
Volume 4, Issue 1, February 2010
ISSN 1574-0137
ELSEVIER
Computer Science Review
Editors-in-Chief:
Josep Díaz
Jaroslav Nešetřil

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cosrev

ELSEVIER

Computer Science Review

Survey

Subspace methods for face recognition

Ashok Rao[a], S. Noushath[b,*]

[a] *Department of Electronics and Communication, C.I.T, Gubbi, Tumkur – 572216, India*
[b] *Department of Information Technology, College of Applied Sciences – Sohar, P.O. Box 135, Postal Code 311, Ministry of Higher Education, Oman*

ARTICLE INFO

Article history:
Received 3 February 2008
Received in revised form
11 September 2009
Accepted 23 November 2009

ABSTRACT

Studying the inherently high-dimensional nature of the data in a lower dimensional manifold has become common in recent years. This is generally known as dimensionality reduction. A very interesting strategy for dimensionality reduction is what is known as *subspace analysis*. Beginning with the Eigenface method, face recognition and in general computer vision has witnessed a growing interest in algorithms that capitalize on this idea and an ample number of such efficient algorithms have been proposed. These algorithms mainly differ in the kind of projection method used (linear or non-linear) or in the criterion employed for classification. The objective of this paper is to provide a comprehensive performance evaluation of about twenty five different subspace algorithms under several important real time test conditions. For this purpose, we have considered the performance of these algorithms on data taken from four standard face and object databases namely ORL, Yale, FERET and the COIL-20 object database. This paper also presents some theoretical aspects of the algorithm and the analysis of the simulations carried out.

1. Introduction

Face recognition is a visual pattern recognition problem which has received tremendous attention by researchers in the current computer vision area for its potential applications and inherent technical challenges [1–3]. It is a task that humans perform effortlessly in their daily lives. The human brain perceives a person's face quickly and uses that knowledge to recognize them under highly unpredictable, uncertain and not normal circumstances. To make a machine perform this task with an equal capability as that of a human is still a herculean task [4]. The wide availability of low-cost and powerful computers, embedded computing systems and sensors has created an enormous interest in automatic face processing tasks.

Face recognition research is not only motivated by the fundamental challenges it poses, but also due to the potential application it offers to the society ranging from human computer interaction to authentication and surveillance. Although encouraging progress in face recognition has been realized, a solitary efficient algorithm which can cope up with all kinds of uncertainty in data, model, environment, etc. is yet to be developed. Face recognition evaluation reports and other independent studies indicate that the performance of many state-of-the-art face recognition algorithms deteriorates for various reasons. In particular for unconstrained tasks where viewpoint, illumination, occlusion, expression and other accessories vary considerably, the algorithm performances are generally inconsistent and in many cases very poor [3].

* *Corresponding author.*
E-mail addresses: ashokrao.mys@gmail.com (Ashok Rao), nawali_naushad@yahoo.co.in, noushath.soh@cas.edu.om (S. Noushath).

1574-0137/$ - see front matter
doi:10.1016/j.cosrev.2009.11.003

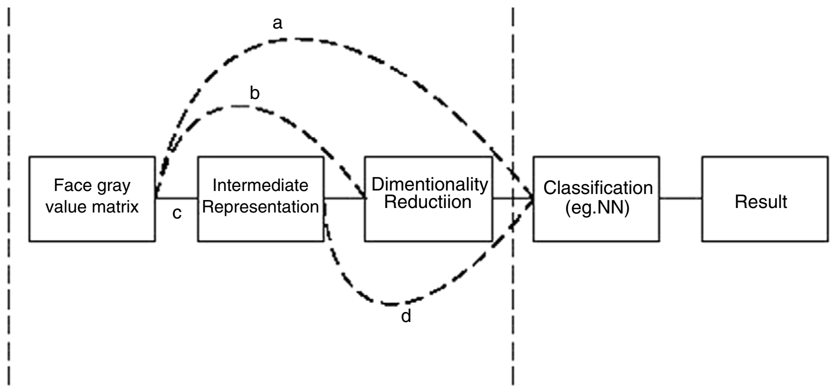

Fig. 1 – Block diagram of a typical face classification approaches (courtesy [3]).

Among various methods proposed to carry out this task, subspace methods are considered to be the most popular and powerful tool for two simple reasons: (i) a strong mathematical model to illustrate the underlying data model is available (ii) it can withstand the variations in image (such as noise, occlusion etc.) unlike the metric for facial features based approaches, which perform poorly even under moderate variations such as illumination or facial expressions [3].

In this work, we have implemented about 25 different subspace algorithms and presented their results under certain testing conditions such as varied facial expression, lighting configurations, *single-sample-per-person* problem and different types of noise contamination. All these experiments were conducted on three standard face databases: ORL, Yale and FERET. We have also attempted to test face recognition algorithms for object recognition using the COIL-20 object database. To do these we have been motivated by following:

(1) From the outcome of the results, we expect some understanding of the fundamental capabilities and limitations of the current face recognition methods, which is crucial in developing successful face recognition methods.
(2) Carrying out independent studies is relevant, since comparisons are normally performed using the implementations of the research group that have proposed the methods. These have shown very subjective claims which needs a more objective analysis.
(3) Testing the performance under noise conditions is not being followed widely, while ascertaining the inherent capabilities of an algorithm. This test condition plays a vital role in identifying an algorithm as truly robust. This in turn helps the algorithm to be adopted for many real time applications.

The rest of the paper is organized as follows: Section 2 presents a brief review of subspace methods. In Section 3, extended versions of popular subspace methods such as PCA, FLD, LPP are described. In Section 4, experimental results and comparative study of various algorithms, starting from the popular Eigenface method to the more recently proposed ones, are carried out. Finally, we close the discussion with concluding remarks in Section 5.

2. Brief overview of subspace methods

In the context of face recognition, the objective of subspace analysis is to find the basis vectors that optimally cluster the projected data according to their class labels. In simple words, a subspace is a subset of a larger space, which contains the properties of the larger space.

Generally, the face images data are high dimensional in nature, typically $\Re^M$, where M is very large. This leads to the problem of the so called *curse-of-dimensionality* and several other related troubles. Especially in the context of similarity and matching based recognition, it is computationally intensive. Instead, if we can represent a set of face images data in a subspace $\Re^k$, where $k \ll M$, the computational burden can be reduced significantly. In addition, for a data of high-dimensional problems, such as face recognition, the entire vector space may cause some processing difficulties, proving futile in many cases. So we look for some suitable subspace which inherits most or all of the properties of the entire vector (training) space. Using these (techniques)subspaces a face image can efficiently be represented as a feature vector of low dimension. The features in such a subspace provide more salient and richer information for recognition than the raw image. It is this success which has made face recognition based on subspace analysis very attractive. Although they have been investigated for decades, a single powerful approach is still yet to be identified. This is because of two reasons: (1) one of size (large dimension) and (2) subjectiveness of the image data. Fig. 1 shows the possible ways of classifying a face image, where all the routes except *route-a* represents a subspace approach [3].

Some of the popular subspace methods are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA) and Locality Preserving Projections (LPP) which are discussed in detail in subsequent sections. Most of the approaches discussed are variants of these four popular subspace techniques.

2.1. *Principal Component Analysis (PCA)*

The objective of PCA is to project the input data to a lower dimensional manifold known as, *principal subspace*, such that it maximizes the variance of the data [5]. More precisely, given a M-dimensional vector representing face samples, the PCA finds the subspace whose basis vectors optimally represent the data with a maximum variance.

Let $\mathbf{W}$ represent a linear transformation matrix which maps the data point from $\Re^M$ to $\Re^k$, where $k \ll M$, as follows:

$$y_i = W^T x_i \quad \forall i = 1, \ldots, N \tag{1}$$

where N is the number of training samples, $\mathbf{W}$ holds Eigen vectors e_i, obtained through the Eigen decomposition $Qe_i = \lambda_i e_i$, $Q = XX^T$ represents the covariance matrix and λ_i's are the eigenvalues associated with Eigen vectors e_i. The x_i's are input data in $\Re^M$ and y_i's are PCA projected features in $\Re^k$.

2.2. *Linear Discriminant Analysis (LDA)*

The objective of LDA is to find the basis vectors which best *discriminates* the classes, by maximizing the between-class measure S_b, while minimizing the within-class matrix S_w [6,7]. This is unlike PCA which best *describes* the data. The S_b and S_w are defined as:

$$S_w = \sum_{i=1}^{C} \sum_{x_j \in C_i} (x_j - \bar{m}_i)(x_j - \bar{m}_i)^T \tag{2}$$

$$S_b = \sum_{i=1}^{C} n_i (\bar{m}_i - \bar{m})(\bar{m}_i - \bar{m})^T \tag{3}$$

where $\bar{m}_i$ is the ith class mean and $\bar{m}$ is the global mean and n_i is the number of samples in ith class. The LDA subspace is spanned by set of vectors, W satisfying the following:

$$W = \arg_W \max \frac{\left|W^T S_b W\right|}{\left|W^T S_w W\right|}. \tag{4}$$

The $\mathbf{W}$ is composed of eigenvectors corresponding to k largest eigenvectors of matrix: $S_w^{-1} S_b$.

2.3. *The Locality Preserving Projections (LPP)*

The LPP [8] is considered as an alternative to the PCA method. The main objective of LPP is to preserve the local structure of the input vector space by explicitly considering the manifold structure. Since it preserves the neighborhood information, its classification performance is much better than other subspace approaches like PCA [5] and FLD [6]. Here, we briefly outline the LPP model [8]. Let there be N number of input data points $(x_1, x_2, \ldots, x_N)$, which are in $\Re^M$. The first step of this algorithm is to construct the adjacency graph $\mathbf{G}$ of N nodes, such that nodes i and j are linked if x_i and x_j are *close* with respect to each other in either of the following two conditions:

(1) K-nearest neighbors: Nodes i and j are linked by an edge, if i is among K-nearest neighbors of j or vice-versa.
(2) ϵ-neighbors: Nodes i and j are linked by an edge if $\|x_i - x_j\|^2 < \epsilon$, where $\|\cdot\|$ is the usual Euclidean norm.

