. Twelve face recognition algorithms based on Eigen and fisher features are compared based on these curves. The performances of all these algorithms are analyzed based on these metrics.

Index Terms
Facial features, Principle Component Analysis, Kernel Fisher Analysis, Kernel Principle Component Analysis, Linear Discriminant Analysis, CMC, EPC, ROC.

1. Introduction

Face recognition aims at identifying the person's distinctiveness by comparing the facial features with the available face data base features. The face data base, with known characteristics, is referred as the face gallery and the input face requiring determining the identity is the probe. One of the problems in face recognition is identification, and the other is the authentication (or verification). Of the two, face identification is more tricky as it cross verifies the gallery completely for minimum variance.

Face recognition has been an important topic of research originated way back in the year 1961. Numerous algorithms are developed on face recognition particularly in the last two to three decades. Improving the Face recognition rate is always the challenge ever since the first algorithm was developed. In 1991, Alex Pentland and Matthew Turk [1], [3] – [5] applied Principal Component Analysis (PCA) which was invented in 1901 to face classification. This has become the standard known as the eigenface method and is today an inspiration for all face recognition algorithms evolved. Sebastain Mike et. al. [2] competing with PCA which is an orthogonal linear transformation proposed Fisher Discriminant Analysis with Kernels (KFD). Being a non linear classification model, KFD has better performance over PCA. The problem with KFD is that it uses all the training samples in the solution not only the difficult ones which makes the algorithm slower and also complex.

We compare algorithms namely PCA [6] – [10], LDA [11] – [12], KPCA [13] – [16], KFA [17] – [18], G-PCA, G-KPCA, G-LDA, G-KFA [21] – [26], PC-PCA, PC-KPCA, PC-LDA and PC-KFA [26].

While numerous face recognition algorithms are being developed, the authors are comparing them with the existing ones very superficially and few simple comparisons are presented. Given that large set of techniques and the theories that are applicable for face recognition, it is evident that the detailed analysis and bench marking these algorithms is very crucial. Effort done by Universities and research laboratories in developing the data sets pushed the comparisons of face recognition algorithms to the higher level. CMC, ROC and EPC curves [19] – [20] were introduced for comparisons. Apart from finding the recognition rate, these curves become the basis for showing the superiority of the author's developed algorithms.

The contributions of this paper are as follows:

- Twelve face recognition algorithms are compared using performance metrics.
- Extensive comparisons are made by taking the performance metrics curves namely CMC, EPC and ROC and showed that the curves are effective for proposed algorithm [20].

2. Related Work

Face recognition methods mainly deal with images which are of large dimensions. This makes the task of recognition very difficult. Dimensionality reduction is a concept which is introduced for the purpose of reducing the image dimensions. PCA is the most widely used dimensionality reduction and also for subspace projection. PCA can supply the client with a lower-dimensional picture, a projection of this object when seen from its informative view point. This can be achieved by taking only the starting few principal components in such a way that the dimension of the transformed data is minimized. The linear combination of pixel values here in PCA are called Eigen faces. PCA is an unsupervised and it ignores all the class labels. It treats the entire data as a whole. It uses SVD for dimensionality reduction. PCA is however not optimized for class

Manuscript received June 5, 2016
Manuscript revised June 20, 2016

separability. An alternative is proposed by Ronald Fisher which is Linear Discriminant Analysis (LDA). This cares for class seperability. Being a supervised, it performs well when the dataset contains more number of face images. PCA, however performs well if the data set is very small. PCA is primarily used for feature extraction whereas LDA is used for classification. For non-linear structures, kernel based algorithms are developed. By using the PCA or LDA for high dimensional mapping, the computational time is greatly increased. To use the PCA and LDA for higher dimensions, kernel based algorithms are developed. The kernel based PCA is called Kernel Principal Component Analysis (KPCA) and the kernel based LDA is called Kernel Fisher Analysis (KFA). These kernel functions enable the algorithms to operate at higher dimensions without computing the data coordinates in the higher space. Rather it simply computes the inner products between the images of all pairs of data in the feature space. These kernel based algorithms are computationally cheaper than the explicit computation of the coordinates which is done in PCA and LDA.

Among all these kernel trick based methods, Kernel Fisher Analysis is computationally simple. It needs only the factorization of gram matrix calculated with the given training examples. The other kernel based methods namely KPCA solves convex optimization problems. The beauty of the KFA is that it is comparable to the famous SVD classifier. These four methods are considered for conducting experiments. Apart from these four, their Gabor and phase congruency based methods are also included. Gabor filter is spatially and frequency localized. Because of this it can achieve desired frequency resolution. For normal face images, Gabor representation is sparser than the pixel representation.

In phase congruency based method, first the phase congruency features are calculated. Instead of probing for dots of high intensity gradients, the model finds out those points in the face image Region Of Interest (ROI) where the 2-D log Gabor filter output over a number of orientations and scales are maximum in phase. Therefore, a point in the image is of high value only if the phase responses of the log-Gabor filters over a range of orientations and scales display different kind of order. Phase congruency acts as an edge (or line) descriptor of an image and is, unlike gradient based edge-detectors, it is not susceptible to the image variations affected by blurring, magnification, illumination and alike [38] – [39]. It is robust to image variations and it stems from the multi-orientation and multi-scale approach to phase congruency calculation and also from the fact that phase rather than magnitude information is considered for edge (or line) detection.

Three performance metrics curves are considered. Cumulative Match Score Curves (CMC) is the curve between the rank on the x-axis and face recognition rate on the y-axis. Expected Performance Curves (EPC) is the graph between the alpha and error rate. Receiver Operating Characteristics (ROC) is the graph between false acceptance rate and verification rate. ROC curves are more informative and EPC curves are hard to compute and read. EPC curves need a separate development set. But this problem of EPC curves can be taken care by relying on the cross-validation techniques.

In this work we take these four namely PCA, KPCA, LDA and KFA and their extension algorithms (Gabor based [21] –[25] and Phase Congruency [26] based face recognition algorithms) and show how by using this new recognition engine for these algorithms, the performance of these algorithms is optimized.

