Performance Analysis of Two Methods for Dimension Reduction in Face Recognition

Anika Gugnani, R.B Dubey

ECE, Hindu College of Engg. Sonepat, India
anikagugnani10@gmail.com, rbdubeyster@gmail.com

ABSTRACT
Face recognition has been a fast growing, challenging and interesting area in real time applications. This work aims to compare the two renowned techniques of feature dimension reduction on the basis of the classification results of three classifiers used to fulfill the task of face recognition. With the fast increasing quantity and complexity of data in an information-rich age, it becomes difficult, challenging or even impossible for engineers or analysts to deal with raw data directly. Dimensionality reduction provides an efficient way for data abstraction and representation as well as feature extraction. It aims to detect intrinsic structures of data and to extract a reduced number of variables that capture and retain the main features of the high-dimensional data. Principal component analysis (PCA) has long been a simple, efficient technique for dimensionality reduction. A newer nonlinear method local linear embedding (LLE) has been proposed for increasingly complex nonlinear data. In this research work, we investigate and compare linear PCA and nonlinear LLE for face recognition. Experimental results on real-world face database show that these linear and nonlinear methods when compared over performance, the non-linear method LLE yield better performance. The classification is carried out using k-nearest neighbor, probabilistic neural network and support vector machine to identify the face.

1. INTRODUCTION
Face Recognition is a type of biometric software application that can identify a specific individual in a digital image by analyzing and comparing patterns. The human face plays an important role in our social interaction, conveying people’s identity. Using the human face as a key to security, biometric face recognition technology has received significant attention in the past several years due to its potential for a wide variety of applications in both law enforcement and non law enforcement.

Linear representation based face recognition [1], for a discriminative dictionary is learned from the training samples to get the query sample for better classification. Dimensionality reduction is also an important issue for face recognition which not only reduces the storage space of face images but also enhances the discrimination of face feature. Existing methods mostly perform dimensionality reduction and dictionary learning separately, which may not fully exploit the discriminative information in the training samples. To fulfill this task of reducing dimension, the joint learning makes the learned projection and dictionary better fit with each other to get a more effective face classification. A face recognition system using a new method based on recent advances in compressed sensing theory [2] is built proposing a method for recognizing faces that is robust to certain types and levels of occlusion.

Face recognition from single sample per person (SSPP) is extremely challenging because only one sample is available for each person. Many discriminat analysis methods, like Fisher faces and its numerous variants, have achieved great success in face recognition, these methods cannot work in this scenario, because more than one sample per person are needed to calculate the within-class scatter matrix. To remove this problem Adaptive Discriminant Analysis (ADA) is adopted in [3] which the within-class scatter matrix of each enrolled subject is inferred using single sample, by leveraging a generic set with multiple samples per
person assuming that subjects who look alike to each other generally share similar within-class variations. In ADA, a limited number of neighbors for each single sample are first determined from the generic set by using KNN regression and within-class scatter matrix of this single sample is inferred as the weighted average of the within-class scatter matrices of these neighbors based on the arithmetic mean. The symmetry of the faces is exploited to generate new samples and devise a representation based method to perform face recognition. The new training samples reflect some possible appearance of the face to perform a two-step classification, which ultimately uses a small number of classes that are near to the test sample to represent and classify it and has a similar advantage as the sparse representation method [4]. A multi-resolution feature fusion for face recognition provided in [5] proposes a face recognition approach using information about face images at higher and lower resolutions to enhance the information content of the features that are extracted and combined at different resolutions. As the features from different resolutions should closely correlate with each other, employing the cascaded generalized canonical correlation analysis (GCCA) to fuse the information to form a single feature vector for face recognition.

A novel method of supervised and unsupervised multi-linear neighborhood preserving projection for face recognition is provided in [6]. Unlike conventional neighborhood preserving projections, it operates directly on tensorial data rather than vectors or matrices and solves problems of tensorial representation for multi-dimensional feature extraction, classification and recognition. The coding residual map learning based adaptive masking for fast face recognition representing the face recognition schemes with sparse coding coefficients and coding residual have demonstrated good robustness to face occlusion. Finally the face image is identified by masking the detected occlusion pixels from face representation [7].

