Image and Vision Computing 30 (2012) 355–366

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

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

On nonlinear dimensionality reduction for face recognition ☆

Weilin Huang, Hujun Yin *

School of Electrical and Electronic Engineering, The University of Manchester, Manchester, M60 1QD, UK

ARTICLE INFO

Article history:
Received 21 September 2011
Received in revised form 19 January 2012
Accepted 25 March 2012

Keywords:
Dimensionality reduction
Nonlinear manifold
Subspace learning
Principal component analysis
Feature representation
Face recognition

ABSTRACT

The curse of dimensionality has prompted intensive research in effective methods of mapping high dimensional data. Dimensionality reduction and subspace learning have been studied extensively and widely applied to feature extraction and pattern representation in image and vision applications. Although PCA has long been regarded as a simple, efficient linear subspace technique, many nonlinear methods such as kernel PCA, local linear embedding, and self-organizing networks have been proposed recently for dealing with increasingly complex nonlinear data. The intensive research in nonlinear methods often creates an impression that they are highly superior and preferred, though often limited experiments were given and the results not tested on significance. In this paper, we systematically investigate and compare the capabilities of various linear and nonlinear subspace methods for face representation and recognition. The performances of these methods are analyzed and discussed along with statistical significance tests on obtained results. The experiments on a range of data sets show that nonlinear methods do not always outperform linear ones, especially on data sets containing noise and outliers or having discontinuous or multiple submanifolds. Certain nonlinear methods with certain classifiers do yield better performances consistently than others. However, the differences among them are small and in most cases are not significant. A measure is used to quantify the nonlinearity of a data set in a subspace. It explains that good performances are achievable in reduced dimensions of low degree of nonlinearity.

1. Introduction

With the ever fast increasing quantity and complexity of the data in many fields, it becomes difficult, challenging or even impossible to deal with raw data directly. Dimensionality reduction has become a necessity for pre-processing data so to facilitate data management, representation and classification. It aims to represent the data in a low-dimensional, essential subspace that captures the intrinsic nature of the data. Images often contain a large number of pixel values and are represented as high-dimensional vectors or arrays. Operating directly on these vectors is inefficient, would lead to high computational costs and storage demands, and poses the curse of dimensionality to many learning tasks. An effective subspace representation has thus become desirable in many image processing applications.

In face recognition, appearance-based approach has been widely used. The holistic and component-based methods are two main ways for representing facial appearances. In holistic representation, a facial image is considered as a vector of pixels and is represented as a single point in the high-dimensional space. Subspace methods are applied in unsupervised manner to reduce the high-dimensional data onto a lower dimensional space while retaining intrinsic features for further classification (e.g. eigenfaces [1] and fisherfaces [2]). Component-based representation is in favor of representing a face by multiple local features. It divides a face image into local sub-blocks (or regions) and then subspace methods are applied on these blocks to extract compact representations (e.g. modular eigenspaces [3] and LBP-based local facial representation [4,5]). In both representations, dimension reduction is applied to a great extent in order to extract features and to facilitate further processing such as classification. Supervised dimension reduction, which jointly optimizes the reduction and classification, is beyond the scope of this paper.

PCA is a primary dimensionality reduction technique and is regarded as the theoretical foundation of many nonlinear subspace techniques. It seeks a linear projection that best fits a data set in the least-square sense and has been widely used due to its computational and analytical simplicity [6]. Eigenface [1] is a well-known application of PCA for face recognition. However, as much recent literature has pointed out, the linearity of PCA limits its power for complex data sets as it is unable to capture nonlinear structure of the data defined by beyond second order statistics. Various nonlinear techniques have been proposed. Kernel PCA (KPCA) [7] extends PCA to nonlinearity by projecting the data into a higher-dimensional feature space via the kernel trick. Manifold-based learning techniques, such as local linear embedding (LLE) [8], ISOMAP [9] and curvilinear component analysis (CCA) [10], and their linear variants (e.g., locality preserving projection (LPP) [11] and orthogonal locality preserving

☆ This paper has been recommended for acceptance by Stefanos Zafeiriou.

* Corresponding author. Tel.: +44 161 306 8714.
E-mail addresses: weilin.huang@postgrad.manchester.ac.uk (W. Huang), h.yin@manchester.ac.uk (H. Yin).

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2012.03.004

projection (OLPP) [12]), detect underlying nonlinear data manifold by preserving local relationships or distances between data points in a neighborhood via minimizing a defined cost function. Several reviews of the existing dimensionality reduction and subspace methods have been given (e.g., [13–18]), and many applications can also been found in the literature, e.g. [19–24].

Adaptive neural networks provide alternative approaches to nonlinear subspace learning and dimension reduction. Kohonen's self-organizing map (SOM) [25] is an abstraction of retinotopic mapping model. Its topology-preserving property is utilized to extract and visualize relative mutual relationships among the data, and thus has been widely used for data clustering and visualization. For a more natural and direct display of data structure, the visualization induced SOM (ViSOM) [26] was proposed and further improved by a growing variant, gViSOM [27]. Local distances are preserved on the map along with the topology. It has been shown that ViSOM represents a metric scaling of the input space and has comparable capability for highly nonlinear manifold learning with other nonlinear PCA methods, such as LLE and ISOMAP [27]. Typical applications of SOM-based methods for face recognition can be found in [28,29,17].

Recently increasing effort and amount of the literature on nonlinear subspace methods indeed demonstrate better capability of nonlinear methods for capturing complex relationships of the data. However, it inevitably deludes to a "common sense" that nonlinear methods are always preferable for data representation [7–10,30,16,31,26,27]. However, most experiments with these nonlinear projections in the literature were often conducted on artificial data sets lying on an assumed low-dimensional, continuous and smooth subspace embedded in high-dimensional space, and their underlying intrinsic subspaces or manifolds are well sampled. No comprehensive evaluation on practical data such as faces has been conducted. Goldberg et al. [16] have already pointed out that the use of manifold learning on arbitrary or noisy data can be problematic. Structures of real-world data sets can be far more complicated than those assumed tidy toy data sets. It is unrealistic to imagine them to have uniform structured distributions. Murphy-Chutorian and Trivedi [32] have shown that, for head pose estimation, head pose data sets of multiple subjects (persons) with different poses or other factors (e.g. lighting) are hard to lie on a single subspace, and even variant poses for a single subject sampled from some continuous measurement device may also lead disjoint distributions.

In this paper, we present an extensive evaluation of linear and nonlinear dimensionality reduction and subspace methods on facial images. We demonstrate that linear and nonlinear subspace methods often yield similar performances in face recognition and nonlinear methods lose their superiorities on data sets when there is discontinuity in the subspace. Extensive experiments on a range of data sets were conducted to elucidate our observations. Though there has been previous work on applying some of these linear and nonlinear methods for some representation tasks and the results seem to favor nonlinear methods [23,21,33,20,28,29], their performance often vary with training/test schemes, preprocessing methods and choices of classifiers. Therefore, an objective evaluation on the properties of these linear and nonlinear methods is essential to best utilizing these advanced techniques in practical face recognition systems. Here we present a thorough investigation and a comprehensive comparison, together with statistical tests of the results. Finally a nonlinear analysis and subsequent discussions on the complexity of real-world face data are presented to further explain the findings.

The remainder of this paper is structured as follows: Section 2 briefly reviews various linear and nonlinear dimensionality reduction and subspace learning methods. Experiments on two-dimensional representation of real-world data sets and results of face recognition on various benchmark data sets are reported in Section 3, together with significance tests on these results. A nonlinearity analysis and discussion are given in Section 4, followed by conclusions in Section 5.

2. Dimensionality reduction methods

In this section, various dimension reduction methods are briefly reviewed. Their role in face representation and recognition will be described in Section 3. There are three main categories based on eigen decomposition, multidimensional scaling and self-organizing map, respectively. PCA, KPCA, LLE, Hessian LLE (HLLE), LPP, OLPP and spectral clustering belong to the first category, while ISOMAP and CCA the second, and SOM, ViSOM and gViSOM the third.

2.1. PCA-based methods

PCA is a classical linear projection aiming at finding ordered orthogonal directions of a data set. After discarding a (large) number of minor components, a (small) number of principal components are retained, which are also known as *eigenfaces* in face recognition. The data (or a face image) is effectively represented by these principal components (or eigen faces). It minimizes the L_2 norm of residual (or error) as

$$\min_{\mathbf{V}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{V}\mathbf{y}_i\|^2 \quad \text{s.t.} \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^d$, are the i-th sample in the data and projected spaces, respectively. This problem is commonly solved as a linear least-square problem by eigen-decomposition or singular value decomposition (SVD), yielding a group of orthogonal basis vectors, $\mathbf{V} = \{\mathbf{v}_k\}_{k=1}^{d} \in \mathbb{R}^{n\times d}$, whose eigenvalues are the largest and in descending order.

The projection from the original n-dimensional data space to the reduced d-dimensional subspace is presented as,

$$\mathbf{y}_i = \mathbf{V}^T\mathbf{x}_i \tag{2}$$

and the reconstruction of $\mathbf{x}_i$ is

$$\mathbf{x}'_i = \mathbf{V}\mathbf{V}^T\mathbf{x}_i. \tag{3}$$

In holistic face recognition, raw face images are projected onto a d-dimensional subspace first, and then classification is conducted in the subspace.