3. Face recognition algorithm

A typical face recognition algorithm is presented in this section. For any face recognition algorithm, there are two phases. One is training phase and the other is the testing phase. In the training phase, the features of all the faces in the gallery are found and stored in the data base. Eigen features are taken in the sample face recognition algorithm shown below in the figure 1. In the testing phase, the features of the probe are calculated. These features and the features of the gallery are given to any of the classifier. SVD classifier is taken as example in the figure. The Eigen features of the probe and the Gallery are taken by the SVD. The classifier looks for the closest feature matching face from the gallery with the probe and gives that face as output. Figure1 shows the sample face recognition algorithm block diagram.

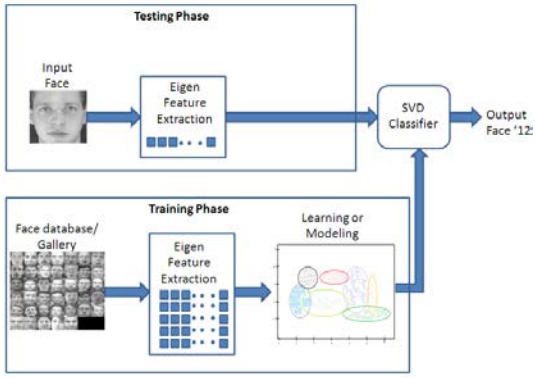

Fig.1 existing face recognition system

4. Principal Component Analysis

PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis

of the ellipse is small, then the variance along that axis is also small, and by omitting that axis and its corresponding principal component from our representation of the dataset, we lose only a commensurately small amount of information.

To find the axes of the ellipse, we must first subtract the mean of each variable from the dataset to center the data around the origin. Then, we compute the covariance matrix of the data, and calculate the Eigen values and corresponding eigenvectors of this covariance matrix. Then, we must orthogonalize the set of eigenvectors, and normalize each to become unit vectors. Once this is done, each of the mutually orthogonal, unit eigenvectors can be interpreted as an axis of the ellipsoid fitted to the data. The proportion of the variance that each eigenvector represents can be calculated by dividing the Eigen value corresponding to that eigenvector by the sum of all Eigen values.

It is important to note that this procedure is sensitive to the scaling of the data, and that there is no consensus as to how to best scale the data to obtain optimal results.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Consider a data matrix, X, with column-wise zero empirical mean (the sample mean of each column has been shifted to zero), where each of the n rows represents a different repetition of the experiment, and each of the p columns gives a particular kind of feature (say, the results from a particular sensor).

Mathematically, the transformation is defined by a set of p-dimensional vectors of weights or loadings $\mathbf{w}_k = (w_1,.....w_p)_{(k)}$ that map each row vector $\mathbf{x}_{(i)}$ of X to a new vector of principal component scores $\mathbf{t}_k = (w_1,.....w_k)_{(i)}$, given by

$$\mathbf{t}_{k(i)} = \mathbf{x}_{(i)}.\mathbf{w}_{(k)} \quad (1)$$

in such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from x, with each loading vector w constrained to be a unit vector.

A. First component

The first loading vector $\mathbf{w}_{(1)}$ thus has to satisfy

$$\mathbf{w}_{(1)} = \arg\max\left\{\sum_i (t_1)^2_{(i)}\right\} = \arg\max\left\{\sum_i (\mathbf{x}_{(i)}.\mathbf{w})^2\right\}$$

Equivalently, writing this in matrix form gives

$$\mathbf{w}_{(1)} = \arg\max\left\{\sum_i \|\mathbf{X}\mathbf{w}\|^2\right\} = \arg\max\left\{\sum_i \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right\}$$

(2)

Since $\mathbf{w}_{(1)}$ has been defined to be a unit vector, it equivalently also satisfies

$$\mathbf{w}_{(1)} = \arg\max\left\{\frac{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\right\} \quad (3)$$

The quantity to be maximized can be recognized as a Rayleigh quotient. A standard result for a symmetric matrix such as X^TX is that the quotient's maximum possible value is the largest Eigen value of the matrix, which occurs when w is the corresponding eigenvector.

With $w_{(1)}$ found, the first component of a data vector $x_{(i)}$ can then be given as a score $t_{1(i)} = x_{(i)} \cdot w_{(1)}$ in the transformed co-ordinates, or as the corresponding vector in the original variables, $\{x_{(i)} \cdot w_{(1)}\}\ w_{(1)}$.

Further components

The kth component can be found by subtracting the first k − 1 principal component from X:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}^T{}_{(s)} \quad (5)$$

and then finding the loading vector which extracts the maximum variance from this new data matrix

$$\mathbf{w}_{(k)} = \arg\max\left\{\left\|\hat{\mathbf{X}}_k\mathbf{w}\right\|^2\right\} = \arg\max\left\{\frac{\mathbf{w}^T\hat{\mathbf{X}}_k^T\hat{\mathbf{X}}_k\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\right\}$$

(6)

It turns out that this gives the remaining eigenvectors of X^TX, with the maximum values for the quantity in brackets given by their corresponding Eigen values. Thus the loading vectors are eigenvectors of X^TX.

The kth component of a data vector $x_{(i)}$ can therefore be given as a score $t_{k(i)} = x_{(i)} \cdot w_{(k)}$ in the transformed co-ordinates, or as the corresponding vector in the space of the original variables, $\{x_{(i)} \cdot w_{(k)}\}\ w_{(k)}$, where $w_{(k)}$ is the kth eigenvector of X^TX.

The full principal components decomposition of X can therefore be given as

$$\mathbf{T}=\mathbf{X}\mathbf{W} \quad (7)$$

where W is a p-by-p matrix whose columns are the eigenvectors of X^TX

B. Covariance

X^TX itself can be recognized as proportional to the empirical sample covariance matrix of the dataset X.

The sample covariance Q between two of the different principal components over the dataset is given by:

$$\begin{aligned}&\mathbf{Q}(\mathbf{PC}_{(J)},\mathbf{PC}_{(K)})\alpha(\mathbf{Xw}_{(j)})^T.(\mathbf{Xw}_{(k)})\\&=\mathbf{w}_{(j)}^T\mathbf{X}^T\mathbf{Xw}_{(k)}\\&=\mathbf{w}_{(j)}^T\lambda_k\mathbf{w}_{(k)}\\&=\lambda_k\mathbf{w}_{(j)}^T\mathbf{w}_{(k)}\end{aligned}$$

(8)

where the eigen value property of $w_{(k)}$ has been used to move from line 2 to line 3. However eigenvectors $w_{(j)}$ and $w_{(k)}$ corresponding to eigen values of a symmetric matrix are orthogonal (if the eigen values are different), or can be orthogonalised (if the vectors happen to share an equal repeated value). The product in the final line is therefore zero; there is no sample covariance between different principal components over the dataset.