The attempt to extract sparse error of robust PCA for face recognition with varying illumination and occlusion defined by two descriptors (sparsity and smoothness) to represent characteristic of the sparse error component and give two recognition protocols which are the weighted based method and the ratio based method to classify face images is given in [8]. An effective and efficient approach in handling lighting variations for face recognition that presents fascinating insights and conclusions to design better illumination preprocessing methods is described in [9]. A method of face recognition with POEM (patterns of oriented edge magnitudes) descriptor is depicted in [10], fully balance the three concerns needed for face recognition that are computational cost, robustness, and discriminative power by applying a self-similarity based structure on oriented magnitudes and prove that it addresses all three criteria which is around 20 times faster than Gabor-based methods. A feature extraction method by utilizing an error estimation equation based on the Euclidean distance and proposes to use classification errors in the transformed feature space, which are estimated using the error estimation equation, as a criterion for feature extraction is presented in [11]. The construction of linear transformation for feature extraction is conducted using an iterative gradient descent algorithm, so that the estimated classification error is minimized. Due to the ability to predict error, it is possible to determine the minimum number of features required for classification. A pose-robust face recognition method to handle the challenging task of face recognition in the presence of large pose difference between gallery and probe faces is proposed in [12]. The method exploits the sparse property of the representation coefficients of a face image over its corresponding view dictionary. By assuming the representation coefficients are invariant to pose, we can synthesize for the probe image a novel face image which has smaller pose difference with the gallery faces. To implement neural networks for face recognition a system combines local image sampling, a self-organizing map neural network and a conventional neural network is presented in [13]. An in-depth insight into face recognition using the PCA and Error Back-Propagation in which real-time face region was detected by suggesting the rectangular feature-based classifier and the robust detection algorithm that satisfied the efficiency of computation and detection performance [13]. By using the detected face region as recognition input image, face recognition method combined with PCA and the multi-layer network which is one of the intelligent classifications was suggested and its performance was evaluated. As a pre-processing algorithm of input face image, this method computes the eigen face through PCA and expresses the training images with it as a fundamental vector. Each image takes the set of weights for the fundamental vector as a feature vector and it reduces the dimension of image at the same time, and then the face recognition is performed by inputting the multi-layer neural network. PCA algorithm for face recognition is a successful technique that has been used to recognize faces in images. However, high computational cost and dimensionality is a major problem of this technique. There is evidence that PCA can outperform over many other techniques when the size of the database is small. A fast PCA based face recognition algorithm is proposed in [14].

An efficient approach for face recognition based on common eigen values based on the polynomial coefficients, covariance matrix and algorithm on common eigen values is provided in [15]. The main advantage of the approach is that the identification of similarity between human faces is carried out without computing actual eigen values and eigenvectors. A
symmetric matrix is calculated using the polynomial coefficients-based companion matrices of two compared images. The nullity of a calculated symmetric matrix is used as similarity measure for face recognition. The value of nullity is very small for dissimilar images and distinctly large for similar face images. Face recognition based on PCA image reconstruction and LDA are provided in [16]. The inner classes covariance matrix for feature extraction is used as generating matrix and then eigenvectors from each person is obtained, then we obtain there constructed images. the residual images are computed by subtracting reconstructed images from original face images. Furthermore, the residual images are applied by LDA to obtain the coefficient matrices. Finally, the features are utilized to train and test support vector machine for face recognition. The need to analyze large amounts of multivariate data raises the fundamental problem of dimensionality reduction; a locally linear embedding (LLE) approach is introduced. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by images of faces as discussed in [17]. LLE by linear programming including the fact dimensionality reduction has always been one of the most challenging tasks in the field of data mining [18].