A variant of PCA in image processing is the block-based PCA (BPCA) [34], which operates on local blocks (e.g. $5\times5=25$) rather than entire images. An image is divided into a number of sub-blocks, and then a PCA is applied on all of these blocks of each image to reduce the block size (e.g. to 2).

Two-dimensional PCA (2DPCA) [35] is another variant of PCA operating on 2D image matrices rather than vectors. The image scatter matrix ($\mathbf{G}$) is calculated directly from the image matrices, $\mathbf{x}^m$, as follows, from which eigenvectors are computed,

$$\mathbf{G} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{x}_i^m - \overline{\mathbf{x}_i^m}\right)^t\left(\mathbf{x}_i^m - \overline{\mathbf{x}_i^m}\right) \tag{4}$$

$$\overline{\mathbf{x}_i^m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^m. \tag{5}$$

2.2. Kernel PCA

Kernel-PCA [7] is a nonlinear extension of PCA. A data set is first projected onto a high-dimensional feature space, F, by using a hypothetic nonlinear function, $\Phi(X)$. Then the standard PCA is performed in the F space via a kernel function, $k(X,Y) = (\Phi(X)\cdot\Phi(Y))$. The covariance matrix $\mathbf{K}$ is computed via the kernel function, Eq. (6), and

the projection of an image $\mathbf{x}$ onto a subspace of F is given in Eq. (7), assuming that the data is centered,

$$K_{ij} := k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\Phi(\mathbf{x}_i) \cdot \Phi\left(\mathbf{x}_j\right)\right) \tag{6}$$

$$(\mathbf{v}_k \cdot \Phi(\mathbf{x})) = \alpha_k^T \mathbf{z} \tag{7}$$

where $\mathbf{v}_k$ and α_k, $k = 1,2,\ldots,d$, are the k-th eigenvector in F space or the k-th eigenvector of the covariance matrix $\mathbf{K}$, respectively. $\mathbf{z}$ is the projected vector of the image to F, $z_i := k(\mathbf{x}_i, \mathbf{x}) = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}))$, $i = 1,2,\ldots,N$, $\mathbf{x}_i$ is the i-th training image. The radial basis function is a commonly used kernel function.

2.3. LLE

LLE [8] is another nonlinear manifold method and is able to map high-dimensional nonlinear data onto a single global coordinate system of lower dimensional subspace. A number of K_{lle} nearest neighbors are first defined for each data point, then the weight ω_{ij} of a data point $\mathbf{x}_i$ from each of its neighbors $\mathbf{x}_j$ is computed by minimizing the cost function, $E(\omega)$, Eq. (8). Finally the output vector $\mathbf{y}_i$ is reconstructed from ω_{ij} by minimizing the embedding cost function, $E(y)$, Eq. (9),

$$E(\boldsymbol{\omega}) = \min_{\omega} \sum_i \left\| \mathbf{x}_i - \sum_j \omega_{ij} \mathbf{x}_j \right\|^2 \tag{8}$$

$$E(y) = \min_{\mathbf{y}} \sum_i \left\| \mathbf{y}_i - \sum_j \omega_{ij} \mathbf{y}_j \right\|^2 \tag{9}$$

where $\sum_j \omega_{ij} = 1$, and $\omega_{ij} = 0$ if data $\mathbf{x}_j$ is not in the neighborhood of $\mathbf{x}_i$. The reconstruction weights preserve the intrinsic geometric properties of local neighborhoods and also make the mapping invariant to rotation, rescaling and translation. The optimal weights can be computed in a closed form by solving a constraint least-square problem of the cost function, Eq. (8), while the embedding vectors $\mathbf{y}_i$ in Eq. (9) are solved as an eigenvalue problem.

Further similar methods include HLLE [36], orthogonal neighborhood preserving projection (ONPP) [37] and spectral clustering [38]. These subspace methods are defined on a local neighborhood and transform global structure to local linear structures. That is, the manifold is constructed on local (linear) graphs. These methods are closely related and can be described under the regularization theory or kernel methods. The difference mainly lies in defining the local reconstruction or graph, either linearly as in LLE and HLLE or by orthogonal constraint on the projection matrix as in ONPP.

2.4. LPP

LPP [11] computes a projection that preserves a certain affinity or similarity graph constructed from input data. It defines the projected data in the same form as PCA, $\mathbf{y}_i = \mathbf{V}^T \mathbf{x}_i$, but minimizes a different object function which puts a heavy penalty on neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ if they are mapped far apart in the projected space,

$$\min_{\mathbf{v}_k} \sum_{i,j=1}^{N} \zeta_{ij} \left\| \mathbf{v}_k^t \mathbf{x}_i - \mathbf{v}_k^t \mathbf{x}_j \right\|^2 \quad k = 1, 2, \ldots, d \tag{10}$$

where ζ_{ij} is the weight of the edge connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the affinity graph. Heat Kernel and cosine model are two common approaches for computing the value of ζ_{ij} in the input space, and $\zeta_{ij} = 0$ if two points are not connected in a same neighborhood. LPP uses this affinity graph to derive an optimal projection in an effort to preserve the local structure of the data. The objective function can be easily converted to

$$\min_{\mathbf{v}_k} \mathbf{v}_k^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{v}_k \quad \text{s.t.} \quad \mathbf{v}_k^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{v}_k = 1 \tag{11}$$

where $\mathbf{L} = \mathbf{D} - \zeta$ is the graph Laplacian [11,39], and D is a diagonal matrix with $D_{ii} = \sum_j \zeta_{ij}$. Then the solution can be computed from the generalized eigenvalue problem,

$$\mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{v}_k = \lambda_k \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{v}_k. \tag{12}$$

The projection is onto eigenvectors, $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d\}$ that correspond to the smallest d eigenvalues. LPP projection is similar with Laplacian eigenmap described in [39]. Its orthogonal extension, refereed as OLPP [12] adds an orthogonal constrain on the projection $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ to give a more discriminative mapping.

2.5. ISOMAP

ISOMAP [9] seeks an underlying manifold of a data set by computing the geodesic, manifold distances between all pairs of data points. It first constructs a neighborhood graph over all data points by connecting each point to all its neighbors in the input space. Then it estimates geodesic distances of all pairs of points by computing the shortest path distances in the neighborhood graph (using the Floyd's algorithm). Finally multidimensional scaling is applied to the Grammian matrix to construct the embedding that best preserves the intrinsic geometry structure of the data.

2.6. CCA

CCA [10] is another method for nonlinear mapping. It detects the intrinsic geometric properties of the data by preserving local distance relationships via minimizing an error function defined as,

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} \left(\mathbf{I}_{ij} - \mathbf{O}_{ij}\right)^2 \varphi\left(\mathbf{O}_{ij}, \theta_y\right) \tag{13}$$

where $\mathbf{I}_{ij}$ and $\mathbf{O}_{ij}$ are the Euclidean distances between points i and j in n-dimensional data space and d-dimensional output space respectively. $\varphi\left(\mathbf{O}_{ij}, \theta_y\right)$ is a monotonically decreasing neighborhood function respecting to the distance in the projected space and is used for preserving local topology and maintaining shorter distances than longer ones.

2.7. SOM

SOM [25] is an unsupervised learning algorithm that uses a set of neurons ranged often in a 2-D lattice (e.g., 10×10) to form a topological mapping of the data. Each neuron has a weight vector of the input dimensions. SOM learns topological structure of the input data by updating the weights of the winner and its neighborhood when presented with an input $\mathbf{x}$ at time t, the weight of neuron l is updated as,

$$\Delta \mathbf{w}_l(t) = \alpha(t) \eta(u, l, t) [\mathbf{x}(t) - \mathbf{w}_l(t)] \tag{14}$$

where $\alpha(t)$ is the learning rate, monotonically decreasing with time t. u denotes the winner neuron and l the updating neuron. $\eta(u,l,t)$ is the neighborhood function, which often uses a Gaussian form, $\eta(u,l,t) = exp(-\|u-l\|^2/2\sigma(t)^2)$, with σ representing the radius of the neighborhood. For dimension reduction and data visualization, high-dimensional data are projected onto the grid or the nearest neurons of the trained SOM. Then the data is represented by the 2D coordinate of the neurons. The SOM can reveal ordinal relationships of the data. However, it is unable to reproduce quantitative distances between the data points on the reduced space.

2.8. ViSOM and gViSOM

To seek a metric representation, the ViSOM [26] was proposed to preserve local distances on the map along with the topology of the data. The updating force in the SOM, as shown in Eq. (14), can be decomposed as, $[\mathbf{x}(t)-\mathbf{w}_l(t)] = [\mathbf{x}(t)-\mathbf{w}_u(t)] + [\mathbf{w}_u(t)-\mathbf{w}_l(t)]$. The second term is the lateral contraction force as it moves the neighboring neuron to the winner. In the ViSOM, the lateral contraction force is regulated so as to maintain uniform inter-neuron distances locally on the map. The update rule is [26],

$$\Delta\mathbf{w}_l(t) = \alpha(t)\eta(u,l,t)[\mathbf{x}(t)-\mathbf{w}_u(t)] + \beta[\mathbf{w}_u(t)-\mathbf{w}_l(t)] \tag{15}$$

where $\beta = \rho_{ul}/\lambda d_{ul} - 1$ is a simple form of constraint, ρ_{ul} and d_{ul} are the distance of neurons u and l in the input space and the distance of their indexes on the map respectively, and λ is a resolution parameter. The neighborhood function η is similar to that of SOM, with neighborhood radius decreasing from an initially large value to a final small value. The distance of two projected points on the map is proportional to the distance of the two points in the input space, making the scaling faithful and quantitatively measurable. The resolution of the map can be enhanced by incorporating the local linear projection (LLP) method [27],

$$\mathbf{x}' = \mathbf{w}_u + \max_{u'=u\pm1}\left\{\frac{(\mathbf{x}-\mathbf{w}_u)\bullet(\mathbf{w}_u-\mathbf{w}_{u'})}{\|\mathbf{w}_u-\mathbf{w}_{u'}\|^2}, 0\right\}. \tag{16}$$

It has been shown that SOM-based algorithms with a pre-fixed map size have difficulties to converge to highly nonlinear manifolds [27]. For improving the convergence and capability of ViSOM, an incremental or growing ViSOM (gViSOM) has been proposed. The details of the gViSOM algorithm can be found in [27].