Another way to characterize the principal components transformation is therefore as the transformation to coordinates which diagonalise the empirical sample covariance matrix.

In matrix form, the empirical covariance matrix for the original variables can be written

$$\mathbf{Q}\alpha\mathbf{X}^T\mathbf{X}=\mathbf{W}\Lambda\mathbf{W}^T \tag{9}$$

The empirical covariance matrix between the principal components becomes

$$\mathbf{W}^T\mathbf{QW}\alpha\mathbf{W}^T\mathbf{W}\Lambda\mathbf{W}^T\mathbf{W}=\Lambda \tag{10}$$

where Λ is the diagonal matrix of eigenvalues $\lambda_{(k)}$ of X^TX

($\lambda_{(k)}$ being equal to the sum of the squares over the dataset associated with each component k: $\lambda_{(k)} = \Sigma_i\ t_k{}^2{}_{(i)} = \Sigma_i\ (x_{(i)} \cdot w_{(k)})^2$)

C. Dimensionality reduction

The transformation $T = X W$ maps a data vector $x_{(i)}$ from an original space of p variables to a new space of p variables which are uncorrelated over the dataset. However, not all the principal components need to be kept. Keeping only the first L principal components, produced by using only the first L loading vectors, gives the truncated transformation

$$\mathbf{T}_L=\mathbf{XW}_L \tag{11}$$

where the matrix T_L now has n rows but only L columns.

In other words, PCA learns a linear transformation

$$t=W^Tx, x\in R^p, t\in R^L \tag{12}$$

Where the columns of p × L matrix W form an orthogonal basis for the L features (the components of representation t) that are decorrelated. By construction, of all the transformed data matrices with only L columns, this score matrix maximizes the variance in the original data that has been preserved, while minimizing the total squared reconstruction error

$$\left\|\mathbf{TW}^T-\mathbf{T}_L\mathbf{W}_L^T\right\|_2^2 \tag{13}$$

or

$$\left\|\mathbf{X}-\mathbf{X}_L\right\|_2^2$$

Such dimensionality reduction can be a very useful step for visualizing and processing high-dimensional datasets, while still retaining as much of the variance in the dataset as possible. For example, selecting L = 2 and keeping only the first two principal components finds the two-dimensional plane through the high-dimensional dataset in which the data is most spread out, so if the data contains clusters these too may be most spread out, and therefore most visible to be plotted out in a two-dimensional diagram; whereas if two directions through the data (or two of the original variables) are chosen at random, the clusters may be much less spread apart from each other, and may in fact be much more likely to substantially overlay each other, making them indistinguishable.

Similarly, in regression analysis, the larger the number of explanatory variables allowed, the greater is the chance of over fitting the model, producing conclusions that fail to generalize to other datasets. One approach, especially when there are strong correlations between different possible explanatory variables, is to reduce them to a few principal components and then run the regression against them, a method called principal component regression.

Dimensionality reduction may also be appropriate when the variables in a dataset are noisy. If each column of the dataset contains independent identically distributed Gaussian noise, then the columns of T will also contain similarly identically distributed Gaussian noise (such a distribution is invariant under the effects of the matrix W, which can be thought of as a high-dimensional rotation of the co-ordinate axes). However, with more of the total variance concentrated in the first few principal components compared to the same noise variance, the proportionate effect of the noise is less—the first few components achieve a higher signal-to-noise ratio. PCA thus can have the effect of concentrating much of the signal into the first few principal components, which can usefully be captured by dimensionality reduction; while the later principal components may be dominated by noise, and so disposed of without great loss.

D. Singular value decomposition

The principal components transformation can also be associated with another matrix factorization, the singular value decomposition (SVD) of X,

$$\mathbf{X} = \mathbf{U}\sum\mathbf{W}^T \quad (14)$$

Here Σ is an n-by-p rectangular diagonal matrix of positive numbers $\sigma_{(k)}$, called the singular values of X; U is an n-by-n matrix, the columns of which are orthogonal unit vectors of length n called the left singular vectors of X; and W is a p-by-p whose columns are orthogonal unit vectors of length p and called the right singular vectors of X.

In terms of this factorization, the matrix X^TX can be written

$$\mathbf{X}^T\mathbf{X} = \mathbf{W}\sum\mathbf{U}^T\mathbf{U}\sum\mathbf{W}^T = \mathbf{W}\sum^2\mathbf{W}^T \quad (15)$$

Comparison with the eigenvector factorization of X^TX establishes that the right singular vectors W of X are equivalent to the eigenvectors of X^TX, while the singular values $\sigma_{(k)}$ ofX are equal to the square roots of the eigenvalues $\lambda_{(k)}$ of X^TX.

Using the singular value decomposition the score matrix T can be written

$$\mathbf{T} = \mathbf{XW} = \mathbf{U}\sum\mathbf{W}^T\mathbf{W} = \mathbf{U}\sum \quad (16)$$

so each column of T is given by one of the left singular vectors of X multiplied by the corresponding singular value. This form is also the polar decomposition of T.

Efficient algorithms exist to calculate the SVD of X without having to form the matrix X^TX, so computing the SVD is now the standard way to calculate a principal L components analysis from a data matrix, unless only a handful of components are required.

As with the eigen-decomposition, a truncated n × L score matrix T_L can be obtained by considering only the first L largest singular values and their singular vectors:

$$\mathbf{T}_L = \mathbf{U}_L\sum_L = \mathbf{XW}_L \quad (17)$$

The truncation of a matrix M or T using a truncated singular value decomposition in this way produces a truncated matrix that is the nearest possible matrix of rank L to the original matrix, in the sense of the difference between the two having the smallest possible Frobenius norm, a result known as the Eckart–Young theorem.