Face detection in gray scale images using locally linear embeddings is focused in [19]. The first objective of this research is to apply the LLE algorithm to 2D facial images to obtain their representation in a sub-space under the specific conditions. The low-dimensional data are then used to train SVM classifiers to label windows in images as being either face or non-face. Six different databases of cropped facial images, corresponding to variations in head rotation, illumination, facial expression, occlusion and aging, were used to train and test the classifiers. The second objective was to evaluate the feasibility of using the combined efficacy of the six SVM classifiers in a two-stage face detection approach. The enhanced supervised LLE which attempts to make the interclass dissimilarity definitely larger than the intraclass dissimilarity in an effort to strengthen the discriminating power and generalization ability of embedded data representation is presented in [20]. Experimental results demonstrate that LLE obtains better performance on face recognition compared with PCA. An introduction to dimensionality reduction which is an important task in machine learning, as it facilitates classification, compression, and visualization of high-dimensional data by mitigating undesired properties of high-dimensional spaces depicted in [21]. New techniques for dimensionality reduction aim at identifying and extracting the manifold from the high-dimensional space. LLE is considered as an effective algorithm for dimensionality reduction [22]. In particular, it proves that firstly, there always exists a linear mapping from the high-dimensional space to the low-dimensional space such that all the constraint conditions in the LLE can be satisfied. The implication of the existence of such a linear mapping is that the LLE cannot guarantee a one-to-one mapping from the high-dimensional space to the low-dimensional space for a given data set; secondly, for a given high-dimensional data set, there always exist a local distance-preserving LLE.

The main purpose of this study is recognition of face and comparison of linear (PCA) and non linear (LLE) methods of recognition. The rest of this paper is organized as follows. Section 2 deals with the proposed methodology. The experimental results are discussed in Section 3. Finally, the concluding remarks are given in Section 4.

2. PROPOSED METHODOLOGY

A schematic block diagram of proposed methodology is outlined in Fig. 1.

![Block diagram of proposed methodology](image)

Fig. 1: Block diagram of proposed methodology

After a face is normalized, feature extraction is performed to provide effective information that is useful for distinguishing between faces of different persons and stable with respect to the geometrical and photometrical variations. For face matching, the extracted feature vector of the input face is matched against those of enrolled faces in the database as shown in Fig. 1.

2.1 Proposed comparison techniques

2.1.1 Principal component analysis (PCA)

PCA is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [17]. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance and each succeeding component has the highest variance possible under the constraint that it be orthogonal to the preceding components.

The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigen space projection. Eigen space is calculated by identifying the
A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. Let’s suppose we have M vectors of size N (= rows of image x columns of image) representing a set of sampled images. $p_j$’s represent the pixel values.

$$x_i = [p_1 : \cdots : p_M]^T; \ i = 1; \cdots ; M$$  \hspace{1cm} (1)

Let $m$ represent the mean image.

$$m = \frac{1}{M} x_i$$ \hspace{1cm} (2)

and let $\overline{\omega}$ be defined as mean centered image

$$\overline{\omega} = x_i - \mu$$ \hspace{1cm} (3)

Our goal is to find a $\omega$ which have the largest possible projection onto each $\omega_i$ $\omega_i$’s are the eigenvectors and eigenvalues of the covariance matrix

$$Y_i = \frac{1}{M} \sum_{j=1}^{M} (\omega_i^T \omega_i)^2$$ \hspace{1cm} (4)

is maximized with the ortho normality constraint as

$$\sum_{i=1}^{M} \omega_i \omega_i^T = I$$ \hspace{1cm} (5)

It has been shown that the $\omega_i$’s and $y_i$’s are given by the eigenvectors and eigenvalues of the covariance matrix

$$C = WW^T$$ \hspace{1cm} (6)

where $W$ is a matrix composed of the column vectors $w_i$ placed side by side. The size of $C$ is $NN$ which is very large. A theorem states that the vectors $\omega_i$ and scalars $\lambda_i$ can be obtained by solving for the eigenvectors and eigenvalues of the $MLEM$ matrix $WW^T$.

Let $d_i$ and $\mu_i$ be the eigenvectors and eigenvalues of $WW^T$, respectively.

$$WW^T = \sum_{i=1}^{M} \omega_i \omega_i^T$$ \hspace{1cm} (7)

By multiplying left to both sides by $W$

$$WW^T(Wd_i)^T = \mu_i d_i$$ \hspace{1cm} (8)

A facial image can be projected onto $M_i \ll M$ dimensions by computing

$$= [v_1 v_2 : \cdots : v_n]^T$$ \hspace{1cm} (9)

Here, $v_i = e_i^T w_i$, $v_i$ is the $i^{th}$ coordinate of the facial image in the new space, which are the principal components. The vectors $\omega_i$ are also images, and called, eigenimages, or eigenfaces. describes the contribution of each eigen face in representing the facial image by treating the eigen faces as a basis set for facial images. The simplest method for determining which face class provides the best description of an input facial image is to find the face class $k$ that minimizes the Euclidean distance

$$z_k = \min (|y - \mu_k|)$$ \hspace{1cm} (10)

where $\mu_k$ is a vector describing the $k^{th}$ face class. If $e_k$ is less than some predefined threshold $\theta_k$, a face is classified as belonging to the class $k$ [23].