3. Experiments and comparisons

3.1. Face data sets

The experiments were conducted on several publicly available real-world face data sets: the single subject face data set used in [9], the Olivetti Research Laboratory (ORL) data set [40], the Yale [2], the AR [41] and the CMU PIE [42] data sets. Their details are listed below:

- *Single Person Face Database* contains 698 face images of a single subject model. All images have the same size of 64×64, and are rendered with continuous variations of pose and lighting direction.
- *ORL Face Database* consists of 40 subjects, 10 different face images for each subject. Images are of the same size of 92×112 and vary (slightly) in terms of lighting conditions, facial expressions or facial details.
- *Yale Face Database* contains 165 face images of 15 subjects with size of 243×320. Each subject has 11 images with variations in both expression and lighting condition.
- *AR Face Database* consists of over 4000 color images of 126 subjects, each having 26 facial images taken in two different sessions separated by two weeks. Each session has 13 images with multiple variations in expression, illumination and occlusion (sun glasses and/or scarf). A subset of cropped faces (of size 165×120) of 50 male and 50 female subjects [19] was used. Eight faces (including neutral expression, smile, angry and scream) of each subject were used in the experiment.
- *CMU PIE Face Database* has 41,368 face images of 68 subjects, each subject consists of 13 different poses, 43 different illumination conditions and 4 different expressions. A subset of cropped face images has been used in the literature [12], where images were manually aligned to the same eye positions, cropped, and re-sized to 32×32. We used the first 34 subjects, each having 170 images with five near frontal poses (C05, C07, C09, C27, C29) and under different illuminations and expressions, in total 5780 images, in the experiment.

The magnitudes of pixel values of face images were rearranged to [0 1] in all the experiments. For computational convenience, the ORL, Yale and AR face images were re-sized to 56×46, 45×60 and 55×40 respectively in the experiments of 2D representation and vector-based face recognition, but were kept in their original sizes for the experiment of block-based face recognition. The size of PIE faces remained as 32×32 in all the experiments.

3.2. 2D representation of face images

In this set of experiments, we investigated the capabilities of linear and nonlinear subspace methods for feature representation. Raw face images of the data sets were represented as vectors in their high-dimensional input spaces. We compared the performances of various subspace methods for projecting the high-dimensional face vectors onto a 2D subspace.

In the single subject case, all images of the subject model were used. In Fig. 1, the subspaces learned by LLE, ISOMAP and gViSOM in (c), (d) and (f) seem to capture better intrinsic structure of the variances of the data than other methods. The pose and lighting are changed smoothly on these three projections. The projections of PCA, KPCA and SOM, however, are more sensitive to lighting impact than pose variation, leading to some degree of overlap for those with pose and lighting changes. Thus in these experiments most nonlinear methods seem to outperform linear PCA in extracting low-dimensional representation of the single subject.

In the multiple subject case, randomly selected ten subjects from the ORL set, five from the Yale set and three from the PIE set were used. The 2D representations of these face databases by PCA, KPCA, LLE, ISOMAP, SOM and gViSOM methods are shown in Figs. 2–4. The results presented are the typical projections of these methods. The parameters of each method were chosen to its typical performance. In the cases of the ORL and Yale sets, the projections of PCA bear similar or comparable performances to nonlinear projections in the terms of between-subject separation and within-subject compactness (exclusive of SOM projection, in which faces of same subject do not cluster metrically on the map). The nonlinear methods do not seem to be particularly advantageous. In the case of the PIE set, where only three subjects were used and each has a large number of images, the performances of linear and nonlinear methods are analogous to those in the case of the single subject. That is, each subject's variations in lighting, pose and expression were captured relatively better by the nonlinear methods. However, when a large number of subjects are presented, the mapping becomes overlapped among subjects and their variations.

In summary, in the single subject case, as the sampling rate is high, the direction and pose vary gradually and smoothly. Nonlinear methods seem able to capture better the variations and thus present better low-dimensional manifolds of the data. While in the case of multiple subjects, it is unlikely that all different subjects lie on a same low-dimensional subspace when lighting and pose variations are present. Furthermore, when different images of the same subjects were captured in different conditions, there may not be a continuous distribution even for all the images from the same subject. Thus real face data sets are unlikely to exist on a single manifold. In other words, their all possible variations cannot be smoothly represented by a single manifold either linear or nonlinear.

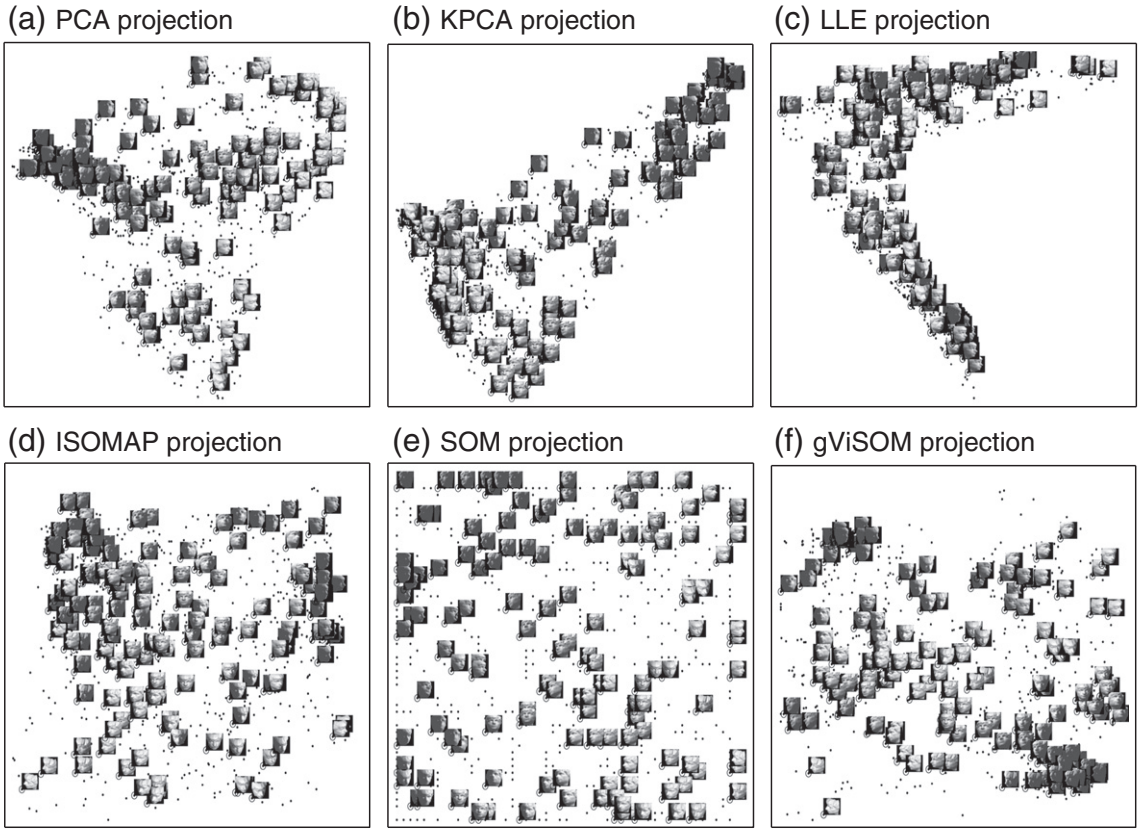

Fig. 1. 2D representations of face images of single subject with variable pose and lighting. The optimal parameters were: $\sigma_{gaus} = 4$ for KPCA, $K_{lle} = 4$, $K_{ISOMAP} = 4$ and $\lambda_{gViSOM} = 1.1$.

3.3. Face recognition

In these experiments, the capabilities of various subspace methods as feature representation for face recognition was investigated. In all implementations, the dimensions of raw face images were first reduced by one of the subspace methods. Then classification was performed by one of the commonly used classifiers described in the next few paragraphs. This is the typical two-stage approach: data/feature reduction, followed by classification. The two processes are often independent for a variety of advantages such as simple implementation and independent of data sets. Jointly optimizing the data reduction and classification is also possible and may lead to better performance. However, such optimization processes can be complex and the resulting models and parameters are optimal to the data set they are trained and may not be generalized to other data sets.