The next step is to construct the weight matrix $\mathbf{Wt}$, which is a sparse symmetric $N \times N$ matrix with weights Wt_{ij} if there is an edge between nodes i and j, and 0 if there is no edge. Two alternative criterion to construct the weight matrix:

(1) Heat-Kernel: $Wt_{ij} = e^{\frac{-\|x_i - x_j\|^2}{t}}$, if i and j are linked.
(2) $Wt_{ij} = 1$, iff nodes i and j are linked by an edge.

The objective function of the LPP model is to solve the following generalized Eigenvalue–Eigenvector problem:

$$XLX^T a = \lambda XDX^T a \tag{5}$$

where $\mathbf{D}$ is the diagonal matrix with entries as $D_{ii} = \sum_j w_{ji}$ and $L = D - W$ is the Laplacian matrix.

The transformation matrix $\mathbf{W}$ is formed by arranging the eigenvectors of Eq. (5) ordered according to their eigenvalues, $\lambda_1 > \lambda_2, \ldots, > \lambda_k$. Thus, the feature vector y_i of input x_i is obtained as follows:

$$x_i \to y_i = W^T x_i \quad \forall i = 1, 2, \ldots, N. \tag{6}$$

Note: The XDX^T matrix is always singular because of the high-dimensional nature of the image space. To alleviate this problem, PCA is used as the preprocessing step to reduce the dimensionality of the input vector space.

2.4. *Independent Component Analysis (ICA)*

The concept of Independent Component Analysis (ICA) has been explored for face recognition [9]. The ICA is a generalization of the PCA algorithm which also extracts the information contained in the higher-order relationships among pixels. It differs from PCA in the following three aspects: (i) The basis vectors obtained through ICA need not be orthogonal in nature, (ii) The ICA can also handle data which are non-Gaussian in nature and (iii) the ICA minimizes the higher-order dependencies, unlike the PCA which minimizes the second order(moments) dependencies of the data. The ICA was investigated with two fundamentally different architectures: Architecture-1 (ICA-I) and Architecture-2 (ICA-II). ICA-I reflect more local properties of the faces (data) while the ICA-II reflect more global properties and thus more closely captures the face information. Readers are advised to go through the Ref. [9] for further details about the ICA architectures.

In general, the following are the main steps involved in any subspace based algorithm for face recognition.

Subspace Based Algorithm for Face Recognition

Input:

- A set of N training samples belonging to C classes.
- k, number of projection vectors to be used
- A test image T.

Output:

- Feature Extraction of training samples: (Training Phase)
- Class label/identity of the test sample T.

Steps:

(1) Acquire the training samples A_i $(i = 1, \ldots, N)$.
(2) Compute the basis vectors W_k using any subspace algorithms (for eg: PCA, LDA, LPP, ICA, etc).
(3) Project the training samples A_i $(i = 1, \ldots, N)$ on to W to extract features: $Feature\_Training_i = A_i * W$ $(i = 1, \ldots, N)$
(4) Project the test sample T onto W: $Feature\_Testing = T * W$.
(5) Compute the identity by finding the nearest match between $Feature\_Testing$ with $Feature\_Training_i$ $(i = 1, \ldots, N)$.

Inputs to any subspace algorithm are, in general, a set of training samples and a scalar (k) to indicate how many basis vectors (or points) have to be used for recognition purposes. Often, this value for k is fixed based on empirical study since it is very subjective in nature. Usually the experiment is repeated for some fixed number of times by varying the number of basis vectors k (for e.g.: by setting $k = 1, 2, 3, \ldots, 20, 25, 30, 35, 40, 45$) and a value for k that corresponds to the best classification result on the image set would be chosen.

In step-1 shown above, all training samples are acquired for training purposes, which may have to undergo basic pre-processing steps such as illumination normalization, cropping to an appropriate size etc. In step-2, the basis vectors W (which are also known as basis images/projection vectors/reduced space/transformation matrix) are computed using any subspace algorithm. Once the basis vectors are computed, in step-3 they are projected onto each training sample to extract features and are stored in an appropriate data structure. Similarly, features are extracted for test samples in step-4. It has to be noted that the basis vectors W are post multiplied in step-3 and step-4, however it can also be pre-multiplied based on the type of the subspace algorithm used. Finally, a nearest neighbor classifier based on the Euclidean distance can be used to compute the identity of the test sample.

3. Some extended version of subspace methods—A brief overview

The above said algorithms are the state-of-the-art subspace methods proposed for face recognition [10]. Many variants of these algorithms are devised to overcome specific anomalies such as storage burden, computational complexity and the Single Sample Per Person (SSPP) problem etc. Especially under the latter circumstance (i.e., SSPP), the performance of many algorithms deteriorate due to the inadequacy of representative training samples. Fortunately, several ways to get around this difficulty have been proposed [11–15]. A couple of attempts were also made to make the Fisherface method compatible under the SSPP problem [16–18]. In addition, many extended algorithms have been proposed based on the standard Eigenface method during the last several years, such as the probabilistic Eigenface method [19], Support Vector Machines (SVM) method [20], feature-line method [21], evolutionary pursuit [22] etc. The correlation techniques for improving the classification performance and also for reducing the computational complexity of the subspace based face recognition have also been presented [23].

To combine the merits of PCA, LDA and Bayesian subspace approaches a single framework model was presented in [24]. Here a 3D parameter space was constructed using the three subspace dimensions as axes. A much improved accuracy was achieved through this unified subspace approach which does not suffer from the individual limitations. To enhance the performance of PCA and LDA methods, instead of extracting single set of features, more than one set of features were extracted by using Gaussian Mixture Models [25,26,10]. In the similar lines, the LPP method was also extended by extracting more than one set of features using Gaussian Mixture Models [27,28]. These models, though computationally expensive, obtained a good performance under large variations of pose and/or illumination. Further, in order to utilize the contribution made by the local parts of the whole image, the PCA and FLD methods are applied component-wise on the whole image [29,30]. This has resulted in an improved accuracy under varied facial expression and lighting configurations and justifies increased computational efforts.

Several attempts have been made to reduce the computational burden of the original PCA and FLD methods [31,32]. In [31], the original PCA and LDA methods were implemented in the DCT domain and it has been theoretically proved that the results obtained in the DCT domain are exactly the same as the one obtained in the time/spatial domain. Similarly, the PCA and FLD algorithms have been implemented in the wavelet domain [32]. The merit of transforming the original image from the time domain to frequency domain is that, in frequency domain we will be working on the reduced coefficient set which are more prominent for recognition task and it also helps to reduce the computational burden.

The standard PCA and FLD methods, though popular and widely admired, are computationally expensive, because, the 2D image has to be aligned into 1D vectors prior to feature extraction. This leads to a large covariance matrix and consequently the Eigen value decomposition task becomes tedious and time consuming. To resolve this, 2DPCA [33] and 2DFLD [34] were proposed which does not involve the matrix to vector conversion.[1] Because of this, several appreciable merits were achieved: (i) Computation of the scatter matrix (or scatter matrices in case of Fisherface method) became simple (ii) Consequently, the computation time was also reduced, and (iii) significant improvement in accuracy was achieved as compared with their 1D counterparts. However, the main drawback is that these methods require a huge feature matrix for representation and recognition tasks. Later, even this drawback of 2DPCA and 2DFLD was resolved by $2D^2$PCA [36–38] and $2D^2$LDA [39] methods respectively. These methods are unlike their 2D counterparts [33,34] which used either the row or the column projections at a time. The recognition accuracy of these methods was either the same or even higher than the corresponding 2D versions, but their computation time and storage requirements were significantly reduced compared with the 2DPCA and 2DFLD methods.

More recently, methods namely DiaPCA [40] and DiaFLD [41] have been proposed. These methods seek optimal projection vectors from diagonal face images so that the correlation between variations of both rows and columns of images can be preserved [40]. A more substantial improvement in accuracy was achieved by the DiaPCA and DiaFLD methods than the 2DPCA and 2DFLD methods respectively. Subsequently, these DiaPCA and DiaFLD methods were combined

[1] Refs. [35,34] are two similar methods found in the literature referred respectively as 2DFLD and 2DLDA. In all our discussions, whenever we refer to Ref. [34], it also implies Ref. [35] and vice-versa.

with 2DPCA and 2DFLD methods respectively, in order to achieve efficiency both in terms of accuracy and storage requirements.

Very recently, a significant improvement to subspace method has been proposed by taking a tensor approach to image data [42,43]. Two crucial aspects of this has been rapid convergence (between 10 and 30 iterations) and as is keenly sought after in optimization techniques, the ability to get over local minimas. These results have taken the standard supervised learning approaches of Support Vector Machines, minimax probability machines, FLD and extended these into tensored structures (multi linear forms), with improved results. In terms of real time applications, while the rapid convergence shown in these papers [42,44] is encouraging, what is not clear is the computational complexity involved. This is because the authors have been using optimization methods based on ideas of alternate projections, which is known to be computationally burdensome.

The aforementioned algorithms are modified versions of their original time/spatial domain PCA and FLD methods. All these methods are modified or extended to overcome a specific anomaly (such as storage burden, computational complexity, small sample size (SSS) problem, SSPP problem etc.) that prevailed in their earlier version. This implies that all these methods are proposed and tuned to overcome only a particular issue while ignoring their performance under other common situations like variations in lighting, occlusions and different noise conditions etc. One of the objectives of this work is to study the behavior of all these algorithms under a common testing framework, apart from testing them in their dedicated test domain for which they were originally proposed. This helps us to more objectively capture the relative strength or limitations exhibited by these algorithms under entirely different experimental scenarios. This will yield a more generalized result which is the need of the hour as we have too many specific/conditional results in literature [28].

4. Experimental results and comparative analysis

We have conducted extensive experiments in order to observe the behavior of the algorithms and to study their theoretical claims under different experimental scenarios. We have considered four standard databases for this purpose, among which three are face databases and one is an object database. In all the experiments, the nearest neighbor classifier (Euclidean distance) is used for classification purposes.[2] The number of projection vectors (k) has a considerable impact on the classification performance of the algorithm. Hence the value of k for all the algorithms is controlled by θ as in the following equation:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq \theta \qquad (7)$$

where λ_i are Eigen values such that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \cdots \geq \lambda_N$, N is the total number of sorted Eigen values corresponding to N distinct images and θ is a preset threshold. In most of our experiments, θ is set to 0.98 to select the value of k. We call this the *confidence interval* throughout the paper.[3] In a few cases, the value of k is empirically fixed to some number which corresponds to best classification result on the image dataset. Finally, all our experiments were carried out on a P4 3 GHz CPU with 1 GB RAM memory on the Matlab 7 platform.