2.1.2 Local linear embedding (LLE)

The LLE is nonlinear generalization of PCA [18]. Alongside earlier applications in visualization and classification, the scheme has most recently found use to such tasks as 3D-object pose estimation, face membership authentication, multi pose face synthesis, facial animation, image denoising, hyper spectral image processing, digital watermarking, feature extraction, gait recognition, and manifold learning to name a few. In most applications, LLE is invoked as a ready-to-use dimensionality reduction tool [7].

The LLE algorithm [24] is based on simple geometric intuitions. Suppose the data consist of $N$ real-valued vectors $X_i$, each of dimensionality $D$, sampled from some smooth underlying manifold. We can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. In the simplest formulation of LLE, one identifies $K$ nearest neighbors per data point, as measured by Euclidean distance. Reconstruction errors are then measured by the cost function:

$$\varepsilon(W) = \frac{1}{J} \sum_{j} \sum_{i} (x_i^j - \sum_{l} W_{lj} y_i^l)^2$$ \hspace{1cm} (11)

that each data point is reconstructed only from its neighbors, enforcing $W_{lj} = 0$ if $X_j$ does not belong to this set; second, that the rows of the $W_{lj}$ weight matrix sum to one $W_{lj}$.

In the final step of the algorithm, each high dimensional observation $X_i$ is mapped to a low dimensional vector $Y_i$ representing global internal coordinates on the manifold. This is done by choosing $d$-dimensional coordinates to minimize the embedding cost function:

$$\varphi(Y) = \frac{1}{J} \sum_{j} (||x_i^j - \sum_{l} W_{lj} y_i^l||^2$$ \hspace{1cm} (12)
This cost function—like the previous one—is based on locally linear reconstruction errors, but here we fix the weights $W_{ij}$ while optimizing the coordinates $Y_i$. The embedding cost in Eqn. (2) defines a quadratic form in the vectors $Y_i$. Subject to constraints that make the problem well-posed, it can be minimized by solving a sparse $N \times N$ eigenvector problem, whose bottom $d$ non-zero eigenvectors provides an ordered set of orthogonal coordinates centered on the origin. Reconstruction weights for each data point are computed from its local neighborhood—indeed the weights for other data points—the embedding coordinates are computed by an $N \times N$ eigen solver, a global operation that couples all data points in connected components of the graph defined by the weight matrix. The different dimensions in the embedding space can be computed successively; this is done simply by computing the bottom eigenvectors from Eqn. (12) one at a time.

The face recognition system consists of training and testing stage. Fig. 2 shows the flowchart of the system. LLE serves as a basis of extracting the feature vector of the subject’s face expression. Inherently, the learned models serve as the training data for minimum-distance classifier. The feature vector extracted with LLE is subsequently used by the minimum-distance classifier to assign particular probability of each expression.

![Fig. 2: The flowchart of face recognition system using LLE](image)

### 2.2 Classification

Classification refers to the process where the features calculated in the feature extraction stage are used by the classifier to map the object into proper object classes. A précised classification can robustly reduce classification time for a large database. Support vector machine, k-nearest neighbor and probabilistic neural network classifier appear to be the most universal and less clumsy classifiers.

#### 2.2.1 Support vector machine (SVM) classifier

A SVM finds a separating hyperplane by maximizing the margin between two classes in the high dimensional input space. A non-linear SVM is a supervisory binary classifier. SVM constructs a maximal margin linear classifier by mapping the original features via a kernel function [25], [26]. The Gaussian kernel is defined as

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$  \hspace{1cm} (13)

Where $x_i$ and $x_j$ are two feature vectors and $\sigma$ controls the size of the Gaussian kernel. The decision function is taken as

$$f(x) = \text{sgn}(\sum_{i=1}^l a_i K(x_i, x) + b)$$  \hspace{1cm} (14)

The sign of the function gives two classes: one corresponding to the positive sign and other to the negative sign. A proper combination of the kernel function and the training-test partition can maximize the performance of SVM-based classifier.