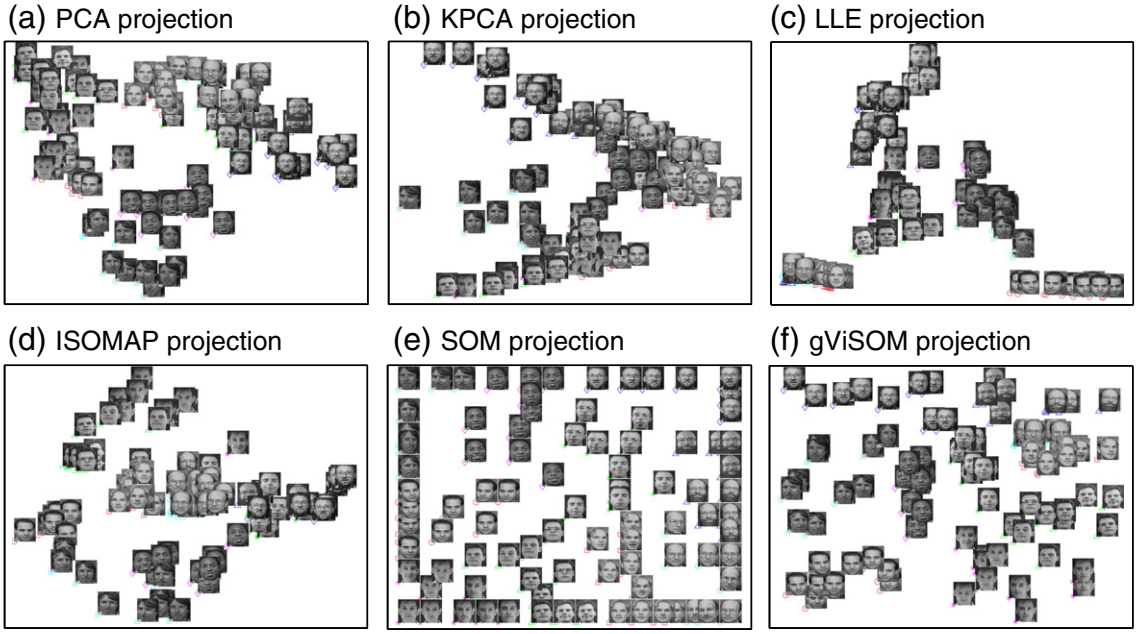

Fig. 2. 2D representations of ten subjects from the ORL set. The optimal parameters were: $\sigma_{gaus} = 4$ for KPCA, $K_{lle} = 10$, $K_{ISOMAP} = 10$ and $\lambda_{gViSOM} = 0.06$.

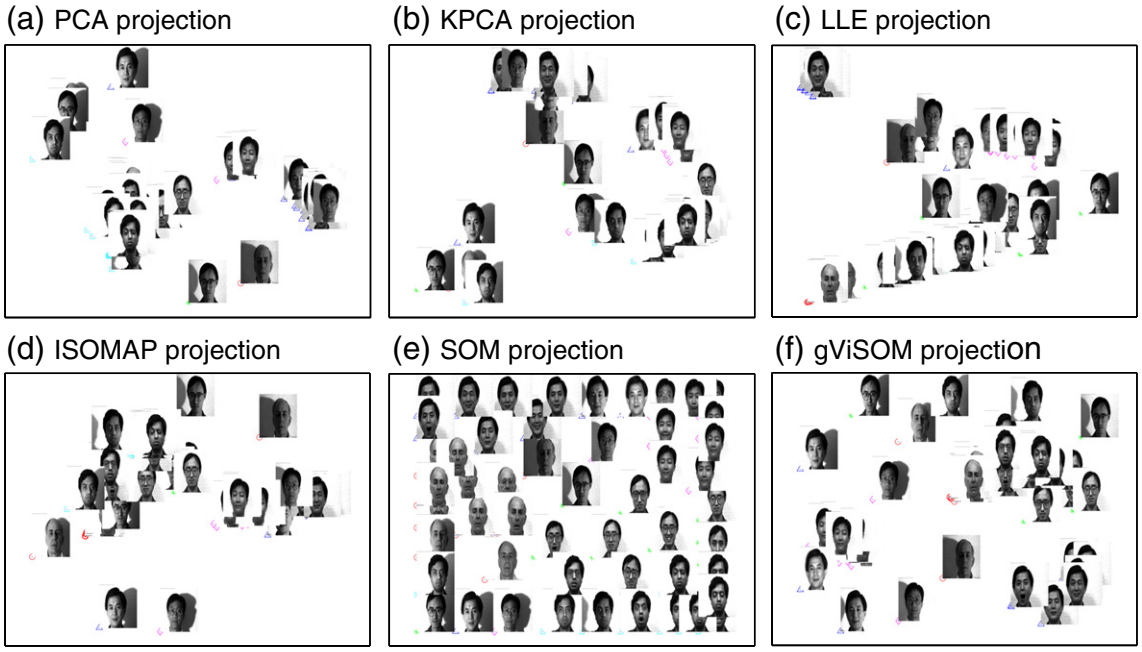

Fig. 3. 2D representations of five subjects from the Yale set. The optimal parameters were: $\sigma_{gaus} = 4$ for KPCA, $K_{lle} = 12$, $K_{ISOMAP} = 12$ and $\lambda_{gViSOM} = 0.05$.

Four commonly used classifiers were employed: the nearest-neighbor (NN), soft k-NN [29], linear discriminant analysis (LDA) [2] and support vector machine (SVM) [43]. The NN simply classifies a test sample by finding the most similar example in the training set and returning the class of that example. In the soft k-NN classifier, each principal component outputs a confidence value, which gives the degree of support for the component in every data representation, and then the final decision is given by considering all of these confidence values. The LDA, a widely used linear classifier, tries to find a linear projection of the data set that minimizes within-class scatter and maximizes between-class separation. The SVM is a nonlinear method which separates data sets by constructing hyperplanes that maximize the margins between data classes. The SVM toolbox, available from [44], was used in the experiment.

For the ORL data set, the number of training images was varied from three to six per subject and the remaining seven to four were used for test. In total ten independent implementations with different randomly chosen training/test images were carried out. The same choices of training/test images were used by all the methods to ensure unbiased comparisons.

For the Yale data set, the methods were trained on ten faces and tested on the remaining one of a subject each time. In each test, test faces had the same facial expression or lighting condition. Eleven implementations were conducted throughout the whole data set

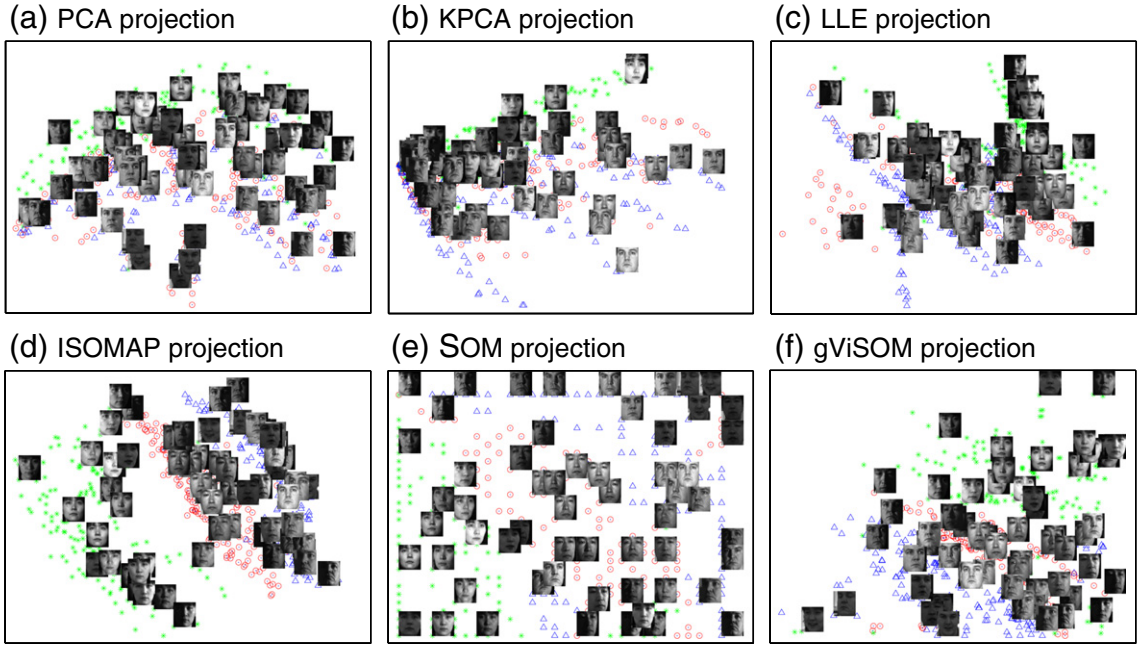

Fig. 4. 2D representations of three subjects from the PIE set. The optimal parameters were: $\sigma_{gaus} = 0.15$ for KPCA, $K_{lle} = 16$, $K_{ISOMAP} = 16$ and $\lambda_{gViSOM} = 0.05$.

corresponding to eleven different facial expressions and lighting conditions.

On the AR data set, 600 faces of three expressions were used for training and the remaining one expression (200 faces) for testing in each implementation. Thus there were totally four implementations.

On the PIE data set, 100 faces from each subject were randomly selected for training and the remaining 70 faces were used for testing in each implementation. Like in the ORL cases, ten independent implementations were carried out for evaluating the performance.

The performances of all these subspace methods were investigated on the same classifier in each experiment and on the same number of reduced dimension. In many cases, good performance of nonlinear subspace methods heavily relies on the choice of their parameters. The parameters of each method in the experiments were experimentally chosen to yield the optimal result for that method. PCA, KPCA, LLE, LPP, ISOMAP and CCA were implemented as *vector-based subspace method*, in which images are represented by single vectors; while 2DPCA, BPCA and SOM-based methods were implemented in the so-called *sub-block based method* [28,29,17], where an image is represented by a number of sub-blocks.