Table 1 – Comparing PCA based methods for five training samples (98% confidence interval).

Methods	Accuracy (%)	Dimension	Time (s)
PCA [5]	94.00	187	17.89
PC^{2}A [11]	94.50	187	19.87
PCA + DCT [31]	94.00	182	14.56
PCA + Wavelet [32]	95.75	185	13.38
Enhanced PC^{2}A [12]	95.75	180	21.34
SPCA [13]	93.75	187	18.87
2DPCA [33]	93.75	112 × 48	7.43
Alternative 2DPCA [36]	94.75	42 × 92	8.03
2D^{2}PCA [36]	87.25	42 × 48	4.59
DiaPCA [40]	93.00	112 × 58	9.76
SpPCA [29]	94.50	40 × 92	33.74
ICA-I [9]	95.50	187	28.87
ICA-II [9]	95.00	187	31.23
LPP [8]	95.00	187	24.54
PCA mixture [25]	94.50	167	78.77

4.1. *Databases and their experimental setup*

The dataset considered in our study are ORL,[4] the Yale [6], the FERET [45], and the COIL-20 object database.[5]

The ORL database contains 400 images corresponding to 40 different persons with each person having ten different views. The variations include changes in expression, facial details and slight variations in pose. This database is used to evaluate the performance under the condition when both sample size and pose are varied. Besides these experiments on ORL, we have also conducted rigorous experiments under different noisy conditions modeled by five different distributions: Uniform (for Salt-and-Pepper behavior), Gaussian, Weibull, Exponential and Beta. Refer to Appendix A for details about noise modeling.

[2] Since ICA is not a unitary transformation, Euclidean distance is not preserved. Hence we used a cosine similarity measure while implementing ICA architectures. However, for other algorithms, their performances are similar, irrespective of these two distance measures.

[3] This should not be confused with the one used in statistics, where a confidence interval is defined as the range of values computed in such a way that it contains the estimated parameter a high proportion. For example, the 98% confidence interval is constructed so that 98% of such intervals will contain the parameter.

[4] www.uk.research.att.com/facedatabase.html.

[5] www1.cs.columbia.edu/CAVE/research/softlib/coil-20.html.

The Yale database contains 165 images of 15 persons, each person having 11 different views. This database was considered to study the performance of the algorithms under varied facial expression and lighting configurations. We have adopted the *Leave-one-out* strategy on this database and analysed the classification performance of all methods by varying the confidence intervals.

We have selected a subset of the FERET database. This partial FERET database [11,12,17] was used to examine the performance of the methods when only one image is available for training. This subset of the FERET database contains 400 gray level frontal view face images of 200 persons, each of which is cropped to a size of 60 × 60. There are 71 females and 129 males: each person has two images (**fa** and **fb**) with different facial expressions. The **fa** images are used as a gallery for training while the **fb** images are used as probes for testing.

Finally, an experiment on the well known COIL-20 object database was performed to check the efficacy of the face recognition algorithms also to be used for object recognition [46]. This database contains 1440 gray scale image of 20 objects. The objects were placed on a motorized turntable against a black background. The turn table was rotated through 360° to vary object pose with respect to fixed camera. Images of the objects were taken at a pose interval of 5°, which correspond to 72 images per object.

4.2. *Results on the ORL database*

First, an experiment was performed using the first five samples per class for training and the remaining images were used for testing. So effectively, 200 images were constituted in both training and testing sets. Table 1 gives the comparison of PCA based methods on recognition accuracy,[6] corresponding dimension of the feature set and running time costs for 98% confidence interval. Similarly, Table 3 presents the results of FLD based methods.

It can be seen from Table 1 that, among different PCA variant methods proposed, the one that is implemented in the wavelet domain (PCA + Wavelet) [32] and the DCT domain (PCA + DCT) [31] outperformed the other methods in terms of accuracy and also in terms of storage requirements. This is because in wavelet and DCT domains, the PCA method is performed on the reduced coefficient set which contains most of the discriminatory information. This, in turn, helps to reduce the computational burden involved while building the scatter matrix and hence the reduced running time costs. On the other hand, the PC^{2}A method [11] claimed to have a 3%–5% improvement in the accuracy over the PCA when only one sample per person is available for training. However, under the scenarios where there are multiple numbers of training samples, its classification performance is comparable with that of the PCA method. Even the running time cost of PC^{2}A method is more because of the extra preprocessing step involved in that method. The subpattern based PCA method (SpPCA) [29] obtained a marginal improvement over the conventional PCA method, but its demerit is the dimension of the feature set, which is comparatively higher than the PCA method. In addition, it is also expensive in terms of running time costs because it involves the subpattern formation and then applying PCA on every subpattern training set. Nevertheless, the optimal number of subpatterns to yield the best recognition accuracy is still not clear. The number of subpatterns is empercially fixed which is again a cumbersome task and also subjective in most cases.

The recognition accuracy obtained by two-dimensional variants of PCA method, i.e., 2DPCA [33], Alternative-2DPCA [36] and DiaPCA [40] are comparable with that of the PCA method. Although the 2DPCA and DiaPCA methods are popular and proven to be efficient, their performance is even poorer than the conventional PCA method, because these methods obtained best recognition accuracy when only fewer number of principal components were used (Refer Table 2) [33,40]. Because we fixed the confidence interval to 98%, more number of principal components were chosen which made the feature set redundant and thus the accuracy was reduced significantly. The same reasoning holds good for the poor performance of 2D^{2}PCA [36] method.

From the table, we can observe that the classification performance of many algorithms are greatly impacted by the number of projection vectors (k) used. Therefore, we conducted one more experiment making full use of the available data, where each algorithm is repeated 40 times by varying both the number of projection vectors k (k is varied from 1 to 40) and training sample. We chose the accuracy that corresponds to best classification performance on the dataset. This result is shown in Table 2, where the values in the parenthesis denote the dimension of feature vector/matrix for which the highest recognition accuracy was obtained. It can be ascertained from Table 2 that the PCA method was outperformed by all other methods in most of the cases, since these algorithms are repeated several times by varying the number of projection vectors and the one which corresponds to the best classification performance was chosen. However, this situation may deprive the algorithm from becoming a candidate for real time commercial systems. This kind of experimentation however has been widely adopted in many research groups to test their algorithms [36,41].

Now we present a similar kind of results for FLD and its variant methods (Refer Table 3) for 98% confidence intervals. It can be observed from Table 3 that the orthogonalized FLD method [47] obtained a significantly improved classification performance over any other method. This is because the Fisher projection vectors obtained through the Eigen decomposition of $S_w^{-1}S_b$ may not relate to distinct Eigen values. This in turn causes non-orthogonality among the FLD projection vectors. As is well known in many practices of linear algebra, establishing the orthogonality among the vectors (that are linearly independent) can greatly improve the performance of the system [48]. The orthogonalized FLD method [47], as the name implies, is the orthogonalized version of the conventional FLD method [6] and thus contains the orthogonal projection vectors. It can also be noticed that the performance of 2D^{2}LDA and DiaFLD + 2DFLD methods

[6] Recognition accuracy is defined as the ratio of correctly classified images to the total test samples.

Table 2 – Best recognition accuracy of PCA based methods for varying number of training samples and principal components.

Methods	Number of training samples				
	2	4	5	6	8
PCA	$86.00^{(36)}$	$93.75^{(37)}$	$95.25^{(27)}$	$98.00^{(13)}$	$99.00^{(9)}$
PC^2A	$88.25^{(34)}$	$94.75^{(39)}$	$96.25^{(3)}$	$98.00^{(31)}$	$99.00^{(9)}$
PCA + DCT	$87.00^{(37)}$	$93.75^{(37)}$	$96.00^{(29)}$	$98.00^{(13)}$	$99.00^{(12)}$
PCA + Wavelet	$85.75^{(28)}$	$93.00^{(36)}$	$95.50^{(36)}$	$97.50^{(33)}$	$99.00^{(25)}$
Enhanced PC^2A	$83.75^{(37)}$	$92.5^{(36)}$	$95.5^{(38)}$	$98.00^{(37)}$	$99.25^{(32)}$
SPCA	$86.25^{(29)}$	$94.5^{(38)}$	$95.75^{(27)}$	$98.00^{(32)}$	$99.25^{(11)}$
2DPCA	$90.25^{(112\times4)}$	$95.00^{(112\times4)}$	$97.50^{(112\times9)}$	$98.75^{(112\times3)}$	$99.25^{(112\times3)}$
Alternative 2DPCA	$89.25^{(92\times13)}$	$95.25^{(92\times13)}$	$97.50^{(92\times13)}$	$98.50^{(92\times8)}$	$99.50^{(3\times92)}$
$2D^2PCA$	$88.75^{(7\times7)}$	$95.00^{(7\times7)}$	$96.75^{(7\times7)}$	$98.50^{(7\times7)}$	$99.50^{(5\times5)}$
DiaPCA	$90.25^{(112\times6)}$	$95.75^{(112\times8)}$	$98.50^{(112\times8)}$	$99.00^{(112\times7)}$	$99.25^{(112\times3)}$
SpPCA	$81.00^{(10)}$	$90.25^{(9)}$	$94.25^{(35)}$	$96.75^{(12)}$	$99.00^{(27)}$
ICA-I	$88.25^{(36)}$	$94.50^{(36)}$	$97.25^{(37)}$	$98.25^{(38)}$	$99.00^{(07)}$
ICA-II	$88.00^{(34)}$	$94.75^{(19)}$	$97.00^{(25)}$	$98.00^{(27)}$	$99.00^{(30)}$
LPP	$88.50^{(23)}$	$94.75^{(18)}$	$96.00^{(38)}$	$98.50^{(18)}$	$99.00^{(31)}$
PCA mixture	$86.50^{(24)}$	$94.00^{(39)}$	$95.75^{(28)}$	$98.00^{(23)}$	$98.00^{(25)}$

Table 3 – Comparing FLD based methods for five training samples (98% confidence interval).