#### 2.2.2 k-Nearest neighbor (k-NN) classifier

In pattern recognition, the k-Nearest Neighbors algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the $k$ closest training examples in the feature space. In $k$-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its $k$ nearest neighbors. If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. $k$-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The $k$-NN algorithm is among the simplest of all machine learning algorithms. Both for classification and regression, it can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where $d$ is the distance to the neighbor. The neighbors are taken from a set of objects for which the class (for $k$-NN classification) or the object property value (for $k$-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

In the classification phase, $k$ is a user-defined constant, and an unlabeled vector is classified by assigning the label which is most frequent among the $k$ training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be...
used, such as the overlap metric. Often, the classification accuracy of k-NN can be improved significantly if the distance metric is learned with specialized algorithms such as large margin nearest neighbor or neighborhood components analysis.

The nearest neighbor classifier relies on a distance function between the points. It basically compares the feature vector of the substitution class image and the feature vectors stored in the database. Nearest neighbor is obtained by finding the distance between the paradigm signal and the database. Suppose C1, C2, C3, … , Ck are the k clusters in the database. By measuring the distance \( d(x(q), C_k) \) between \( x(q) \) and the \( k \)th cluster \( C_k \), the class is obtained. The feature vector with minimum difference is found to be the closest matching vector.

2.2.3 Probabilistic neural network classifier

The neural networks are trained initially using a training set and a weight matrix is obtained for every neural network. Next, classification is performed depending upon the neural network outputs at every node. A new feature vector or approximator function is taken to train the training set which classifies the unspoken words into respective class, whose probability is closest to the feature vector of the training set. The PNN architecture consists of four layers: input layer, pattern layer, summation layer, and decision layer.

Fig. 3 shows a PNN structure that recognizes \( c \) classes. The first layer shows the input pattern with \( n \) features. The number of nodes in the pattern layer is equal to the number of training instances. The number of nodes in the summation layer is equal to the number of classes in the training instances. The input layer is fully connected to the pattern layer. The input layer does not perform any computation and simply distributes the input to the neurons in the pattern layer. The pattern layer is semi-connected to the summation layer. Each group of training instances corresponding to each class is just connected to one node in the summation class. In other words, the summation units simply sum the inputs from the pattern units that correspond to the category from which the training pattern was selected.

Fig. 3. Organization of different layers of PNN

The PNN works by creating a set of multivariate probability densities that are derived from the training vectors presented to the network [27]. The input instance with unknown category is propagated to the pattern layer. Once each node in the pattern layer receives the input, the output of the node will be computed

\[
p_i^c = \frac{1}{(2\pi)^{d/2} \sigma_d^{d/2}} \exp \left[ -\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2} \right]
\]

(15)

Where \( d \) is the number of features of the input instance \( x \). \(<\text{math}>\sigma</text/math>\) is the smoothing parameter, and \(<\text{math}>x_{ij}</text/math>\) is a training instance corresponding to category \( c \). The summation layer neurons compute the maximum likelihood of pattern \( x \) being classified into \( c \) by summarizing and averaging the output of all neurons that belong to the same class

\[
p_i(x) = \frac{1}{(2\pi)^{d/2} \sigma_d^{d/2}} \sum_{i=1}^{N} \exp \left[ -\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2} \right]
\]

(16)

where \(<\text{math}>N</text/math>\) denotes the total number of samples in class \( c \). If the \( a \) priori probabilities for each class are the same, and the losses associated with making an incorrect decision for each class are the same, the decision layer unit classifies the pattern \( x \) in accordance with the Bayes’s decision rule based on the output of all the summation layer neurons

\[
C(x) = \arg \max \{ p_i(x) \}, i = 1, 2, \ldots, c
\]

where \( C(x) \) denotes the estimated class of the pattern \( x \) and \( m \) is the total number of classes in the training samples. If the \( a \) priori probabilities for each class are not the same, and the losses associated with making an incorrect decision for each class are
different, the output of all the summation layer neurons will be
\[ C(x) = \arg \max \{ \pi(x) \cdot \text{cost } i(x) \cdot \text{apro } i(x) \}, \]
\[ i = 1, 2, \ldots c \]
where \( \text{cost } i(x) \) is the cost associated with misclassifying the input vector and \( \text{apro } i(x) \) is the prior probability of occurrence of patterns in class \( c \).