The results (classification rates) are the mean results of these independent implementations. The standard deviations are also calculated, together with the significance *t*-test results (p-values) between the best performer and the others.

3.3.1. Vector-based subspace learning

Each face image was vectorized (2576, 2700, 2200 and 1024 dimensions for ORL, Yale, AR and PIE, respectively). Six methods, PCA, KPCA, LPP, LLE, ISOMAP and CCA, were implemented to reduce the dimensions of the images. The classification was conducted by using the NN, soft *k*-NN, LDA or SVM classifier. The Gaussian function was used for the kernel reconstruction in the KPCA. Cosine weight model was used in LPP for constructing affinity graph. For optimal parameter selection, we varied the radius of the Gaussian function (σ_{gaus}) for KPCA, the size of neighborhood (K_{lle}) for LLE, the radius of neighborhood (ϵ_{iso}) for ISOMAP and the number of epochs (e_{cca}) for CCA in a fixed number of reduced dimensions set empirically. The optimal parameters used are: $\sigma_{gaus}=15$, 30, 10, 20, $K_{lle}=24$, 36, 36, 84, $\epsilon_{iso}=17$, 30, 20, 25 and $e_{cca}=30$, 70, 10, 20, for ORL, Yale, AR and PIE databases, respectively. The representative number of reduced dimensions (*RDim*) used for four databases are 50, 50, 140 and 100. The resulting best classification rates of these methods are compiled in Table 1. Note that the same number of reduced dimension was used by all the methods in each data set and these numbers were selected by trading off between information (high classification rate) and compactness (few dimensions) for representation.

The results in Table 1 show that similar performances were obtained by these subspace methods with various classifiers (with the best classification rates marked in bold in each case) on the ORL except LPP,

Table 1
Correct classification rates of vector-based subspace methods on ORL, Yale, AR and PIE databases. Mean rates of independent runs, standard deviations and *t*-test p-values (in brackets) between the best performer and the others.

	Classification rate (in %) ± standard deviation (p-value)					
	PCA	KPCA	LPP	LLE	ISOMAP	CCA
Training faces (#)	ORL database					
	NN classifier					
3	87.07 ± 2.50 (0.12)	86.71 ± 2.27 (0.063)	86.25 ± 2.57 (0.035)	**88.36 ± 2.33** (– –)	88.32 ± 2.99 (0.49)	87.46 ± 2.69 (0.22)
4	92.21 ± 1.88 (0.44)	91.63 ± 1.79 (0.19)	90.33 ± 2.05 (0.015)	92.29 ± 1.33 (0.48)	92.29 ± 1.83 (0.48)	**92.33 ± 1.72** (– –)
5	**94.45 ± 1.27** (– –)	94.35 ± 1.28 (0.43)	91.65 ± 2.25 (**1.4e−3**)	94.15 ± 1.15 (0.29)	94.20 ± 1.46 (0.34)	94.40 ± 1.54 (0.47)
6	96.44 ± 0.64 (0.37)	96.44 ± 0.64 (0.37)	92.81 ± 0.94 (**9.7e−9**)	95.81 ± 1.06 (0.051)	96.06 ± 0.85 (0.11)	**96.56 ± 0.88** (– –)
	Soft kNN classifier					
3	85.71 ± 1.79 (**5.9e−3**)	85.93 ± 2.31 (0.015)	85.11 ± 2.61 (**4.2e−3**)	**88.46 ± 2.49** (– –)	87.11 ± 2.75 (0.13)	86.71 ± 2.50 (0.067)
4	91.17 ± 1.67 (0.024)	90.79 ± 1.88 (0.012)	89.42 ± 2.22 (**4.7e−4**)	**92.58 ± 1.27** (– –)	91.25 ± 1.50 (0.023)	91.46 ± 2.04 (0.081)
5	93.60 ± 2.18 (0.25)	93.75 ± 1.60 (0.27)	90.80 ± 2.54 (**6.9e−4**)	**94.15 ± 1.25** (– –)	93.90 ± 1.50 (0.35)	93.30 ± 1.38 (0.083)
6	95.19 ± 1.06 (0.022)	95.94 ± 0.94 (0.26)	91.81 ± 1.05 (**1.2e−8**)	**96.25 ± 1.13** (– –)	96.19 ± 0.94 (0.45)	95.25 ± 0.68 (0.014)
	LDA classifier					
3	87.25 ± 3.01 (0.01)	88.14 ± 3.27 (0.052)	86.50 ± 2.50 (**1.1e−3**)	**90.32 ± 2.32** (– –)	87.25 ± 2.73 (**7.3e−3**)	85.04 ± 2.57 (**1.7e−5**)
4	**93.63 ± 1.23** (– –)	92.88 ± 1.29 (0.10)	90.46 ± 2.05 (**2.5e−4**)	93.63 ± 1.46 (0.50)	93.04 ± 1.98 (0.22)	92.29 ± 1.68 (0.029)
5	**96.15 ± 1.15** (– –)	95.55 ± 1.35 (0.15)	91.60 ± 1.78 (**8.7e−7**)	95.15 ± 0.95 (0.024)	94.60 ± 1.80 (0.018)	94.15 ± 2.15 (0.011)
6	**97.56 ± 1.19** (– –)	97.44 ± 0.69 (0.39)	93.00 ± 1.03 (**1.1e−8**)	96.94 ± 0.95 (0.11)	96.31 ± 1.69 (0.037)	96.81 ± 1.23 (0.091)
	SVM classifier					
3	89.96 ± 2.89 (0.17)	89.96 ± 2.89 (0.17)	85.92 ± 3.11 (**4.8e−4**)	90.07 ± 2.57 (0.18)	**91.25 ± 3.00** (– –)	88.93 ± 3.21 (0.058)
4	94.96 ± 1.47 (0.24)	94.88 ± 1.40 (0.20)	90.17 ± 2.23 (**2.5e−6**)	94.46 ± 1.68 (0.093)	**95.42 ± 1.42** (– –)	94.25 ± 1.43 (0.042)
5	96.70 ± 1.42 (0.43)	96.75 ± 1.10 (0.46)	91.65 ± 2.39 (**3.6e−6**)	96.15 ± 1.05 (0.10)	**96.80 ± 1.18** (– –)	95.45 ± 1.25 (0.012)
6	**97.81 ± 0.56** (– –)	97.69 ± 0.78 (0.35)	92.63 ± 0.75 (**1.8e−13**)	97.19 ± 0.88 (0.040)	97.69 ± 0.94 (0.37)	97.00 ± 0.63 (**3.6e−3**)
Classifier	Yale database					
NN	82.42 ± 17.08 (0.15)	82.42 ± 17.08 (0.15)	**89.70 ± 14.44** (– –)	85.45 ± 15.32 (0.26)	83.64 ± 15.32 (0.18)	83.03 ± 18.07 (0.17)
soft kNN	84.85 ± 15.98 (0.22)	84.24 ± 13.22 (0.17)	**89.70 ± 13.22** (– –)	84.85 ± 15.10 (0.22)	84.85 ± 14.77 (0.21)	83.64 ± 15.32 (0.17)
LDA	87.27 ± 17.19 (0.40)	86.67 ± 18.18 (0.37)	**89.09 ± 15.43** (– –)	86.06 ± 17.96 (0.34)	84.85 ± 19.94 (0.29)	85.45 ± 20.17 (0.32)
SVM	84.24 ± 18.51 (0.26)	87.27 ± 15.98 (0.39)	**89.09 ± 15.43** (– –)	86.67 ± 18.18 (0.37)	87.88 ± 13.99 (0.42)	86.67 ± 16.97 (0.36)
Classifier	AR database					
NN	90.75 ± 7.38 (0.28)	90.63 ± 7.31 (0.27)	90.00 ± 12.50 (0.31)	**93.75 ± 6.63** (– –)	90.63 ± 7.81 (0.28)	89.38 ± 8.44 (0.22)
soft kNN	89.00 ± 9.75 (0.24)	89.37 ± 9.69 (0.25)	88.75 ± 13.88 (0.28)	**93.50 ± 6.75** (– –)	90.25 ± 9.13 (0.29)	87.63 ± 11.31 (0.20)
LDA	93.13 ± 8.06 (0.38)	93.00 ± 8.50 (0.37)	90.00 ± 12.75 (0.26)	**94.75 ± 5.88** (– –)	91.63 ± 10.56 (0.31)	84.25 ± 17.38 (0.15)
SVM	93.63 ± 7.31 (0.39)	93.88 ± 7.19 (0.41)	90.25 ± 12.38 (0.25)	**95.00 ± 5.75** (– –)	94.50 ± 6.75 (0.46)	92.75 ± 8.63 (0.34)
Classifier	PIE Database					
NN	90.13 ± 0.51 (**0.0**)	89.50 ± 0.58 (**0.0**)	**95.76 ± 0.27 (– –)**	94.28 ± 0.44 (**1.2e−8**)	90.22 ± 0.55 (**0.0**)	92.11 ± 0.45 (**2.8e–15**)
soft kNN	**95.70 ± 0.27 (– –)**	95.00 ± 0.46 (**2.8e−4**)	95.48 ± 0.27 (0.42)	94.24 ± 0.42 (**9.1e−9**)	95.60 ± 0.33 (0.23)	91.84 ± 0.49 (**3.2e−15**)
LDA	97.40 ± 0.21 (0.041)	**97.55 ± 0.15** (– –)	96.03 ± 0.31 (**9.7e−12**)	94.77 ± 0.43 (**3.0e−14**)	97.37 ± 0.15 (**7.4e−3**)	96.96 ± 0.11 (**2.5e−9**)
SVM	97.31 ± 0.17 (**1.0e−7**)	97.50 ± 0.20 (**8.5e−5**)	96.05 ± 0.28 (**8.1e−14**)	95.18 ± 0.36 (**2.9e−15**)	**97.86 ± 0.14** (– –)	97.73 ± 0.18 (0.044)

which has poor performance. PCA yields similar or comparable performance to most of the nonlinear methods. While the performances vary across the subspace methods and classifiers used, the differences among them in most case on these two data sets are small and most are statistically insignificant — as indicated by the p-values (in brackets). Only in one case on the ORL data set with three training images and the soft k-NN classifier, significant improvement ($p<0.01$, marked in bold) can be claimed, where PCA is the significantly worst ($p=5.9e-3$), while other nonlinear methods are indifferent from the best performing LLE. But with the LDA classifier, PCA is not overwhelmingly inferior ($p=0.01$), while ISOMAP and CCA are ($p=7.3e-3$, $p=1.7e-5$).