Methods	Accuracy	Dimension	Time (s)
FLD [6]	89.50	187	15.61
Orthogonalized FLD [47]	96.00	187	17.41
FLD + DCT [31]	90.75	166	13.39
FLD + wavelet [32]	90.00	185	12.19
2DFLD [34]	86.75	112 × 89	10.10
Alternative 2DFLD [39]	90.00	108 × 92	9.80
$2D^2LDA$ [39]	60.00	108 × 89	5.1
DiaFLD [41]	89.00	112 × 89	11.35
DiaFLD + 2DFLD [41]	59.25	49 × 89	4.75
dcFLD [30]	90.75	37 × 92	41.33
LDA mixture [26]	91.00	149	92.44

worsened as the number of projection vectors increased. This is due to the fact that there will be high correlation and redundancy among the Fisher discriminant vectors that get introduced when more number of projection vectors are used.

For reasons due to orthogonality, it is not surprising to note that the Orthogonalized FLD method also obtained good classification performance in another experiment where both sample size and number of projection vectors are varied (see Table 4). Under this experiment, the performance of FLD based methods improved.

We would also like to make a point that the performance of other FLD methods would certainly improve if their projection vectors are also made orthogonal. To justify this, we have also conducted an experiment where the projection vectors of other methods are made orthogonal through Gram–Schmidt decomposition process (Refer to Table 5[7]). The results clearly indicate that there is a considerable improvement in classification performance, especially the $(2D)^2$LDA method, compared to the results obtained in their conventional non-orthogonal form. Hence, we can say that for a similar number of projection vectors, the orthogonalized versions always achieve a better accuracy than their conventional forms. Nevertheless, it is worth mentioning that the orthogonalized version is more computationally intensive than its corresponding non-orthogonalized version because the former approach is backed with an extra Gram–Schmidt decomposition preprocessing step. Note that we have not explicitly mentioned the word *orthogonal* for PCA based approaches as they are inherently orthogonal in nature. However, for a fair comparison, we only present the result of these FLD methods in their original non-orthogonal form.

An experiment was conducted to check the efficiency of PCA, 2DPCA, $2D^2$PCA and SpPCA methods in representing face images using the same number of principal components. These algorithms are chosen since they differ mainly in the projection technique employed in each algorithm. Taking one image from the ORL database as an example, we determine its five reconstructed images for a varying number of projection vectors k (where k is varied from 10 to 50 insteps of 10). From Fig. 2, it is clear that the $2D^2$PCA method represents facial images as good or comparable with that of the 2DPCA method. Note that the former method uses greatly reduced coefficients for representation than the latter method. The bottom two rows correspond to the reconstructed images of the SpPCA method [29]. It can be observed from the figure that as the size of the subpattern increases, its reconstruction quality decreases.

4.3. *Performance of algorithms under noise conditions*

In signal or image processing applications, there is a strong urge for the development of algorithms which can withstand any conceivable noise that can arise within the practical framework of the signal. Many successful algorithms have been proposed in the literature that work well under clean or noise free conditions. However, many of them cannot successfully deal with noisy data and thus their performance drops significantly. Consequently, the performance of algorithms under noise conditions becomes crucial to check the efficacy of algorithms in real time or practical pattern recognition and computer vision tasks [49]. The details of noise modeling and the composition of a noisy

[7] The prefix *ortho* in Table 5 and other related tables indicates the orthogonalized version of the corresponding FLD algorithm.

Table 4 – Best recognition accuracy of FLD based methods for varying number of training samples and principal components.

Methods	Number of training samples				
	2	4	5	6	8
FLD	$85.75^{(40)}$	$94.50^{(29)}$	$96.25^{(31)}$	$97.75^{(10)}$	$99.25^{(13)}$
Orthogonalized FLD	$89.25^{(31)}$	$96.50^{(22)}$	$97.00^{(39)}$	$98.50^{(20)}$	$99.75^{(14)}$
FLD + DCT	$87.25^{(30)}$	$94.00^{(29)}$	$96.00^{(35)}$	$98.25^{(10)}$	$99.25^{(13)}$
FLD + Wavelet	$87.00^{(28)}$	$94.50^{(31)}$	$96.25^{(35)}$	$97.75^{(29)}$	$99.50^{(21)}$
2DFLD	$91.25^{(112\times2)}$	$96.50^{(112\times5)}$	$98.00^{(112\times7)}$	$99.25^{(112\times96)}$	$99.50^{(112\times7)}$
Alternative 2DFLD	$88.75^{(92\times8)}$	$95.25^{(92\times8)}$	$97.00^{(92\times9)}$	$98.50^{(92\times10)}$	$99.75^{(92\times3)}$
2D^{2}LDA	$89.75^{(8\times8)}$	$96.25^{(11\times11)}$	$97.50^{(8\times8)}$	$98.75^{(9\times9)}$	$99.75^{(5\times5)}$
DiaFLD	$89.75^{(112\times8)}$	$95.25^{(112\times4)}$	$97.25^{(112\times8)}$	$98.50^{(112\times4)}$	$99.25^{(112\times5)}$
DiaFLD + 2DFLD	$89.25^{(7\times7)}$	$95.75^{(7\times7)}$	$98.00^{(7\times7)}$	$98.75^{(7\times7)}$	$99.75^{(4\times4)}$
dcFLD	$87.00^{(12\times92)}$	$94.75^{(18\times92)}$	$96.50^{(13\times92)}$	$98.50^{(12\times92)}$	$100.00^{(6\times92)}$
LDA mixture	$88.00^{(38)}$	$96.00^{(38)}$	$96.25^{(32)}$	$98.00^{(30)}$	$99.50^{(28)}$

Table 5 – Comparing results of FLD based methods under clean conditions: Conventional vs. Orthogonal (for 98% confidence interval).

Methods	Number of training samples				
	2	4	5	6	8
FLD	85.00	87.00	89.50	91.25	95.25
Ortho-FLD	89.25	94.00	96.00	97.50	99.00
FLD + DCT	86.00	90.50	90.75	92.75	97.50
Ortho-FLD + DCT	91.00	96.00	97.00	98.75	99.00
FLD + Wavelet	85.75	89.00	90.00	89.25	95.25
Ortho-FLD + Wavelet	88.50	94.25	95.00	97.75	99.00
2DFLD	70.50	82.50	86.75	91.75	98.00
Ortho-2DFLD	87.25	93.50	95.50	96.75	99.00
A2DFLD	70.25	85.00	90.00	93.50	98.25
Ortho-A2DFLD	88.00	94.75	97.00	97.50	99.25
2D^{2}LDA	27.25	51.00	60.00	68.75	85.75
Ortho-2D^{2}LDA	86.00	93.25	95.50	96.75	99.00
DiaFLD	76.75	86.75	89.00	93.00	97.50
Ortho-DiaFLD	87.25	93.50	95.50	97.00	99.00
DiaFLD + 2DFLD	27.00	49.50	59.25	68.25	86.50
Ortho-DiaFLD + 2DFLD	69.75	82.25	85.25	90.75	97.00

test set is given in Appendices A and B respectively. Note that we have used first five clean images from each class of ORL database as the training samples.

Table 6 presents the average recognition accuracy[8] (ARA) obtained by various algorithms under five different noise models. From the table, we can infer many interesting points that are not unusual. With this kind of analysis, we can study the true behavior of algorithms which also help to contemplate best performing algorithms as possible candidates for real time applications.

We have ascertained the following crucial things from this experiment:

(1) The 2D^{2}PCA and LPP are the only two algorithms to exhibit robustness under all five noise models.

(2) 2D^{2}LDA and 2D^{2}PCA methods exhibited more robustness under Weibull and exponential noise conditions when compared to other methods proposed in their respective domain.

(3) The PCA and FLD methods implemented in the DCT domain, i.e., PCA + DCT and FLD + DCT, emerged as the *next* best performing algorithms in this test contest. However their performance under Weibull and exponential noise was poor because the sufficient number of DCT coefficients required for reconstruction has been altered quite significantly by Weibull and exponential noise (note the plot of these two distributions in Fig. A.1). It has also definitely altered the data related to high and/or low frequency coefficients.

Fig. 2 – Some Reconstructed Images (RI) with increasing principal components. First Row: RI by PCA, Second Row: RI by 2DPCA, Third Row: RI by 2D^{2}PCA, Fourth and Fifth Row: RI image by SpPCA method with subpattern dimension 112 and 644 respectively.

[8] Average of ten different experiments conducted by varying the noise density from 0.1 to 1.0 insteps of 0.1.

Table 6 – Average recognition accuracy of algorithms under noise conditions for 98% confidence intervals.

Methods	Different noise conditions				
	Gaussian	Salt-and-Pepper	Exponential	Weibull	Beta
PCA methods					
PCA	98.50	64.75	82.50	73.50	100.00
PCA + DCT	98.50	66.50	76.25	62.25	99.75
PCA + Wavelet	95.00	62.00	94.00	98.00	99.75
PC^2A	97.50	63.50	80.00	67.50	99.75
Enhanced PC^2A	79.00	55.25	36.75	36.25	94.25
SPCA	98.00	63.75	79.50	66.50	100.00
2DPCA	92.00	62.75	64.50	56.50	98.00
Alternative 2DPCA	92.00	61.50	70.25	56.75	99.00
$2D^2PCA$	98.75	67.25	95.50	99.00	99.75
DiaPCA	91.25	61.75	59.50	52.00	97.50
DiaPCA + 2DPCA	97.75	65.50	94.50	98.75	99.75
SpPCA	92.00	65.25	80.00	48.25	98.00
ICA-I	98.50	75.25	31.75	32.00	98.50
ICA-II	98.50	75.00	40.25	54.00	100
LPP	97.00	66.50	95.75	99.25	100
PCA mixture	98.50	68.00	94.00	99.00	99.00
FLD Methods					
FLD	94.75	67.00	73.25	59.75	99.50
Orthogonalized FLD	89.00	61.25	24.25	32.00	90.50
FLD + DCT	95.25	65.25	68.25	54.00	98.75
FLD + Wavelet	89.00	60.75	90.50	92.75	98.50
2DFLD	76.75	54.25	77.50	71.75	77.75
Alternative 2DFLD	87.25	58.25	60.50	48.25	89.75
$2D^2LDA$	73.25	49.25	92.50	96.25	77.00
DiaFLD	87.25	58.75	74.50	66.75	91.25
DiaFLD + 2DFLD	54.25	40.25	87.50	89.25	60.00
dcFLD	90.50	56.75	48.35	80.00	66.50
LDA mixtures	96.00	68.00	78.00	82.00	99.50

(4) PCA and FLD methods implemented in the wavelet domain, i.e., PCA + Wavelet and FLD + Wavelet, remained insensitive and exhibited robustness even under exponential and Weibull noise conditions. This is as expected, since the wavelet has the advantage of scale and space orientation, it is able to represent this noise disturbance in a few scales across space and few space points across scales.