5. Kernel Principal Component Analysis

To perform kernel based PCA, the following steps have to be carried out. First we compute the dot product matrix

$$K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij} \quad (18)$$

Next we solve

$M\lambda\alpha=K\alpha$ (19)

By diagonalizing K, and normalize the Eigen vector expansion coefficients α^n

By requiring

$$1=\lambda_n(\alpha^n.\alpha^n) \quad (20)$$

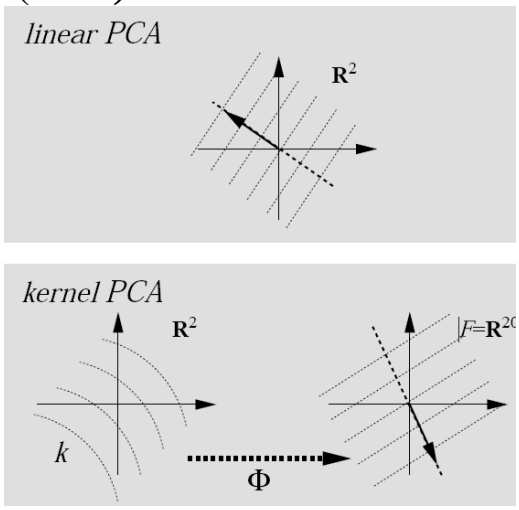

Fig. 2. Linear PCA and Kernel PCA graphical view

To extract the principal components corresponding to the kernel k of a test point **x,** we then compute projections onto the Eigen vectors by

$$(KPCA)_n(\mathbf{x}) = (\mathbf{V}^n.\phi(\mathbf{x})) = \sum_{i=1}^{M}\alpha_i^n k(\mathbf{x}^i, \mathbf{x}) \quad (21)$$

We know that this procedure exactly corresponds to standard PCA in some high-dimensional feature space except that we do not need to perform expensive computation in that space.

6. Linear Discriminant Analysis

Intuitively, the idea of LDA is to find a projection where class separation is maximized. Given two sets of labeled data, $\mathbf{C_1}$ and $\mathbf{C_2}$, define the class means $\mathbf{m_1}$ and $\mathbf{m_2}$ to be

$$m_i = \frac{1}{l_i}\sum_{n=1}^{l_i}\mathbf{x}_n^i \quad (22)$$

where l_i is the number of examples of class $\mathbf{C_i}$. The goal of linear discriminant analysis is to give a large separation of the class means while also keeping the in-class variance small. This is formulated as maximizing

$$\mathbf{J}(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \quad (23)$$

where $\mathbf{S_B}$ is the between-class covariance matrix and $\mathbf{S_W}$ is the total within-class covariance matrix:

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{i=1,2}\sum_{n=1}^{l_i}(\mathbf{x}_n^i - \mathbf{m}_i)(\mathbf{x}_n^i - \mathbf{m}_i)^T$$

(24)

Differentiating **J(w)** with respect to **w**, setting equal to zero, and rearranging gives

$$(\mathbf{w}^T\mathbf{S}_B\mathbf{w})\mathbf{S}_W\mathbf{w} = (\mathbf{w}^T\mathbf{S}_W\mathbf{w})\mathbf{S}_B\mathbf{w} \quad (25)$$

Since we only care about the direction of **w** and $\mathbf{S_Bw}$ has the same direction as $(\mathbf{m_2}\text{-}\mathbf{m_1})$, $\mathbf{S_Bw}$ can be replaced by $(\mathbf{m_2}\text{-}\mathbf{m_1})$ and we can drop the scalars $(\mathbf{w^TS_Bw})$and $(\mathbf{w^TS_Ww})$ to give

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 = \mathbf{m}_1) \quad (26)$$

7.Kernel Fisher Analysis{\displaystyle \mathbf {w} \propto \mathbf {S} _{W}^{-1}(\mathbf {m} _{2}-\mathbf {m} _{1}).}

To extend LDA to non-linear mappings, the data, given as the l points $\mathbf{x_i}$, can be mapped to a new feature space, **F**, via some function ϕ. In this new feature space, the function that needs to be maximized is

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B^{\phi}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W^{\phi}\mathbf{w}} \quad (27)$$

where

$$\mathbf{S}_B^{\phi} = (\mathbf{m}_2^{\phi} - \mathbf{m}_1^{\phi})(\mathbf{m}_2^{\phi} - \mathbf{m}_1^{\phi})^T$$

$$\mathbf{S}_W^{\phi} = \sum_{i=1,2}\sum_{n=1}^{l_j}(\phi(\mathbf{x}_n^i) - \mathbf{m}_1^{\phi})(\phi(\mathbf{x}_n^i) - \mathbf{m}_1^{\phi})^T$$

and

$$\mathbf{m}_i^{\phi} = \frac{1}{l_i}\sum_{j=1}^{l_i}\phi(\mathbf{x}_j^i)$$

Further, note that $\mathbf{w} \in \mathbf{F}$. Explicitly computing the mappings $\phi(\mathbf{x_i})$ and then performing LDA can be computationally expensive, and in many cases intractable. For example, **F** may be infinitely dimensional. Thus, rather than explicitly mapping the data to **F**, the data can be implicitly embedded by rewriting the algorithm in terms of dot products and using the kernel trick in which the dot product in the new feature space is replaced by a kernel function, $\mathbf{k(x,y)=\phi(x).\phi(y)}$

LDA can be reformulated in terms of dot products by first noting that **w** will have an expansion of the form

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i\phi(\mathbf{x}_i) \quad (28)$$

Then note that

$$\mathbf{w}^T\mathbf{m}_i^{\phi} = \frac{1}{l_i}\sum_{j=1}^{l}\sum_{k=1}^{l_i}\alpha_j k(\mathbf{x}_j, \mathbf{x}_k^i) = \alpha^T\mathbf{M}_i$$

where

$$(\mathbf{M}_i)_j = \frac{1}{l_i}\sum_{k=1}^{l_i}k(\mathbf{x}_j, \mathbf{x}_k^i)$$

With these equations for the numerator and denominator of **J(w)**, the equation for **J** can be rewritten as

$$J(\alpha) = \frac{\alpha^T\mathbf{M}\alpha}{\alpha^T\mathbf{N}\alpha} \quad (29)$$