3. RESULTS AND DISCUSSIONS
Classification results of three classifiers: (a) SVM (b) KNN (c) PNN with different values extracted from the two dimensionality reduction techniques PCA and LLE are presented in this section. Firstly, the face images are decomposed in MATLAB environment and the features are extracted. The extracted features are fed into different classifiers and the performance of two types of techniques is tested for effective classification rate of face images.
Section 3.1 gives detail about the data set used in this work methodology. Section 3.2 and Section 3.3 gives the analysis and classification results of classifiers. Hence, the work is divided into three major parts; first the face images set is generated, then the images are analyzed. From their analysis, different features are extracted and then using these features, classification is performed yielding better results with LLE compared to PCA. The comparison results are given in section 3.3.

3.1 Database
Face databases are important because they are necessary for most methods which need to be trained with a training set. If effective face database is constructed, face recognition methods may give better results, so face databases are important to test the various methods. The proposed approach is tested on The Japanese Female Facial Expression (JAFFE) database. The database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects. The image size is 256 x 256.

Fig. 4: Cropped Version of JAFFE Database
Training data set consists of 40 images, having the different facial expressions of Japanese female models (19 images of one model and other 21 of second model). These 2 sets of images of different model are denoted by class 1 and class 2. Ten test images are entered as test set to analyze the desired performance. Feature vector is extracted from each image using PCA and LLE. When the test image and train image of the same person with same expression are compared, the matching score is referred to as genuine; otherwise imposter giving results in form of class 1 or class 2 as depicted.

3.2 Classification Results
The classification results of the three classifiers used for analyzing the performance of two dimensionality reduction techniques PCA and LLE are given in further sections.

3.2.1 SVM Classifier
The results of SVM classifier are shown in below figures applied on PCA and LLE algorithms for two classes 1 and classes 2 using linear kernel function for both algorithms under similar conditions.
Fig 5(a): Results of SVM classifier over PCA for faces of class 1

Fig 5(b): Results of SVM classifier over PCA for faces of class 2.
Fig. 5(c): Results of SVM classifier over LLE for faces of class 1.

Fig. 5(d): Results of SVM classifier over LLE for faces of class 2.

Fig. 5(a) - 5(d) gives the recognition result of SVM classifier, as a function of the number of testing subjects. As we know, SVM has been used in many fields because it has good generalization ability even with the limited training samples. In this experiment, it gives a base to analyze the performance of the two methods effectively.

3.2.2 KNN Classifier

The images belonging to particular class in the high dimensional space will be also the neighbors in the lower dimensional space and then the minimum distance method is used to process the feature matrix. For each test facial image, we compare the Euclidean distance between the test image and the center of every known class in the lower dimensional space. Whichever class space having the minimum distance to the test image, the corresponding class will be designated as the class contained in the test image.
Fig. 6(a) (i) Results of KNN classifier over PCA for faces of class 1.

Fig. 6(a) (ii) Results of KNN classifier using PCA for faces of class 1.

Fig. 6(b) (i) Results of KNN classifier over PCA for faces of class 2.
Fig. 6(b) (ii) Results of KNN classifier using PCA for faces of class 2.

Fig. 6(c) (i) Results of KNN classifier over LLE for faces of class 1.
The comparative face recognition performance of the PCA and LLE algorithm (K = 6) is shown in Fig. 4.10 - 4.13. Because face images in JAFFE database vary slightly in illumination condition.

3.2.3 PNN Classifier

PNN classifier classifies the results of two algorithms applied on the basis of probability densities that are derived from the training vectors. If the a priori probabilities for each class are not the same, misclassification results and the losses associated with making an incorrect decision for each class are different associated with a cost for misclassifying the input vector.

Fig. 6(c) (ii) Results of KNN classifier using LLE for faces of class 1.

Fig. 6(d) (i) Results of KNN classifier over LLE for faces of class 2.

Fig. 6(d) (ii) Results of KNN classifier using LLE for faces of class 2.
Fig. 7(a) Results of PNN classifier using PCA for faces of class 1.