(5) The performances of all PCA based methods under Beta noise conditions were comparable with each other. This Beta noise model has had very little influence on the performance. This issue needs to be addressed in detail, perhaps by varying the parameter values of α and β in its probability distribution function (Ref. Eq. (A.5)).

(6) The behavior of PC^2A, Enhanced PC^2A and SPCA methods, which were specially proposed to handle the SSPP problem, was far behind the performance of the conventional PCA method under noisy conditions.

(7) All methods perform generally poorly under corruption by Salt-and-Pepper noise. This is due to the fact that when we include such severe distortions (like an impulse in gray scale which is equivalent to white noise in frequency domain), the image data undergoes a significant change in behavior. This is closer to data exhibiting characteristics in the non-gaussian/non-stationary behavior. Since the ICA architectures I and II can capture non-gaussian/non-stationary behavior, they tend to yield a better response. However, they yield a very poor response to Weibull and exponential noise contamination.

(8) The orthogonalized FLD method is the most sensitive algorithm to all noise conditions. Its performance was even poorer under exponential and Weibull noise conditions. This is because any noise influence which disturbs the orthogonality to a little extent significantly brings down the performance of this algorithm. The moment orthogonality conditions are disturbed marginally, the algorithm's performances are hit by a large margin, which is quite evident in Table 6. In that case, it will be interesting to see the behavior of matrix based FLD methods in their orthogonal form.

Most of the algorithms performed well under Gaussian and Beta noise, however this was not the case with Salt-and-Pepper, Exponential and Weibull noise data. What is important is to look for those algorithms which perform very well under all types of noise disturbances. Currently, it appears that PCA + GMM appears to be the best among the class of subspace methods under noise conditions. Further endorsement is available even under LDA models. Therefore, it is hardly surprising that mixture models, in particular the GMM, are likely to dominate pattern recognition and statistical signal processing approaches in the future [10,50].

Motivated by the latter observations made in our previous experiment, we conducted one more experiment to study the effect of noises on the FLD methods after they are

Table 7 – Results of FLD based methods under noise conditions: Conventional vs. Orthogonal (90% confidence interval).

Methods	Different noise conditions				
	Gaussian	Salt-and-Pepper	Exponential	Weibull	Beta
FLD	94.00	65.75	78.50	67.25	98.50
Ortho-FLD	86.00	58.75	15.50	29.50	87.75
FLD + DCT	94.00	64.75	74.00	60.00	99.25
Ortho-FLD + DCT	84.75	56.75	13.25	29.25	88.50
FLD + Wavelet	85.75	60.50	92.50	96.50	98.00
Ortho-FLD + Wavelet	73.50	50.75	81.50	71.00	87.25
2DFLD	76.50	52.75	74.75	68.25	76.00
Ortho-2DFLD	83.75	58.75	27.25	33.25	94.75
A2DFLD	88.25	58.50	57.50	48.00	89.00
Ortho-A2DFLD	85.50	58.00	17.75	29.25	93.00
$2D^2$LDA	74.75	50.25	91.75	94.75	74.00
Ortho-$2D^2$LDA	82.75	55.50	42.25	40.50	90.25
DiaFLD	86.75	59.25	71.00	64.75	90.25
Ortho-DiaFLD	89.75	62.25	45.75	43.50	97.75
DiaFLD + 2DFLD	51.25	36.75	77.75	72.25	50.25
Ortho-DiaFLD + 2DFLD	62.25	45.75	41.25	41.50	66.00

Table 8 – Results of FLD based methods under noise conditions: Conventional vs. Orthogonal (95% confidence interval).

Methods	Different noise conditions				
	Gaussian	Salt-and-Pepper	Exponential	Weibull	Beta
FLD	95.25	68.50	76.50	63.00	99.50
Ortho-FLD	86.75	59.75	23.50	31.25	87.75
FLD + DCT	95.50	66.25	75.25	61.75	99.75
Ortho-FLD + DCT	90.25	62.00	19.00	31.00	90.50
FLD + Wavelet	86.00	57.50	91.75	96.25	97.75
Ortho-FLD + Wavelet	74.00	50.00	81.50	70.50	87.00
2DFLD	77.50	54.25	76.75	70.00	76.75
Ortho-2DFLD	85.50	58.75	23.00	31.25	95.75
A2DFLD	88.00	58.25	59.75	48.25	89.75
Ortho-A2DFLD	85.00	59.25	16.00	29.25	94.50
$2D^2$LDA	74.25	49.50	92.25	95.00	75.25
Ortho-$2D^2$LDA	85.50	57.25	38.00	38.50	92.75
DiaFLD	86.75	59.00	72.75	66.25	90.25
Ortho-DiaFLD	90.50	62.75	43.50	42.50	98.25
DiaFLD + 2DFLD	53.50	38.50	84.25	83.25	54.25
Ortho-DiaFLD + 2DFLD	69.50	48.50	35.00	38.00	70.50

made orthogonal. For this, the ARA of all the methods were computed for varying confidence intervals. Tables 7–9 compares the ARA of all FLD based methods for confidence intervals 90%, 95% and 98% respectively. Some observations from this experiments are:

(1) Most of the FLD based method's accuracy reduced as the confidence interval increased, unlike the orthogonalized FLD method whose performance was consistent as the confidence interval was increased.
(2) The accuracy of one-dimensional FLD methods such as orthogonalized FLD, FLD + DCT and FLD + wavelet is less than in the results obtained by their non-orthogonal counterparts for all confidence intervals.
(3) Conversely, the performance of the matrix based (2D) orthogonalized FLD methods are better (except under exponential and Weibull noise) than the results obtained in their original form. This is because in 1D methods, it is impossible to capture the advantage of orthogonality with respect to two components (spatial and feature variations like frequency components). In typical images, redundancy exists in both these domains. Therefore in 2D version of these algorithms there is the potential to minimize this redundancy in both spatial (left-to-right, top-to-bottom) and frequencies (horizontal and vertical).
(4) In the case of exponential and Weibull noises, most orthogonal methods irrespective of 1D or 2D versions, performance is very poor. This deserves further study which is under investigation.

In more general cases, it is safe to conclude the following under noise conditions: The FLD methods outperformed their orthogonalized versions for one dimensional data. However in dealing with images as 2D, orthogonalized algorithms outperformed their corresponding non-orthogonalized form.

Table 9 – Results of FLD based methods under noise conditions: Conventional vs. Orthogonal (98% confidence interval).

Methods	Different noise conditions				
	Gaussian	Salt-and-Pepper	Exponential	Weibull	Beta
FLD	94.75	67.00	73.25	59.75	99.50
Ortho-FLD	89.00	61.25	24.25	32.00	90.50
FLD + DCT	95.25	65.25	68.25	54.00	98.75
Ortho-FLD + DCT	87.50	60.25	15.50	30.25	90.25
FLD + Wavelet	89.00	60.75	90.50	92.75	98.50
Ortho-FLD + Wavelet	74.25	51.50	81.00	71.25	87.25
2DFLD	76.75	54.25	77.50	71.75	77.75
Ortho-2DFLD	86.75	59.25	24.50	31.75	97.00
A2DFLD	87.25	58.25	60.50	48.25	89.50
Ortho-A2DFLD	86.75	59.75	16.00	29.25	95.50
$2D^2$LDA	73.25	49.25	92.50	96.25	77.00
Ortho-$2D^2$LDA	86.50	59.00	37.75	38.50	94.75
DiaFLD	87.25	58.75	74.50	66.75	91.25
Ortho-DiaFLD	91.00	62.75	39.50	39.75	98.00
DiaFLD + 2DFLD	54.25	40.25	87.50	89.25	60.00
Ortho-DiaFLD + 2DFLD	75.50	52.00	45.25	43.25	79.75

4.4. *Results on the FERET database for SSPP problem*

One of the most practical challenges faced by the current face recognition community is theperformance of the algorithms under a limited number of representative training sets.[9] Many face recognition algorithms reported here rely on the representative training samples for each person. Their performance is greatly hindered and a few of them, like *Fisherface* and its variants, completely collapse (because of the non-invertiblity of the within class matrix) if presented under the so-called SSPP problem. Hitherto, some interesting studies have been reported by Zhou et al. [11,12,17,13,51] and others [16] to tackle this situation. In this experiment, we consider these methods along with the original PCA method for comparative purposes. We compare the recognition performance of these algorithms for the 90% confidence interval (because these algorithms are verified for 90% confidence intervals in respective papers) and when the size of the database increases gradually from 20 to 200 insteps of 20. This result is depicted in Fig. 3.

From Fig. 3, it can be ascertained that FLD based methods (SpFLD[10] [17] and Sampled FLD[11] [16]) dominated other PCA based variants. This is apparent when the size of the database is increased (100 onwards). Among others, the PC^2A (PCSA) method obtained good accuracy and significantly outperformed other methods by a good margin. The performance of the SVD perturbed algorithm (SPCA) [13] and Enhanced-PC^2A method (EPCSA) deteriorated consistently when the size of the database was increased.

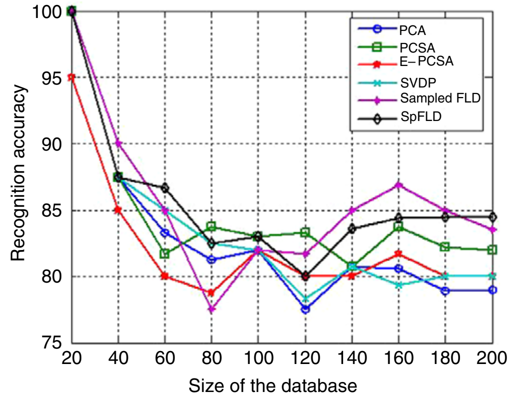

Fig. 3 – Experiment on FERET database under SSPP problem.