Then, differentiating and setting equal to zero gives

$$(\alpha^T\mathbf{M}\alpha)\mathbf{N}\alpha = (\alpha^T\mathbf{N}\alpha)\mathbf{M}\alpha \quad (30)$$

Since only the direction of **w**, and hence the direction of α, matters, the above can be solved for α as

$$\alpha = \mathbf{N}^{-1}(\mathbf{M}_2 - \mathbf{M}_1) \quad (31)$$

Note that in practice, **N** is usually singular and so a multiple of the identity is added to it

$$\mathbf{N}\varepsilon = \mathbf{N} + \varepsilon\mathbf{I} \quad (32)$$

Given the solution for α, the projection of a new data point is given by

$$y(\mathbf{x}) = (\mathbf{w}.\phi(\mathbf{x})) = \sum_{i=1}^{l}\alpha_i k(\mathbf{x}_i, \mathbf{x}) \quad (33)$$

1) Gallery images

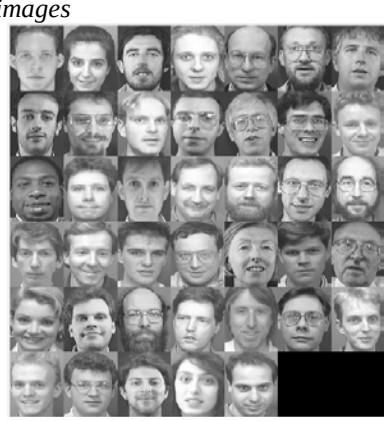

Fig. 3. First face image of all 40 people in the ORL database

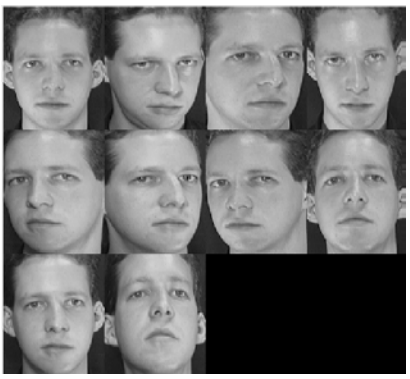

Fig. 4. All 10 images of first person from ORL database.

Here for experiment purpose the simple database AT&T ORL data base has been taken. There are 40 set of faces in the data base and each set has 10 images. The first 8 images from each set are considered for training and the remaining two images from the data set are considered for testing purpose. The total number of images considered for training are 320 and for testing are 40. The features of all the face images in the training group are extracted using Eigen feature extraction. This Eigen feature extraction preserves the edges and also the directionality of the edge information. Here SVD classifier is used. This is a non probabilistic binary classifier which looks for optimal hyperplane as a decision function. In the testing phase, the test image is taken and given to the SVD classifier for classification.

The face recognition rate is calculated as

$$\frac{\text{Total number of images in the data set which are properly matched}}{\text{Total number of persons in the dataset}} * 100 \quad (34)$$

The images considered in the numerator of (34) are the test images. These images are excluded from the dataset of the denominator.

Fig. 5. Eigen faces of first 10 and 20 images from ORL database.

Fig. 6. Eigen faces of first 30 and 40 images from ORL database.

Fig. 7. Eigen faces of first 50 and 60 images from ORL database.

Fig. 8. Eigen faces of first 70 and 80 images from ORL database.

Fig. 9. Eigen faces of first 90 and 100 images from ORL database.

Fig. 10. Eigen faces of first 200 and 300 images from ORL database.

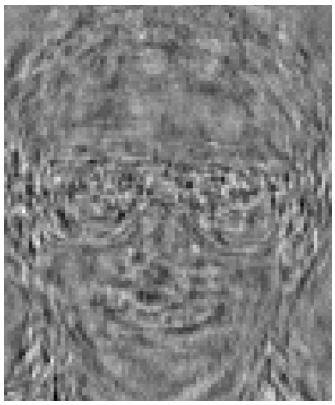

Fig. 11. Eigen faces of all 400 images from ORL database.

Figures from 5 to 11 shows the Eigen faces of first 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300 and 400 images from ORL database.

8. Experimental Results

Experiments have been conducted on proposed algorithm by taking ORL AT&T data base [29]. For training phase the first eight face images are taken and for the testing purpose the last two face images are taken.

TABLE I. Outputs of different face recognition algorithms for test face 1

Face 1												
Original AT&T Database	KFA	KPCA	LDA	PCA	G-KFA	G-KPCA	G-LDA	G-PCA	PC-KFA	PC-KPCA	PC-LDA	PC-PCA
S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1
S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2
S3	S3	S35	S3	S3	S38	S3	S3	S3	S35	S30	S38	S30
S4	S4	S4	S4	S36	S17	S4	S4	S4	S4	S4	S17	S4
S5	S5	S5	S5	S17	S5	S5	S5	S5	S5	S5	S5	S5
S6	S6	S4	S6	S13	S6	S6	S17	S6	S4	S6	S6	S6
S7	S7	S7	S7	S8	S7	S7	S7	S7	S7	S7	S7	S7
S8	S8	S30	S8	S8	S8	S8	S8	S8	S30	S8	S8	S8
S9	S9	S9	S12	S22	S9	S9	S9	S9	S9	S9	S9	S9
S10	S10	S10	S10	S10	S10	S10	S10	S10	S10	S10	S10	S10
S11	S11	S11	S34	S15	S22	S11	S11	S11	S11	S32	S22	S32
S12	S12	S12	S12	S12	S40	S12	S12	S12	S12	S12	S40	S12
S13	S13	S13	S13	S40	S25	S13	S13	S13	S13	S13	S25	S13
S14	S14	S28	S14	S14	S14	S14	S14	S14	S28	S20	S14	S20
S15	S15	S29	S15	S15	S40	S15	S15	S15	S29	S1	S40	S1
S16	S1	S16	S28	S1	S20	S16	S28	S16	S16	S16	S20	S16
S17	S17	S17	S36	S17	S17	S17	S17	S17	S17	S17	S17	S17
S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18
S19	S19	S19	S19	S25	S37	S19	S19	S19	S19	S6	S37	S6
S20	S20	S40	S20	S22	S20	S20	S20	S20	S40	S20	S20	S20
S21	S21	S21	S24	S6	S21	S21	S21	S21	S21	S21	S21	S21
S22	S22	S22	S1	S22	S22	S22	S22	S22	S22	S22	S22	S22
S23	S23	S23	S23	S23	S22	S23	S23	S23	S23	S23	S22	S23
S24	S24	S3	S24	S31	S21	S24	S24	S24	S3	S24	S21	S24
S25	S25	S7	S39	S25	S25	S25	S25	S25	S7	S38	S25	S38
S26	S26	S4	S26	S26	S6	S26	S26	S26	S4	S26	S6	S26
S27	S27	S27	S27	S14	S27	S27	S27	S27	S27	S27	S27	S27
S28	S1	S28	S28	S28	S28	S28	S18	S28	S28	S26	S28	S26
S29	S40	S30	S29	S40	S18	S29	S40	S29	S30	S29	S18	S29
S30	S30	S30	S30	S23	S30	S30	S30	S30	S30	S30	S30	S30
S31	S31	S30	S31	S31	S31	S31	S31	S31	S30	S31	S31	S31
S32	S32	S21	S32	S32	S32	S32	S32	S32	S21	S16	S32	S16
S33	S33	S33	S31	S30	S25	S33	S33	S33	S33	S35	S25	S35
S34	S34	S34	S34	S1	S34	S34	S34	S34	S34	S34	S34	S34
S35	S35	S3	S30	S35	S35	S35	S35	S35	S3	S35	S35	S35
S36	S36	S36	S36	S12	S36	S36	S36	S36	S36	S36	S36	S36
S37	S37	S32	S37	S37	S27	S37	S37	S37	S32	S37	S27	S37
S38	S38	S38	S38	S38	S6	S38	S38	S38	S38	S38	S6	S38
S39	S39	S33	S39	S39	S2	S39	S39	S39	S33	S21	S2	S21
S40	S40	S21	S40	S25	S40	S40	S40	S40	S21	S40	S40	S40