LPP and LLE consistently have the highest classification rates on the Yale and AR data set respectively. However, the margins between the best classification rates and others are small and statistically insignificant to make a general claim of its superiority. PCA again has similar performance with most of other nonlinear methods.

On the PIE database, though the performance of these methods in many cases are statistically significant compared with corresponding best rates, the results obtained are varied considerably between classifiers. PCA has the highest rate by using the soft$k$-NN while LPP, KPCA and ISOMAP performed best in NN, LDA and SVM classifiers, respectively. The rates of PCA in NN and SVM are statistically worse than the methods of top performance, but they are still comparable or better than some other nonlinear methods. Again, it is difficult to identify a single method which in general has best performance in this database.

From Table 1, it can be easily observed that the performance of nonlinear methods fluctuate largely among different databases. For example, LPP has the best performance in the Yale database, but its classification rates are the lowest in the ORL data set. While PCA presents much more stable performance than the nonlinear methods.

On computational complexity, linear PCA for extracting a d-dimensional subspace from a data set of N points in a n-dimensional input space is only $O(dnN)$ by computing the first d eigenvectors via SVD decomposition of data matrix, $\mathbf{D}\in\mathbb{R}^{n\times N}$, $d\ll n$. However, the computational cost of nonlinear methods, for instance LLE, is $O(nN^2+nK_{lle}^3N+dN^2)$, where $O(nN^2)$ for finding K_{lle} nearest neighbors for every point, $O(nK_{lle}^3N)$ for computing the weight matrix via solving a constrained least square problem and $O(dN^2)$ for computing the d-dimensional output vectors by solving a eigenvector problem. In ISOMAP, the total computation is $O(nN^2+N^3+dN^2)$, for constructing neighborhood graph, computing shortest paths by Floyd's algorithm and constructing d-dimensional embedding, respectively. As to be analyzed in Section 4, in most subspace learning, the dimensions (d) of the learned subspace are often far less than the numbers of data points, N (e.g. $d=50$ and $N=400$ in ORL). Hence, the computation of PCA is even far lighter than the first step of LLE or ISOMAP algorithm for constructing their neighborhood graphs, $O(dnN)\ll O(nN^2)$, while the computation of the last two steps of LLE and ISOMAP further increase dramatically with the number of training data points. In addition, the performance of these nonlinear methods varies with the choices of their parameters and in some case the variations are great. A good performance can only be achieved after certain parameter optimization processes, which are always time-consuming along with the high computational cost of the algorithms themselves.

3.3.2. Block-based subspace learning

2DPCA, BPCA and SOM-based methods were implemented as block-based subspaces for dimensionality reduction. The ORL faces are used here as examples for describing the processing of the block-based subspace learning. Each image was first locally sampled by moving a window of size 5×5 (block size, S_{blk}) over the entire image by shifting 4 pixels (block distance) each time, giving $23\times28=644$ blocks in total. That is, each face image was represented by a matrix having 644 25-dimensional vectors. These 25-dimensional vectors were used as the input for SOM-based methods. The details of the training algorithms are described in Section 2. After training, all 25-dimensional vectors in each image were passed through the trained map, and represented by the 2-dimensional index values of the corresponding winners on the map. Thereby, from the trained map, each face image generates two feature faces, each being reconstructed from one of the two indices, as shown in Fig. 5. These feature faces were used for the classification. For example, each feature face of an ORL face is of 23×28 in size, which resembles a reduced face image. In BPCA, each image block, 25-dimensional vector, of a face was projected onto 2-dimensional subspace, providing similarly two feature faces. 2DPCA was implemented directly on raw face image matrix, and each face was represented by a matrix with reduced number of columns (e.g. $RD_{2dpca}=5$ in the ORL data set with reduced dimensions of $92\times5=410$). Finally, classification, conducted on these face representations, was performed by the NN, soft k-NN and SVM classifiers.

The parameter selection for block-based methods was also conducted experimentally similar to the previous case. The 2DPCA used the numbers of reduced columns (RD_{2dpca}) from 3 to 40, and the optimal parameters were set as: $RD_{2dpca}=5$, 7, 13, 15 for the ORL, Yale, AR and PIE databases, respectively. SOM-based methods varied the map size (M_{som}) from 5×5 to 50×50; ViSOM and gViSOM also varied the value of resolution parameter (λ, defined in Section 2.8). The optimal parameters were set to $M_{som}=30\times30$, 13×13, 18×18, 15×15, $\lambda_{visom}=0.35$, 0.90, 0.50, 0.18 and $\lambda_{gvisom}=0.48$, 0.65, 0.56, 0.40, for the ORL, Yale, AR and PIE databases, respectively. In these block-based methods (excluding 2DPCA), the number of retained dimensions (RD_{blk}) is proportional to the total numbers of sub-blocks in an image which is in inverse ratio to the size of sub-block ($S_{blk}\times S_{blk}$), and it is computed as

$$RD_{blk}=ceil\left(\frac{N_{row}}{S_{blk}-1}\right)\times ceil\left(\frac{N_{col}}{S_{blk}-1}\right)\times 2 \tag{17}$$

where $ceil(x)$ returns a minimum integer that is equal or larger than x. N_{row} and N_{col} denote the height and width of an image. In the ORL

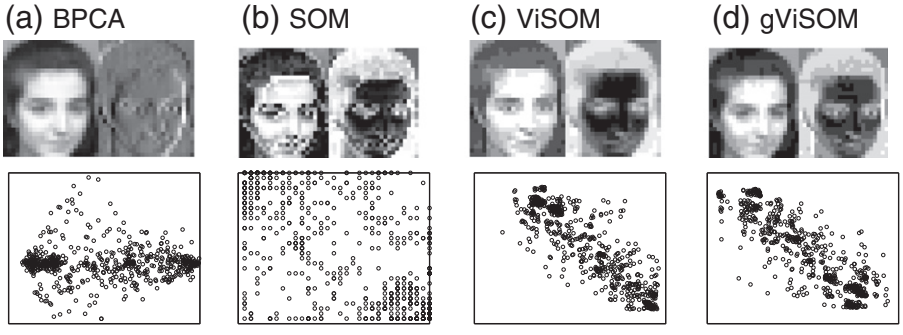

Fig. 5. Examples of feature faces (top) and projections of trained maps (bottom). Left and right feature faces correspond to the x- and y-axes of the projection.

Table 2
Correct classification rates of block-based subspace methods on ORL, Yale, AR and PIE databases. Mean rates of independent runs, standard deviations and *t*-test p-values (in brackets) between the best performer and the others.