For a fair comparison and qualitative assessment of the algorithms that can handle the *SSPP* problem, we conducted a similar experiment to compare the performance of other PCA based methods with those algorithms whose performance under *SSPP* was effective and acceptable in the previous experiment. This result is shown in Table 10. It is clear from the Table 10 that, the PCA method implemented in the DCT and wavelet domains outperformed both PC^2A [11] and Sampled FLD [16] methods. This suggests that instead of going for an extra preprocessing step (as in the case of PC^2A and other associated algorithms [12,17,13]), it would indeed be useful and of great advantage if the original PCA method could be implemented in the domains like DCT and wavelet. The other matrix based PCA methods [33,36,40] also performed well under SSPP problem and outperformed other conventional algorithms. It is worth mentioning that, from the experimental data presented in Table 10, it is difficult to assess the behavior of these methods from the perspective of performance evaluation, because, it is difficult to make out the best performing algorithm. In spite of this, the result presented instigates us to think in the following way, which

[9] Given a stored database of faces, the goal is to identify a person from it later in time in any different and unpredictable poses, lighting, etc from just one image [51].
[10] We fixed the size of the sub-pattern to 5 × 5 and used the nearest neighbor strategy for the classification while implementing this algorithm.
[11] While implementing this algorithm, we sampled the original image by 2 pixels along vertical and horizontal directions, which generated four sampled image for a single image.

Table 10 – Comparing different methods under *SSPP* condition with a 90% confidence interval.

Database size	PC^2A	Sampled FLD	2DPCA	$2D^2PCA$	DiaPCA	PCA + DCT	PCA + Wavelet
20	100	100	100	100	100	100	100
40	87.50	90.00	85.00	87.50	87.50	87.50	87.50
60	81.66	85.00	88.33	88.33	83.33	86.67	86.67
80	83.75	77.50	85.00	88.75	83.75	83.75	90.00
100	83.00	82.00	85.00	89.00	85.00	83.00	88.00
120	83.33	81.66	86.66	88.33	85.00	86.66	87.50
140	80.71	85.00	87.85	87.86	85.71	84.28	87.14
160	83.75	86.87	88.75	89.38	86.25	85.00	87.50
180	82.22	85.00	88.77	88.33	86.67	85.00	87.22
200	82.00	83.50	87.50	89.00	87.00	86.00	87.50

deserves further study. What would happen to the algorithms which were specially designed to handle the *SSPP* [12,11,16,13] problem, had they been implemented in domains like DCT or wavelet? Conversely, What would happen to algorithms such as PCA + DCT [31], PCA + Wavelet [32] and other matrix based PCA methods [33,36,40], had they been backed with preprocessing step that is being used in [11–13]? We leave this to the interest of readers to explore, which might lead to some interesting findings.

4.5. *Results on the Yale database*

Experiments on the Yale database are carried out by adopting *Leave-one-out* strategy, i.e., leaving out one image per person each time for testing, and all of the remaining images are used for training. Each image is manually cropped and resized to 235 × 195 pixels in our experiment. This experiment is repeated 11 times (for every algorithm) by leaving out a different image per person every time. Again, results for PCA and FLD based methods are given separately. Table 11 shows the average of eleven times results for varying confidence intervals.

By looking at the entries of Table 11, many meaningful inferences can be drawn. At first glance on the table, we notice that the orthogonalized FLD method obtained a consistent accuracy for varying confidence intervals and the performances of all other methods are far behind this method. It is ascertained from this experiment that, except the DiaFLD method, no other matrix based FLD method obtained favorable result. This is again due to the dependency of projection vectors or due to the non-orthogonality among the projection vectors. We can see that the performance of $2D^2LDA$ and DiaFLD + 2DFLD got poor results and was unacceptable. Their performance degraded as the percentage of the confidence interval increased. So far in the performance contest, the Orthogonalized FLD method remained promising.

Similar experiments conducted on PCA based methods are presented in Table 12. From the table, it is clear that the classification performance of most of the algorithms are comparable to each other and they exhibited a consistent performance for varying confidence intervals. The so-called linear dependency problem does not exist in PCA based methods because they are inherently orthogonal in nature.

4.6. *Results on the COIL-20 object database*

Inspired by the conviction that successful methods developed for face recognition (such as Eigenface [5]) can be extended for object recognition, in this section, we verify the applicability of the subspace methods for objects by considering the COIL-20 database. This database contains 1440 gray level images of size 128 × 128 pixels corresponding to 20 objects.

For comparative analysis of PCA based approaches, we consider first 36 views of each object for training and the remaining views for testing. So, the size of both the training and testing database is 720. Table 13 gives the comparison of all the methods on best recognition accuracy and corresponding dimension of feature vector/matrices for 98% confidence intervals.

It is clear that the 2DPCA, DiaPCA and Enhanced-PC^2A methods outperformed all other methods in terms of accuracy. It is interesting to note that the number of Eigenfaces (or projection vectors) taken by the latter method is also less than other methods, next only to the PCA +

Table 11 – Comparing different FLD based methods on Yale database under *Leave-one-out* strategy.

Methods	Confidence intervals (%)			
	95	97	98	99
FLD	86.0606	86.667	84.8484	85.4545
Orthogonalized FLD	89.6969	88.4848	89.69	87.8787
FLD + DCT	84.2424	84.2424	86.0606	86.6667
FLD + Wavelet	84.2424	83.6363	83.0303	86.0606
2DFLD	82.4242	81.81812	81.2121	81.2121
Alternative 2DFLD	78.7878	78.1818	77.5757	76.9696
DiaFLD	86.667	87.2727	87.2727	87.2727
DiaFLD + 2DFLD	24.2424	22.4242	22.4242	21.2121
$2D^2LDA$	21.2121	20.6060	20.00	20.00
dcFLD	82.2424	83.6363	82.2424	82.2424

Table 12 – Comparing different PCA based methods on Yale database under *Leave-one-out* strategy.

Methods	Confidence intervals (%)			
	95	97	98	99
PCA	87.2727	87.8787	87.8787	86.667
PCA + DCT	86.6667	87.2727	87.2727	85.4545
PCA + Wavelet	87.2727	87.8787	87.8787	87.8787
PC^2A	87.2727	87.2727	87.2727	87.2727
Enhanced PC^2A	86.0606	86.0606	86.0606	86.0606
2DPCA	87.2727	87.2727	87.8787	87.8787
$2D^2PCA$	91.5151	90.9090	91.5151	90.3030
DiaPCA	88.4848	87.8787	87.8787	87.8787
SpPCA	87.2727	87.2727	87.8787	87.8787
SPCA	84.2424	83.6363	83.0303	83.0303

Table 13 – Comparing various methods on COIL 20 object database (98% confidence interval).

Methods	Accuracy (%)	Dimension
PCA	87.64	612
PCA + DCT	89.51	550
PCA + Wavelet	90.00	586
PC^2A	88.61	608
Enhanced PC^2A	92.77	561
SPCA	88.61	607
2DPCA	92.78	112 × 23
$2D^2PCA$	90.35	39 × 23
Alternate 2DPCA	87.98	39 × 92
DiaPCA	92.63	112 × 25
SpPCA	89.76	112 × 17

Table 14 – Recognition accuracy (%) of FLD based methods on COIL 20 database (for 98% confidence interval) in two different forms.

Methods	Conventional form	Orthogonal form	Relative improvement (in %)
FLD	81.74	92.84	13.57
FLD + DCT	82.22	92.70	12.74
FLD + Wavelet	83.47	93.68	12.23
2DFLD	83.26	95.06	14.17
Alternate 2DFLD	86.11	92.77	7.73
$2D^2LDA$	61.66	94.30	52.93
DiaFLD	86.45	93.19	7.79
DiaFLD+2DFLD	84.57	88.47	4.61
dcFLD	90.31	92.47	2.39

DCT method. However, PCA + DCT method involves lot of computations. The accuracy obtained by other methods are comparable with each other.

Table 14 presents the results of FLD based methods for object recognition. All these algorithms are also implemented in their orthogonal form. One can observe the significance of this *orthogonalization* in almost all the cases. In particular, the $2D^2LDA$ method, whose accuracy has improved by more than 50% when compared to its performance in conventional non-orthogonal form. The orthogonalized result of the 2DFLD method is the highest recognition accuracy achieved for object recognition compared to any other method including the PCA based ones.

5. Discussion and conclusion

Subspace analysis, which is one of the fastest growing areas of signal processing in general and more particular in face recognition research, has made an overwhelming impression in recent years due to its efficiency and ability to withstand image variations such as occlusion, noise etc. In this paper, we have provided a comprehensive performance evaluation of some popular methods starting from the *Eigenface* method [5] to the more recently proposed ones. We first explained what the subspace analysis method is, and revealed its influence on current face recognition research. We then described various algorithms based on the time/spatial domain and also in frequency domain which uses subspace concepts.

Beginning with the two state-of-the art subspace methods, Eigenfaces and Fisherfaces, we implemented many other methods and carried out a variety of experiments under various important test conditions such as pose variations, influence of projection vectors on the algorithm, changes in lighting and facial expressions, and under noise conditions. The latter test condition, i.e., noise environment, plays a vital role and it has lately got tremendous attention in many signal processing communities while testing the efficacy of algorithms. This, we believe, could help us examine the best performing algorithms for real time scenarios. However, most contemporary research is not focussed on it. One of the global requirements is to look for algorithms which work very well to reasonably well across all kinds of noise contaminations. Hence in our work, we have considered five different noise conditions namely, Gaussian, Weibull, Exponential, Beta and Salt-and-Pepper of various intensities to examine the robustness of the algorithms.