TABLE II. Outputs of different face recognition algorithms for test face 2

Face 2												
Original AT&T Database	KFA	KPCA	LDA	PCA	G-KFA	G-KPCA	G-LDA	G-PCA	PC-KFA	PC-KPCA	PC-LDA	PC-PCA
S1	S1	S1	S1	S40	S1	S1	S1	S1	S1	S1	S1	S1
S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2	S2
S3	S3	S3	S35	S3	S38	S30	S35	S3	S3	S3	S38	S30
S4	S4	S4	S4	S36	S17	S4	S4	S4	S4	S4	S17	S4
S5	S5	S5	S5	S9	S5	S5	S5	S5	S5	S5	S5	S5
S6	S17	S6	S4	S13	S6	S6	S4	S6	S6	S6	S6	S6
S7	S7	S7	S7	S8	S7	S7	S7	S7	S7	S7	S7	S7
S8	S8	S8	S30	S8	S8	S8	S30	S8	S8	S8	S8	S8
S9	S9	S12	S9	S9	S9	S9	S9	S9	S9	S9	S9	S9
S10	S10	S10	S10	S40	S10	S10	S10	S10	S10	S10	S10	S10
S11	S11	S34	S11	S15	S22	S32	S11	S11	S11	S11	S22	S32
S12	S12	S12	S12	S12	S40	S12	S12	S12	S12	S12	S40	S12
S13	S13	S13	S13	S13	S25	S13	S13	S13	S13	S13	S25	S13
S14	S14	S14	S28	S30	S14	S20	S28	S14	S14	S14	S14	S20
S15	S15	S15	S29	S40	S40	S1	S29	S15	S15	S15	S40	S1
S16	S28	S28	S16	S16	S20	S16	S16	S16	S16	S1	S20	S16
S17	S17	S36	S17	S21	S17	S17	S17	S17	S17	S17	S17	S17
S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18	S18
S19	S19	S19	S19	S16	S37	S6	S19	S19	S19	S19	S37	S6
S20	S20	S20	S40	S20	S20	S20	S40	S20	S20	S20	S20	S20
S21	S21	S24	S21	S6	S21	S21	S21	S21	S21	S21	S21	S21
S22	S22	S1	S22	S22	S22	S22	S22	S22	S22	S22	S22	S22
S23	S23	S23	S23	S23	S22	S23	S23	S23	S23	S23	S22	S23
S24	S24	S24	S3	S31	S21	S24	S3	S24	S24	S24	S21	S24
S25	S25	S39	S7	S25	S25	S38	S7	S25	S25	S25	S25	S38
S26	S26	S26	S4	S26	S6	S26	S4	S26	S26	S26	S6	S26
S27	S27	S27	S27	S14	S27	S27	S27	S27	S27	S27	S27	S27
S28	S18	S28	S28	S28	S28	S26	S28	S28	S28	S1	S28	S26
S29	S40	S29	S30	S40	S18	S29	S30	S29	S29	S40	S18	S29
S30	S30	S30	S30	S23	S30	S30	S30	S30	S30	S30	S30	S30
S31	S31	S31	S30	S31	S31	S31	S30	S31	S31	S31	S31	S31
S32	S32	S32	S21	S32	S32	S16	S21	S32	S32	S32	S32	S16
S33	S33	S31	S33	S30	S25	S35	S33	S33	S33	S33	S25	S35
S34	S34	S34	S34	S34	S34	S34	S34	S34	S34	S34	S34	S34
S35	S35	S30	S3	S35	S35	S35	S3	S35	S35	S35	S35	S35
S36	S36	S36	S36	S40	S36	S36	S36	S36	S36	S36	S36	S36
S37	S37	S37	S32	S22	S27	S37	S32	S37	S37	S37	S27	S37
S38	S38	S38	S38	S38	S6	S38	S38	S38	S38	S38	S6	S38
S39	S39	S39	S33	S39	S2	S21	S33	S39	S39	S39	S2	S21
S40	S40	S40	S21	S25	S40	S40	S21	S40	S40	S40	S40	S40

All the 12 different prominent face recognition algorithms namely PCA, KFA, KPCA, LDA, Gabor based algorithms and Phase Congruency based algorithms are compared [35].