	Classification rate (in %) ± standard deviation (p-value)				
	2DPCA	BPCA	SOM	ViSOM	gViSOM
Training faces (#)	ORL database				
	NN classifier				
3	**89.17 ± 2.99** (– –)	86.04 ± 2.46 (0.010)	86.89 ± 2.89 (0.050)	88.86 ± 2.43 (0.40)	88.64 ± 2.64 (0.34)
4	**93.67 ± 1.50** (– –)	90.71 ± 1.63 (**2.6e−4**)	91.45 ± 1.75 (**3.5e−3**)	93.29 ± 1.54 (0.29)	92.96 ± 1.72 (0.17)
5	**95.65 ± 1.35** (– –)	92.75 ± 1.10 (**3.0e−5**)	93.35 ± 1.28 (**5.2e−4**)	95.40 ± 1.20 (0.33)	95.10 ± 1.22 (0.18)
6	97.12 ± 0.81 (0.49)	93.83 ± 0.60 (**1.3e−10**)	95.00 ± 0.75 (**9.1e−7**)	**97.13 ± 0.58** (– –)	96.75 ± 0.65 (0.092)
	Soft kNN classifier				
3	90.93 ± 2.86 (0.051)	89.18 ± 2.99 (**3.7e−3**)	92.36 ± 2.49 (0.26)	93.04 ± 2.93 (0.47)	**93.14 ± 2.87** (– –)
4	94.63 ± 1.13 (**1.4e−3**)	93.71 ± 1.46 (**1.3e−4**)	95.75 ± 1.33 (0.10)	96.50 ± 1.17 (0.47)	**96.54 ± 1.32** (**– –**)
5	96.35 ± 1.15 (**2.4e−3**)	94.95 ± 0.67 (**1.8e−7**)	97.00 ± 1.10 (0.038)	97.40 ± 0.84 (0.13)	**97.85 ± 0.91** (– –)
6	97.81 ± 0.69 (**1.7e−4**)	96.38 ± 1.13 (**4.1e−6**)	98.31 ± 0.73 (**7.4e−3**)	98.94 ± 0.48 (0.21)	**99.16 ± 0.68** (– –)
	SVM classifier				
3	90.14 ± 3.40 (0.29)	88.75 ± 3.61 (0.082)	89.61 ± 2.96 (0.17)	90.57 ± 3.46 (0.40)	**90.96 ± 3.18** (– –)
4	94.54 ± 1.62 (0.20)	93.83 ± 1.83 (0.053)	94.08 ± 2.07 (0.10)	95.04 ± 2.20 (0.43)	**95.21 ± 1.79** (– –)
5	96.15 ± 1.45 (0.14)	95.85 ± 1.28 (0.051)	96.00 ± 1.60 (0.11)	96.60 ± 1.48 (0.35)	**96.85 ± 1.31** (– –)
6	97.38 ± 0.88 (0.086)	97.25 ± .88 (0.044)	97.38 ± 0.90 (0.089)	97.65 ± 0.75 (0.24)	**97.87 ± 0.63** (– –)
Classifier	Yale database				
NN	85.45 ± 16.52 (0.43)	83.03 ± 18.62 (0.32)	84.24 ± 18.51 (0.38)	83.64 ± 16.86 (0.34)	**86.67 ± 15.76** (– –)
soft kNN	84.85 ± 18.40 (0.39)	85.45 ± 15.10 (0.35)	86.06 ± 15.43 (0.39)	**87.88 ± 12.56** (– –)	87.27 ± 12.56 (0.46)
SVM	86.67 ± 18.18 (0.38)	84.85 ± 21.16 (0.31)	88.48 ± 19.17 (0.47)	**89.09 ± 15.21** (– –)	87.27 ± 17.19 (0.40)
Classifier	AR database				
NN	91.38 ± 7.44 (0.086)	91.13 ± 7.81 (0.081)	**95.37 ± 4.69** (– –)	92.13 ± 7.06 (0.12)	92.00 ± 7.00 (0.11)
soft kNN	95.63 ± 3.81 (0.013)	97.88 ± 1.44 (0.043)	98.75 ± 1.13 (0.39)	98.75 ± 0.88 (0.38)	**98.88 ± 0.94** (– –)
SVM	94.00 ± 7.00 (0.37)	94.50 ± 6.25 (0.43)	95.00 ± 6.25 (0.50)	**95.00 ± 5.75** (– –)	95.00 ± 6.00 (0.50)
Classifier	PIE database				
NN	**91.26 ± 0.48** (– –)	79.98 ± 0.85 (**0.0**)	87.50 ± 0.72 (**1.3e−11**)	87.75 ± 0.70 (**3.0e−11**)	86.00 ± 0.55 (**1.4e−15**)
soft kNN	**93.58 ± 0.48** (– –)	79.38 ± 0.85 (**0.0**)	87.64 ± 0.42 (**0.0**)	85.12 ± 0.58 (**0.0**)	85.14 ± 0.42 (**0.0**)
SVM	**97.99 ± 0.14** (– –)	94.04 ± 0.42 (**0.0**)	93.82 ± 0.52 (**4.4e−16**)	97.23 ± 0.29 (**2.3e−7**)	96.63 ± 0.41 (**3.0e−9**)

case, the reduced dimensions by these methods are still large, $23 \times 28 \times 2 = 1288$.

The size of sub-blocks varied from 4 to 21, and $S_{blk} = 5$ was found optimal for these databases. The corresponding classification results are shown in Table 2. Again, these results are the mean rates, standard deviations and significance *t*-test p-values.

It can be seen that SOM-based methods perform better than PCA-based methods in most cases. Table 2 shows that 2DPCA is only better with the NN classifier with five training images or less on the ORL set, while ViSOM or gViSOM have the best performance on the remaining ORL cases and all cases of the Yale set. On the ORL, gViSOM based on soft *k*-NN classifier considerably outperformed 2DPCA, BPCA and SOM methods in term of classification rates, reaching as high as 99.16% in the case of six training faces per subject. The p-values indicate that in these cases, the ViSOM or gViSOM are significantly better. But the improvements of ViSOM/gViSOM on the Yale set are not significant over the other methods. For the AR database, SOM has comparable performance with ViSOM/gViSOM, though the differences among various methods are not significant enough. 2DPCA performs significantly better than other methods on the PIE database, while the rates of BPCA are significantly lower than SOM-based methods.

In summary, in most cases, ViSOM-based methods have shown their superiority for feature extraction. This can also be observed from their feature faces presented in Fig. 5, where the feature images projected by the ViSOM/gViSOM resemble better the original image than SOM or PCA-based methods due to its metric preserving property. However, the performance differences are small or marginal in many cases.

3.3.3. Discussions

For performance comparison between vector-based and block-based subspace methods, one can see that the block-based methods seem to perform marginally better than the vector-based methods in the term of classification rates, where considerable improvements were achieved with the soft *k*-NN classifier on the ORL and AR data sets, though they are not distinctive in the NN and SVM classifier cases. However, the vector-based methods require much smaller numbers of reduced dimensions (e.g. only 50, 50, 140 and 100 retained dimensions for ORL, Yale, AR and PIE faces respectively). While the block-based methods seem to capture better face features by using local sampling, but they requires larger numbers of retained dimensions (e.g. 1288, 9760, 2520 and 128 dimensions for the ORL, Yale, AR and PIE data respectively, computed by Eq. (17)). In addition, large numbers of retained dimensions of the block-based methods can cause singularity problem when applying LDA [45–48], as the number of reduced dimensions may be greater than the numbers of face images. From the extensive experiments and nonlinear analysis in Section 4, one observation can be reached is that the numbers of reduced dimensions used by vector-based methods are mainly determined by the total numbers of training data, while these numbers in block-based methods largely depend on the sizes of face images.

For computation, SOM-based methods have the complexity of $O(M_{som}nN)$ for both training and mapping input vectors onto the trained map. M_{som} is the total number of notes in SOM-based maps. This computational cost is similar or a slightly higher than that of linear PCA. ViSOM and gVISOM have a few times higher computational cost than the SOM. An important virtue of SOM-based methods is that they work in adaptive fashion for updating the weights to an

approximated solution, which can be easily applied for online learning model where data is presented sequentially. The PCA-based methods can only operate in a batch mode.

4. Nonlinearity analysis

The nonlinear analysis in this section only discusses on the holistic representation of face data, as implemented by the vector-based subspace methods. Block-based subspace learning using component-based features, represents a single face as multiple input points (each facial component or block corresponds to a point), which lie in different locations with large distances between them. Thus nonlinear analysis of the structure of these component-based data cannot reveal the true underlying structure of face data and is completely different with holistic representation.

It is obvious that a data set containing N data points is linear in an $N-1$ or higher dimensional space or subspace. In the ORL, Yale and AR face databases, the dimension of face data is much greater than the number of images. Thus face data are completely linear in their input spaces. For example, ORL data set containing 400 face images is linear in 399 or higher dimensional subspaces. However, are they still linear in a further reduced subspace, such as in a 50-dimensional subspace used in previous experiments? If not, how nonlinear are they? A PCA-based method is used to measure the degree of nonlinearity of data. The nonlinearity rate (*NLR*) can be computed as [14],

$$NLR = 1 - \sum_{k=1}^{d} \lambda_k / \sum_{i=1}^{n} \lambda_i \tag{18}$$

where λ_k is the largest kth eigenvaluse of the scatter matrix $S = \sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})\left(\mathbf{x}_i - \overline{\mathbf{x}}^t\right)$, $\mathbf{x}_i$ is a data point in the n-dimensional input space and $\overline{\mathbf{x}}$ is the mean vector of total data points (the number of data points is N). *NLR* is the nonlinearity rate computed in a d-dimensional subspace.

This nonlinearity measure has been tested on several artificial data sets and it is found that a data set has high degree of nonlinearity when the value of *NRL* is higher than 0.3. The nonlinearities of the ORL (400 faces), Yale (165 faces), AR (800 faces) and PIE (5780 faces) in reduced subspaces with dimensions varied from 5 to 200 are plotted in Fig. 6. The *NLR* values show that the ORL face data has higher degree of nonlinearity than the Yale, AR and PIE faces in the subspaces with same numbers of dimensions. Note, though the PIE set has a much larger number of images than other databases, its *NLR* values are lower due to smaller variations in the images and much smaller size. The *NLR* values of the ORL, AR, Yale and PIE data sets in 50-dimensional subspaces are about 0.18, 0.14, 0.05 and 0.05 respectively. The *NLR* of all data sets becomes 0.1 or lower in higher than 100-dimensional subspace, indicating that these data sets have low degrees of nonlinearity in the reduced subspaces (i.e. are fairly linearly distributed). It also explains why nonlinear methods have similar performance to PCA in these subspaces.