We first carried out the comparative analysis on the ORL database under the conditions when pose and sample size were varied. On this database two types of experiments were performed: (i) Checking the recognition accuracy of an algorithm by varying the number of projection vectors and choosing the best among that and (ii) Verifying the performance for a fixed confidence interval. We believe that the former condition may not be always suitable for real time cases because of its hard and monotonous computations involved just to classify an image to a correct class. Hence the classification performance of algorithms throughout the paper was computed for a fixed confidence interval. However, under this test condition, the accuracy of FLD based algorithms is greatly hindered due to the high correlation that can exist among its projection vectors. Nevertheless, to alleviate this problem, a simple technique to orthogonalize the projection vectors using the Gram–Schmidt decomposition process was suggested. Later, using this database, we verified the robustness of various algorithms under five different noise conditions. Under this experiment, $2D^2PCA$ and LPP methods emerged as the only algorithms to exhibit a high degree of robustness to all noise conditions. We have also presented a detailed analysis about the behavior of other algorithms under different noise conditions.

The single-sample-per-person problem is gaining more popularity in current face recognition technology due to its challenges and significance regarding real world applications [51]. In this paper, we have also addressed this issue and revealed its implication to face recognition research. We have also conducted an experiment on a partial FERET database to review the relative performance of some current algorithms under the so-called *SSPP* condition. Besides these, we have also verified the pertinence of algorithms such as PC^2A and enhanced PC^2A, SPCA, which were specially designed to handle the *SSPP* problem under multiple training sample scenario.

The Yale database was used to examine the performance of algorithms under varied facial expression and lighting conditions. *Leave-one-out* strategy was used to evaluate the performance of the algorithms for varying confidence

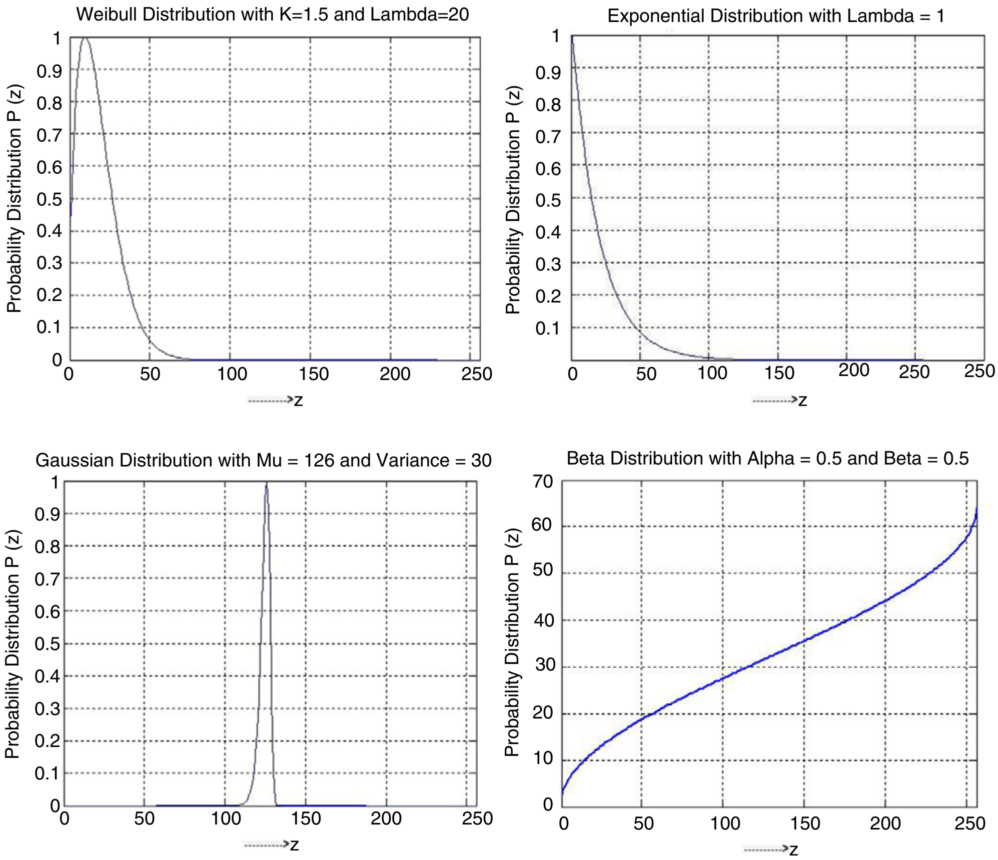

Fig. A.1 – Clockwise from top-left: (a) Weibull distribution (b) Exponential distribution (c) Beta distribution and (d) Gaussian distribution.

intervals. Apart from checking the performance of these algorithms on standard face databases, we have also checked the applicability of these face recognition algorithms for object recognition by considering the COIL-20 database. It is worth mentioning that some closely related face processing tasks such as, face detection, facial expression analysis and normalization were deliberately left out in the paper. For detailed discussion on these topics, one can refer to [52,53,2].

To summarize, the following are the essential contributions of this work:

(1) Performance evaluation of twenty-five different subspace algorithms under several real time test conditions.
(2) Suggested a mathematical model (Gram–Schmidt decomposition process) as a preprocessing step for FLD based algorithms.
(3) Attempted to test the face recognition algorithms for object recognition tasks.
(4) Conducted extensive experiments to study the robustness of subspace algorithms under five different noise conditions.
(5) Provided the analysis of experiments and substantiated through appropriated reasons based on the performance.
(6) We have inched closer to qualifying a globally *best* algorithm for face recognition in real time scenarios.

Although these subspace analysis methods obtain satisfactory results in one or the other testing conditions, more study deserves to be carried out on how the generalization capacity of algorithms differ, under similar test conditions, if they had been implemented in other domains like DCT, Block DCT, and wavelets. It is also meaningful to investigate the performance of algorithms, under clean and as well the noisy conditions, for a varying percentage of DCT coefficients and also for a different basis of wavelets. The latter issue is much more interesting because choosing the best basis is still an unresolved problem [54]. Moreover, face samples are non-stationary in nature which obviously necessitates the use of wavelets in analysing and describing them. This gives a lot of scope for further study and future avenues of work. Specific modification of these algorithms and its efficacy under streaming issues needs to be looked into. This aspect will gain more importance as IP-TV and DTH schemes of image transport are becoming global standards.

Acknowledgement

We would like to thank the anonymous reviewers for their useful comments which considerably improved the presentation of the paper. Authors would also like to thank Dr. Manjunath Aradhya, Dayananda Sagar College of Engineering, Bangalore for his contributions in this work.

Appendix A. Noise modelling

In this section we present the details of noise modeling using five various distributions in their discretized version.

The noise conditions considered are Gaussian, Weibull, Exponential, Beta and Salt-and-Pepper. Most of the literature contain noise modeled as Gaussian and Salt-and-Pepper. Practical real time tasks do involve extreme valued distributions like Weibull (atmosphere disturbance, satellite images with cloud clutter etc). We have used models like exponential and beta in order to capture other categories in between gaussian and Weibull distributions. Note that the Gaussian used is narrow in gray scale therefore its effect is wide band in the frequency domain.

We add the spatial noise to the original image, which is modeled by the statistical behavior of the gray level values. This can be viewed as random variables, characterized by the Probability Distribution Functions (PDF). The following are the PDFs that we use to model the spatial noise.

A.1. *Gaussian noise*

To model the Gaussian noise, the Gaussian distribution (also called the normal distribution) is used, which is of great importance in many fields. It is a family of distributions of the same form but differing in their scale parameter σ (stand deviation) and the location parameter μ (mean). The PDF of a Gaussian random variable is given as follows:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(z-\mu)^2}{2\sigma^2}} \tag{A.1}$$

where z is the gray value, μ is the mean of z, and σ is its standard deviation. A plot of this function is given in Fig. A.1. The control parameters μ and σ are fixed to 0 and 0.5 respectively in our experiment.

A.2. *Weibull noise*

The Weibull noise can be modeled using the Weibull distribution which has the PDF of the following form.

$$p(z, k, \lambda) = \frac{k}{\lambda}\left(\frac{z}{\lambda}\right)^{k-1} e^{-\left(\frac{z}{\lambda}\right)^k} \tag{A.2}$$

for $x \geq 0$ and $f(z, k, \lambda) = 0$ for $x < 0$, where k is the shape parameter and λ is the scale parameter of the distribution. This distribution can mimic the behavior of other statistical distributions, i.e., when $k = 1$ the Weibull distribution reduces to the exponential distribution and when $k = 3.4$ it behaves as the standard normal distribution. A plot of this function is given in Fig. A.1. In our case, we have assigned the value of k and λ to 0.5 and 2 respectively.

A.3. *Exponential noise*

The exponential PDF has the following form:

$$p(z, \lambda) = \lambda e^{-\lambda z} \tag{A.3}$$

$p(z, \lambda) = 0$ for $\lambda < 0$. Where λ is called the rate parameter of the distribution. The plot of this function is shown in Fig. A.1. We can also model the exponential noise using the Weibull distribution by initializing its shape parameter k to 1 in Eq. (A.2). We have fixed the value of λ to 1 while modeling the exponential noise.

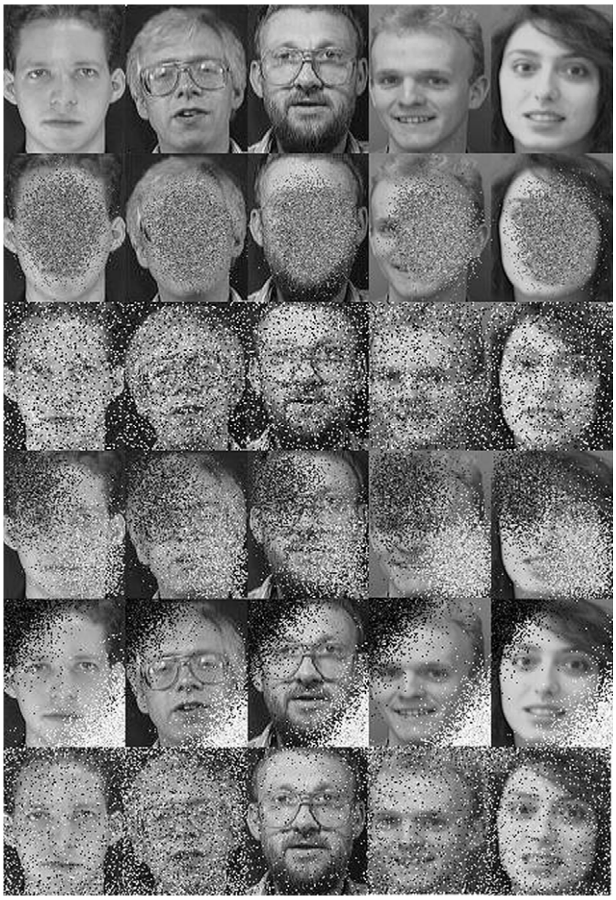

Fig. B.1 – Few example noisy images of varying densities. Row-wise (1) Original images, (2) Gaussian noise, (3) Salt-and-Pepper noise, (4) Weibull noise, (5) Exponential noise and (6) Beta noise.