Fig. 7(b) Results of PNN classifier using PCA for faces of class 2.

Fig. 7(c) Results of PNN classifier using LLE for faces of class 1
Generally, SVM, KNN, and PNN all can be used to solve this classification problem; SVM adjusts their support vectors automatically during training, while developer needs to decide the number of hidden layers in PNN. SVM works better with small training set. The difference is based on parameter setting. If a number of support vectors are there, SVM becomes a slow predictor while neural network classifier makes a speedy prediction with smaller model size. Training time is much less for SVMs and KNNs as compared to PNNs.

3.3 Comparison results

This section presents the comparison between the classification results. The bases of comparison have been taken to be the three performance parameters as discussed below.

3.3.1 The results of the SVM classifier

Table 1: Analysis of performance parameter by using SVM classifier

<table>
<thead>
<tr>
<th>Method Used</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Dimension Reduction Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA + KNN</td>
<td>32</td>
<td>08</td>
<td>80%</td>
</tr>
<tr>
<td>LLE + KNN</td>
<td>33</td>
<td>07</td>
<td>83%</td>
</tr>
</tbody>
</table>

Generally, SVM, KNN, and PNN all can be used to solve this classification problem; SVM adjusts their support vectors automatically during training, while developer needs to decide the number of hidden layers in PNN. SVM works better with small training set. The difference is based on parameter setting. If a number of support vectors are there, SVM becomes a slow predictor while neural network classifier makes a speedy prediction with smaller model size. Training time is much less for SVMs and KNNs as compared to PNNs.

3.3.2 The results of the KNN classifier

Table 2: Analysis of performance parameter by using KNN classifier

<table>
<thead>
<tr>
<th>Method Used</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Dimension Reduction Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA + KNN</td>
<td>34</td>
<td>06</td>
<td>85%</td>
</tr>
<tr>
<td>LLE + KNN</td>
<td>36</td>
<td>04</td>
<td>90%</td>
</tr>
</tbody>
</table>

3.3.3 The results of the PNN classifier

Table 3: Analysis of performance parameter by using PNN classifier

<table>
<thead>
<tr>
<th>Method Used</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Dimension Reduction Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA + SVM</td>
<td>30</td>
<td>10</td>
<td>75%</td>
</tr>
<tr>
<td>LLE + SVM</td>
<td>32</td>
<td>08</td>
<td>80%</td>
</tr>
</tbody>
</table>

3.4 Overall comparison result

Fig. 8: Classification results of SVM, KNN, and PNN classifiers over PCA and LLE.

Fig. 8 describes the performance analysis of the three classifiers using PCA and LLE dimension reduction techniques. X-axis of the above figure represents the methods of dimension reduction and Y-axis represents the percentage of classification rates of the three classifiers, showing that LLE gives better results than PCA in case of all classifiers.
4. CONCLUSIONS

Here, two different methods of dimension reduction for face recognition have been discussed along with the three classifiers for analysis of their performance. The need of face recognition has motivated us to do this piece of research work by improving the performance of the different dimension reduction techniques. In this work, eigen values and local linear embedding is utilized for feature extraction purpose. The performance of two types of reduction techniques namely Principal Component Analysis (PCA) and Local Linear Embedding (LLE) are tested for different classifiers and their performance parameters are calculated. The performance of three types of classifiers, i.e., K-nearest neighbor (KNN), probabilistic neural networks (PNN) and support vector machine (SVM) is tested for effective classification of faces.

These different classifiers SVM, KNN and PNN are used to obtain the recognition rate. Discussion is done on the performance of these classifiers. The problems that arise with SVM used as a classifier are: as the size of the pattern increases, the training time increases and also the computational complexity increases for SVM. In order to overcome the drawback of SVM, neural networks classifiers i.e. KNN and PNN are developed. The classification results obtained from SVM, KNN and PNN for PCA and LLE are 70%, 80%, 85% and 80%, 83% and 90% respectively. From results of classifiers comparative analysis show that LLE out performs over all others types of classifiers used in this analysis.

This study may be extended for enlarging the database to check the robustness of the system and using the Bhattacharya distance over Euclidean distance for analyzing the performance of both methods.

REFERENCES