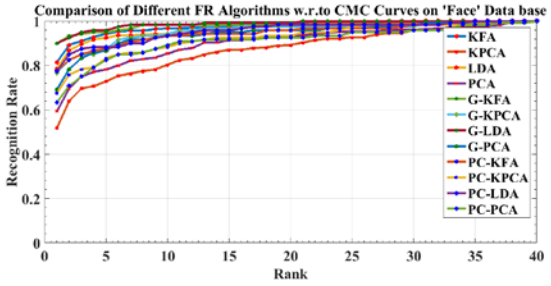

Fig. 12. Comparison of FR algorithms with respect to CMC curves

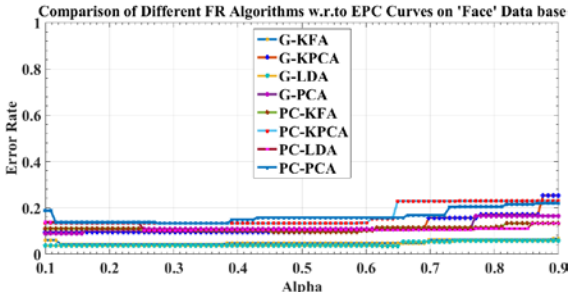

Fig. 13. Comparison of FR algorithms with respect to CMC curves

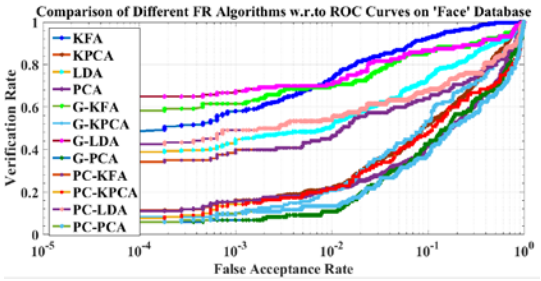

Fig. 14. Comparison of FR algorithms with respect to CMC curves

TABLE III. Different Datasets and their total number of images and persons

Data base	Total number of persons	Pose, Illumination and facial expression variations	Total number of face images
Yale Database [27]	15	11	165
Yale Face Database 'B' [28]	10	64 illumination 9 poses	5760
MIT CBCL [30]	10	524	5240
University of Essex, UK [31]	395	20	7900
JAFFE [32]	60	7	420
Sheffield [33]	20	25-30	564
Caltech [34]	27	10-20	450
Senthil Kumar IRTT v1.2 [36]	10	10	100
Senthil Kumar IRTT v1.1 [37]	5	16	80

TABLE IV. Different Datasets and their total number of images and persons used in this experiment

Database	Total number of people considered	Total number of faces per person	Faces considered for testing	Faces considered for training	Face recognition rate (in %)	
					PCA Algorithm	KPCA Algorithm
Yale Database	15	11	9	2	88.26	97.25
Yale Face Database 'B'	10	10	8	2	80.01	81.38
MIT CBCL	10	10	8	2	64.25	61.07
University of Essex, UK	40	20	16	4	70.0	77.0
JAFFE	60	7	6	1	71.2	80
Sheffield	20	25	20	5	61.8	77.5
Caltech	25	15	13	2	70.12	62.5
Senthil Kumar IRTT v1.2	10	10	8	2	86.5	79.26
Senthil Kumar IRTT v1.1	5	16	14	2	75.8	77.9

In case of testing images taken are more than one, then the face recognition rate is calculated by taking the average of the face recognition rates of all the testing images.

The performance metrics for different algorithms shown below are with ORL database. Table IV and Table V shows the comparison of face recognition rates of PCA and KPCA. One algorithm based on the Eigen face features and another algorithm based on the fisher face features are compared.

9. Conclusions

In this paper, three performance metrics for face recognition algorithm are introduced. 12 prominent face recognition algorithms based on Eigen face features and Fisher face features are compared with respect to these performance metrics. Ten face data bases are taken for comparing the face recognition rate of these algorithms.

References

[1] M. Turk and A. Pentland (Jun. 1991). "Face Recognition Using Eigenfaces." Proceedings of CVPR IEEE Computer Society. [Online]. pp. 586-591. Available: https://www.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf .

[2] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Muller. "Fisher Discriminant Analysis with Kernels." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8B05D1BE9201C5AACD3A040ED70C4B48?doi=10.1.1.35.9904&rep=rep1&type=pdf.

[3] Paul Viola and Michael J. Jones (Jul. 2001). "Robust Real-Time Object Detection." Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling. [Online]. 4. Available: https://www.cs.cmu.edu/~efros/courses/lbmv07/papers/viola-ijcv-01.pdf.

[4] Paul Viola and Michael J. Jones (2001). "Rapid Object Detection Using A Boosted Cascade Of Simple Features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society. [Online]. 1, pp.I511- I518. Available: https://www.cs.cmu.edu/~efros/courses/lbmv07/papers/viola-cvpr-01.pdf.

[5] Paul Viola and Michael J. Jones (May 2004). "Robust real-time face detection." International journal of computer vision. [Online] 57(2) pp. 137–154. Available: http://www.vision.caltech.edu/html-files/ee148-2005-spring/pprs/viola04ijcv.pdf.

[6] Bernhard Scholkopf, Alexander Smola, and Klaus Robert Muller (Dec 1996). "Nonlinear Component Analysis as a Kernel Eigen Problem." Technical report no. 44. Max-Planck-institut Für Biologische Kybernetik. [Online]. Available: ftp://ftp.cfar.umd.edu/.snapshot/hourly.3/pub/.../scholkopf98-kernel.pdf.

[7] V. I. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, Jul. 1997.