A.4. *Salt-and-Pepper noise*

The Salt-and-Pepper behavior can be modeled by any of the above three distributions by modifying the control parameter of the respective distribution. However, it can also be modeled using the following PDF:

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise.} \end{cases} \tag{A.4}$$

If $b > a$, graylevel b will appear as a light dot in the image and conversely, graylevel a will appear as a dark dot. We have used the standard Matlab command *imnoise* to create theSalt-and-Pepper effect for a given image. Since it is too common and the discretized version is used, the distribution plot is not given.

A.5. *Beta noise*

The beta distribution is a family of continuous probability distributions defined on the interval $[0, 1]$ differing in the values of their two non-negative shape parameters α and β. The PDF of this distribution is given by

$$p(z; \alpha, \beta) = \frac{1}{\mathrm{B}(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}. \tag{A.5}$$

The plot of this function is shown in Fig. A.1. We have fixed the values of both the shape parameters α and β to 0.5.

Appendix B. Noise test sample composition

We selected first image from each class and generated ten noisy images by varying the noise density from 0.1 to 1.0 insteps of 0.1. Likewise, effectively 50 noise images are created corresponding to five different distributions. Suppose that there are C number of classes, then a total of $50 \times C$ noisy images will be generated. In our simulations,we considered the ORL database which contains 40 classes. Hence, 2000 noisy images are created corresponding to this image set. Fig. B.1 shows the example noisy images created according to the aforementioned distributions.

REFERENCES

[1] A.F. Abate, Michele Nappi, Daniel Riccio, Gabriele, 2D and 3D face recognition: A survey, Pattern Recognition Letters 28 (14) (2007) 1885–1906.

[2] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: A survey, Proceedings of the IEEE 83 (5) (1995) 705–740.

[3] Stan Z. Li, Anil K. Jain, Handbook of Face Recognition, Springer, 2004.

[4] I. Craw, N. Costen, How should we represent faces for automatic recognition? IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (8) (1999) 725–736.

[5] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[6] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[7] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, 1990.

[8] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

[9] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Transactions on Neural Networks 13 (6) (2002) 1450–1464.

[10] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[11] Jianxin Wu, Zhi-Hua Zhou, Face recognition with one training image per person, Pattern Recognition Letters 23 (2002) 1711–1719.

[12] Songcan Chen, Daoqiang Zhang, Zhi-Hua Zhou, Enhanced PC^2A for face recognition with one training image per person, Pattern Recognition Letters 25 (2004) 1171–1181.

[13] Daoqiang Zhang, Songcan Chen, Zhi-Hua Zhou, A new face recognition method based on SVD perturbation for single example image per person, Applied Mathematics and Computation 163 (2005) 895–907.

[14] Aleix M. Martinez, Recognizing imprecisely localized partially occluded, and expression variant faces from a single sample per class, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (6) (2002) 748–763.

[15] X. Tan, Songcan Chen, Z.-H. Zhou, F. Zhang, Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft kNN ensemble, IEEE Transactions on Neural Networks 16 (4) (2005) 875–886.

[16] Hongtao Yin, Ping Fu, Shengwei Meng, Sampled FLDA for face recognition with single training image per person, Neurocomputing 69 (16–18) (2006) 2443–2445.

[17] Songcan Chen, Jun Liu, Zhi-Hua Zhou, Making FLDA applicable to face recognition with one sample per person, Pattern Recognition Letters 23 (2002) 1711–1719.

[18] J. Huang, P.C. Yuen, W.S. Chen, J.H. Lai, Component based LDA method for face recognition with one training sample, in: IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003, pp. 120–126.

[19] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 696–710.

[20] P.J. Phillips, Support vector machines applied to face recognition, Advances in Neural Information Processing Systems 11 (3) (1998) 809.

[21] Stan Z. Li, J. Lu, Face recognition using the nearest feature line method, IEEE Transactions on Neural Networks 10 (2) (1999) 439–443.

[22] C. Liu, H. Wechsler, Evolutionary pursuit and its application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (6) (2000) 570–582.

[23] Bhagavathula V.K. Vijayakumar, Marios Savvides, Chunyan Xie, Correlation pattern recognition for face recognition, Proceedings of the IEEE 94 (11) (2006) 1963–1976.

[24] Xiaogang Wang, Xiaoou Tang, A unified framework for subspace face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1222–1227.

[25] Hyun Chul Kim, Daijin Kim, Sung Yang Bang, Face recognition using the mixture-of-Eigenfaces method, Pattern Recognition Letters 23 (2002) 1549–1558.

[26] Hyun Chul Kim, Daijin Kim, Sung Yang Bang, Face recognition using LDA mixture model, Pattern Recognition Letters 24 (2003) 2815–2821.

[27] S. Noushath, Ashok Rao, G. Hemantha Kumar, Mixture-of-Laplacianfaces and its application to face recognition, in: 2nd International Conference on Pattern Recognition and Machine Intelligence, PReMI-07, 2007, pp. 568–575.

[28] R. Raghavendra, Ashok Rao, G. Hemantha Kumar, Subjective performance of texture based algorithm for face verification: The role of databases, in: ICVGIP2008, 2008, pp. 421–428.

[29] Songcan Chen, Yulian Zhu, Subpattern-based principal component analysis, Pattern Recognition 37 (2004) 1081–1083.

[30] S. Noushath, G. Hemantha Kumar, V.N. Manjunath Aradhya, P. Shivakumara, Divide-and-conquer strategy incorporated Fisher linear discriminant analysis for efficient face recognition, in: International Conference on Advances in Pattern Recognition, 2007, pp. 40–45.

[31] Weilong Chen, Meng Joo Er, Shiqian Wu, PCA and LDA in DCT domain, Pattern Recognition Letters 26 (2005) 2471–2482.

[32] Jen-Tzung Chien, Chia-Chen, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1644–1649.

[33] J. Yang, D. Zhang, A. Frangi, J. Yang, Two-dimensional PCA: A new approach to appearance-based face representation and recognition, IEEE Transactions on Pattern Analalysis and Machine Intelligence 26 (1) (2004) 131–137.

[34] Ming Li, Baozong Yuan, 2DLDA: A statistical linear discriminant analysis for image matrix, Pattern Recognition Letters 26 (5) (2005) 527–532.

[35] H. Xiong, M. Swamy, M. Ahmed, Two-dimensional FLD for face recognition, Pattern Recognition 38 (2005) 1121–1124.

[36] D. Zhang, Z.H. Zhou, $2D^2PCA$: 2-directional 2-dimensional PCA for efficient face representation and recognition, Neurocomputing 69 (1–3) (2005) 224–231.

[37] Hui Kong, Lei Wang, Eam Khwang Teoh, Xuchun Li, Jian-Gang Wang, Ronda Venkateshwarlu, Generalized 2D principal component analysis for face image representation and recognition, Neural Networks 18 (2005) 585–594.

[38] Wangmeng Zuo, David Zhang, K. Wang, Bidirectional PCA with assembled matrix distance measure, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 36 (4) (2006) 863–872.

[39] S. Noushath, G. Hemantha Kumar, P. Shivakumara, $2D^2$LDA: An efficient approach for face recognition, Pattern Recognition 39 (7) (2006) 1396–1400.

[40] Daoqiang Zhang, Zhi-Hua Zhou, Songcan Chen, Diagonal principal component analysis for face recognition, Pattern Recognition 39 (2006) 140–142.

[41] S. Noushath, G. Hemantha Kumar, P. Shivakumara, Diagonal Fisher linear discriminant analysis for efficient face recognition, Neurocomputing 69 (2006) 1711–1716.

[42] Dacheng Tao, Xuelong Li, Xindong Wu, Weiming Hu, Stephen J. Maybank, Supervised tensor learning, in: ICDM—Fifth IEEE International Conference on Data Mining, 2005, pp. 450–457.

[43] Dacheng Tao, Xuelong Li, Xindong Wu, Weiming Hu, Stephen J. Maybank, Supervised tensor learning, Knowledge and Information System 13 (1) (2007) 1–42.

[44] Dacheng Tao, Xuelong Li, Xindong Wu, Stephen J. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10) (2007) 1700–1715.

[45] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face recognition algorithms, Image and Vision Computing 16 (5) (1998) 295–306.

[46] H. Murase, S.K. Nayar, Visual learning and recognition of 3D objects from appearance, International Journal of Computer Vision 14 (1) (1995) 5–24.

[47] Fengxi Zong, Shuhai Liu, Jingyu Yang, Orthogonalized Fisher discriminant, Pattern Recognition 38 (2005) 311–313.

[48] Gilbert Strang, Linear Algebra and its Applications, fourth ed., Thomson Brooks/Cole, 2007.

[49] A.N. Rajagopalan, Rama Chellappa, Nathan T. Koterba, Background learning for robust face recognition with PCA in the presence of clutter, IEEE Transactions on Image Processing 14 (2005) 832–843.

[50] Xiaogang Wang, Xiaou Tang, Bayesian face recognition based on gaussian mixture models, in: ICPR'04—Seventeenth IEEE International Conference on Pattern Recognition, 2004, pp. 142–145.

[51] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, Fuyan Zhang, Face recognition from a single image per person: A survey, Pattern Recognition 39 (9) (2006) 1725–1745.

[52] M.H. Yang, D. Kreigman, N. Ahuja, Detecting faces in images: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58.

[53] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human faces and facial expression: A survey, Pattern Recognition 25 (1) (1992) 65–77.

[54] A. Jensen, A. La Cour-Harbo, Ripples in Mathematics: The Discrete Wavelet Transform, Springer, International Edition, 2001.