[8] Lindsay I Smith (Feb. 2002). "A Tutorial on Principal Components Analysis". [Online]. Available: www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

[9] C. E. Thomaz and G. A. Giraldi, "A New Ranking Method For Principal Components Analysis And Its Application To Face Image Analysis," Image and Vision Computing, vol. 28, no.6, pp. 902-913, Jun. 2010.
[10] Jon Shlens. "A Tutorial On Principal Component Analysis - Derivation, Discussion and Singular Value Decomposition." [Online]. Available: https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf.
[11] Max Welling. "Fisher Linear Discriminant Analysis." [Online]. Available: www.ics.uci.edu/~welling/classnotes/papers_class/Fisher-LDA.pdf.
[12] Aly A. Farag and Shireen Y. Elhabian. "A Tutorial on Data Reduction Linear Discriminant Analysis (LDA)." [Online]. Available: www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid437773.pdf.
[13] Bernhard Scholkopf, Alexander Smola, and Klaus Robert Muller. "Kernel Principal Component Analysis." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.7613&rep=rep1&type=pdf.
[14] Xiaoying Wang, Haifeng Hu, and Jianquan Gu, "Pose robust low-resolution face recognition via coupled kernel-based enhanced discriminant analysis," IEEE/CAA Journal of Automatica Sinica, vol. 3, no. 2, pp. 203-212. Feb. 2016.
[15] Max Welling. "Kernel Principal Components Analysis." [Online]. Available: www.ics.uci.edu/~welling/classnotes/papers_class/Kernel-PCA.pdf.
[16] Rita Osadchy. "Kernel PCA." [Online]. Available: www.cs.haifa.ac.il/~rita/uml_course/lectures/KPCA.pdf.
[17] Ling Chen, Hong Man, and Ara V. Nefian (2005). "Face recognition based multi-class mapping of fisher scores." Pattern recognition, special issue on image understanding for digital photographs. [Online]. 38 pp. 799-811. Available: http://dl.acm.org/citation.cfm?id=1746538.
[18] Seung-Jean Kim, Alessandro Magnani, and Stephen Boyd. "Optimal Kernel Selection in Kernel Fisher Discriminant Analysis." *[Online]. Available:* www.autonlab.org/icml_documents/.../059_Optimal_Kernel_Selec.pdf.
[19] Samy Bengio and Johnny Mari´ethoz. "The Expected Performance Curve: a New Assessment Measure for Person Authentication." [Online]. Available: publications.idiap.ch/downloads/reports/2004/bengio_2004_odyssey.pdf.
[20] Samy Bengio, Johnny Mari´ethoz, and Mikaela Keller. "The Expected Performance Curve." [Online]. Available: bengio.abracadoudou.com/publications/pdf/bengio_2005_icml.pdf.
[21] Zhenhua Chai, Zhenan Sun, Heydi Méndez-Vázquez, Ran He, and Tieniu Tan, "Gabor Ordinal Measures for Face Recognition," IEEE Transactions on Information Forensics and Security, vol. 9, no. 1, pp-14-26. Jan. 2014.
[22] Javier R. Movellan. "Tutorial on Gabor Filters." [Online]. Available: mplab.ucsd.edu/tutorials/gabor.pdf.
[23] Vitomir štruc and Nikola paveši´c (May 2010). "The complete Gabor-Fisher Classifier for Robust Face Recognition." EURASIP journal on Advances in Signal Processing. [Online]. pp. 1-26. Available: http://luks.fe.uni-lj.si/sl/osebje/vitomir/pub/asp2010.pdf.
[24] Vitomir štruc and Nikola paveši´c (2009). "Gabor-Based Kernel Partial-Least-Squares Discrimination Features for Face Recognition." INFORMATICA. [Online]. 20(1) pp. 115–138. Available: http://www.mii.lt/informatica/pdf/info744.pdf.
[25] *Michael J. Lyons, Shigeru Akemastu, Miyuki Kamachi, and Jiro Gyoba. "Coding facial expressions with Gabor wavelets." 3rd IEEE international conference on automatic face and gesture recognition. [Online]. Available:* http://www.kasrl.org/fg98-1.pdf.
[26] Vitomir štruc and Nikola Paveši´c, "Phase Congruency Features for Palm Print Verification," IET Signal Processing, vol. 3, no. 4, pp. 258– 268. Jul. 2009.
[27] Yale face database. [Online]. Available:http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip.
[28] Yale face B database. [Online]. Available: http://vision.ucsd.edu/~iskwak/extyaledatabase/extyaleb.html.
[29] ORL Database. AT&T Laboratories, Cambridge. [Online]. Available: http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.zip.
[30] The MIT-CBCL face recognition database. [Online]. Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html.
[31] Dr Libor Spacek. University of Essex, UK. [Online]. Available: http://cswww.essex.ac.uk/mv/allfaces/index.html.
[32] JAFFE Database. [Online]. Available: http://www.kasrl.org/jaffe.html.
[33] Sheffield. [Online]. Available: http://www.sheffield.ac.uk/eee/research/iel/research/face.
[34] Markus Weber. Frontal face dataset, California Institute of Technology. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html.
[35] Vitomir ˇ Struc. "The PhD face recognition toolbox." [Online]. Available: http://luks.fe.uni-lj.si/sl/osebje/vitomir/face_tools/PhDface/.
[36] Senthil. IRTT Face Database Version1.2. [Online]. Available: http://www.geocities.ws/senthilirtt/Senthil_IRTT_FaceDatabase_Version1.2.zip.
[37] Senthil. IRTT Face Database Version1.1. [Online]. Available: http://www.geocities.ws/senthilirtt/Senthil_IRTT_FaceDatabase_Version1.1.zip.
[38] Harihara Santosh Dadi and P G Krishna Mohan, "Performance Evaluation of Eigen faces and Fisher faces with different pre-processed Data sets," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 4, no. 5, pp. 2110 – 2116. May 2015.
[39] Harihara Santosh Dadi and P G Krishna Mohan, "Enhancement of Face Recognition Rate by Data Base Pre-processing," International Journal of Computer Science and Information Technologies, IJCST, vol. 6, no. 3, pp. 2978-2984. Jun. 2015.

D. Harihara Santosh obtained his B. Tech. and M. Tech degrees from JNT University, Hyderabad in the year 2005 and 2010. Presently he is pursuing Ph.D, in Video Processing at JNTU, Hyderabad. He is presently pursuing his Ph.D. under the Guidance of Dr. P.G. Krishna He has 9 publications in both International and National Journals and presented 22 papers at various International and National Conferences. His areas of interests are Image and Video Processing.

P.G.Krishna Mohan presently working as Professor in Institute of Aeronautical College of Engineering, Hyderabad. He Worked as Head of ECE Dept. , Member of BOS for ECE faculty at University Level, Chairman of BOS of EIE group at University level, Chairman of BOS of ECE faculty for JNTUCEH, Member of selection committees for Kakitiya, Nagarjuna University, DRDL and convener for Universite a Hidian committees. He has more than 43 papers in various International and National Journals and Conferences. His areas of interests are Signal Processing, Communications.