From our extensive experiments on vector-based face recognition with various reduced dimensions in the ORL, Yale and AR databases shown in Fig. 7, three observations can be drawn. First, the performance of all methods increases in general with the retained dimensions, as lower nonlinearities lead to better performance. Second, PCA yields reasonable performances at reduced dimensions of as low as 40 on the ORL and Yale (with *NLR*s of 0.21 and 0.06, respectively), and of 80 on the AR (with *NLR* of 0.10). With the increase in reduced dimensions (decrease in nonlinearity), all methods have stable performance with the NN classifier, while PCA yields similar or comparable performance to those nonlinear methods in most cases. Third, the performances of all methods deteriorate in highly reduced subspaces or very low dimensions (thus highly nonlinear). For example, on the ORL data set, all methods have low classification rates in the subspaces of less than 20 dimensions, of which the *NLR* values are higher than 0.3. In the viewpoint of feature extraction, a compact yet informative representation of facial images is desirable, while the representation should also lie in subspaces having low degree of nonlinearity where dimensionality reduction methods, linear or nonlinear, can perform effectively.

5. Summary and conclusions

In this paper, we investigated the unsupervised linear and nonlinear subspace or dimensionality reduction methods for facial image representation and recognition. Their performances have been systemically evaluated and compared on synthetical data and a range of real-world facial image data sets independent on subsequent classification. The main findings are summarized below.

Linear vs *nonlinear*: nonlinear methods have their flexibility in learning the structure of a continuous and smooth data set with a large number of samples, which lies in a single subspace embedding into a high-dimensional input space. But they fail to consistently outperform linear methods on real-world data sets having more complicated distributions, such as multi-subspace, discrete distributions and sampling effects. In a subspace of low nonlinearity, similar or comparable performance can be expected from either linear or nonlinear methods.

Vector-based vs *block-based*: block-based subspace methods have their advantages for feature representation as they capture more features than vector-based methods by using local block sampling, and

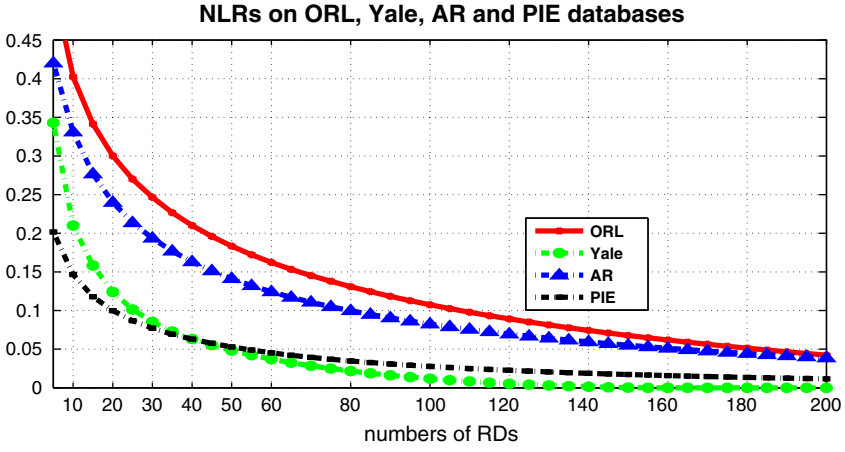

Fig. 6. *NLR* values of ORL, Yale and AR face databases.

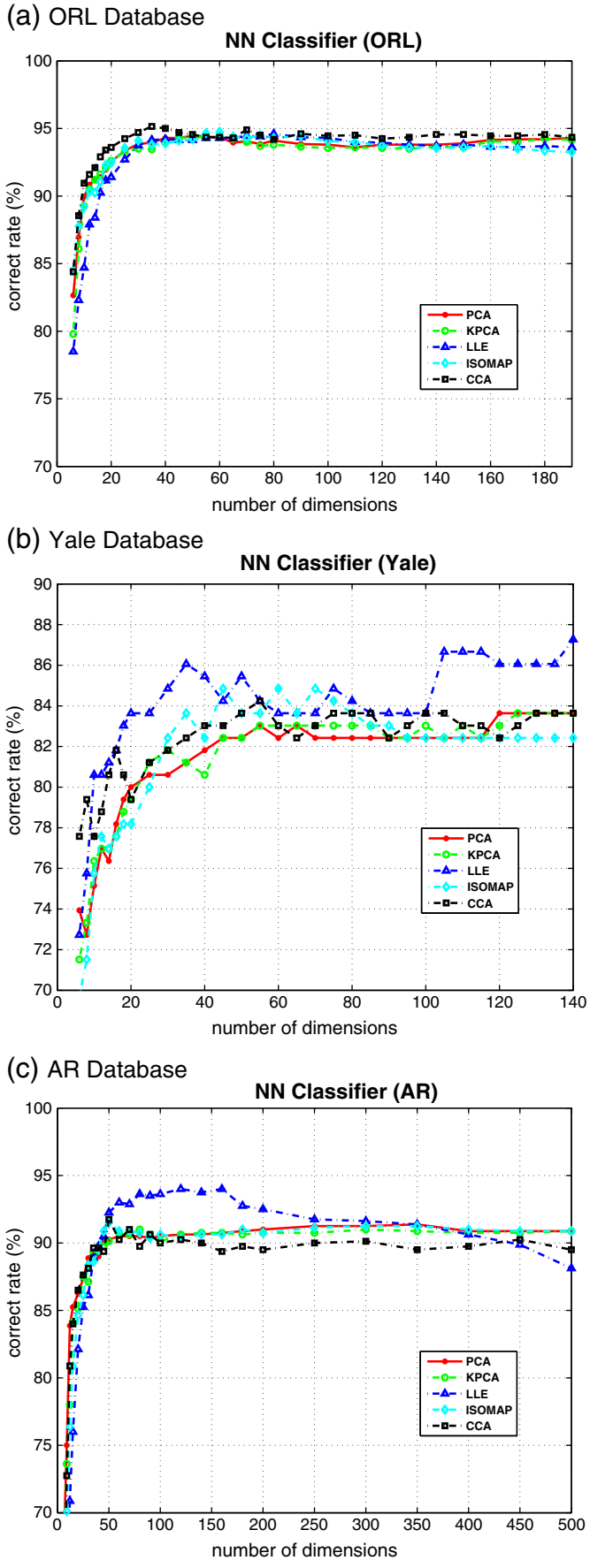

Fig. 7. Classification rates of vector-based methods (with the NN classifier) against various reduced dimensions on the ORL, Yale and AR databases.

metric preserving mapping (e.g. ViSOM) further improves such performance. However, block-based methods often require more retained dimensions than vector-based methods; and this increases computational costs and storages.

PCA-based vs *adaptive neural networks*: PCA/eigen-decomposition based methods can be formulated as a closed-form optimization problem where a global solution is guaranteed. While adaptive neural networks often achieve an approximated solution by updating their free parameters or weights iteratively via minimizing a locally set cost function, and usually no unique solution can be guaranteed. However, eigen-based methods are not adaptive and thus unsuitable for cases where online learning is required.

The main aim of the work was to comprehensively investigate the performance of linear and nonlinear dimensionality reduction methods for face representation and recognition. Nonlinear subspace methods can easily demonstrate their virtue on artificial nonlinear data. On real-world facial image data, the difficulty stems from the existence of multiple subjects, limited number of training samples and great variability in individual appearance, further coupled with variations in lighting, expression and background. Thus single nonlinear subspace or manifold struggles to provide a more effective, convincing alternative to the simple, efficient linear PCA for representing and classifying these complex data, even with much added computational expenses.

References

[1] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1991) 71–86.
[2] P. Bellhumer, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell., 19, 1997, pp. 711–720.
[3] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1994, pp. 84–91.
[4] A. Ahonen, A. Hadid, M. Pietikäainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 2037–2041.
[5] W. Huang, H. Yin, A dissimilarity kernel with local features for robust facial recognition, Proc. IEEE Int. Conf. on Image Processing, 2010, pp. 3785–3788.
[6] R. Duda, P. Hart, D. Stork (Eds.), Pattern Classification, 2nd edition, Wiley, New York, USA, 2001.
[7] B. Scholköpf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[8] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[9] J. Tenenbaum, V. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[10] P. Demartines, J. Hérault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets, IEEE Trans. Neural Netw. 8 (1997) 148–154.
[11] X. He, P. Niyogi, Locality preserving projections, Advances in Neural Information Processing Systems (NIPS), 2003.
[12] D. Cai, X. He, J. Han, H.-J. Zhang, Orthogonal laplacian faces for face recognition, IEEE Trans. Image Process. 15 (2006) 3608–3614.
[13] E. Kokiopoulou, J. Chen, Y. Saad, Trace optimization and eigenproblems in dimension reduction methods, Numer. Linear Algebra Appl. 18 (2011) 565–602.
[14] W. Huang, H. Yin, Linear and nonlinear dimensionality reduction for face recognition, Proc. IEEE Int. Conf. on Image Processing, 2009, pp. 3337–3340.
[15] A. Talukder, D. Casasent, A closed-form neural network for discriminatory feature extraction from high-dimensional data, Neural Netw. 14 (2001) 1201–1218.
[16] Y. Goldberg, A. Zakai, D. Kushnir, Y. Ritov, Manifold learning: the price of normalization, J. Mach. Learn. Res. 9 (2008) 1909–1939.
[17] H. Yin, W. Huang, Adaptive nonlinear manifolds and their applications to pattern recognition, Inf. Sci. 180 (2010) 2649–2662.
[18] J. Wang, B. Zhang, M. Qi, J. Kong, Linear discriminant projection embedding based on patches alignment, Image Vis. Comput. 28 (2010) 1624–1636.
[19] A.M. Martínez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 228–233.
[20] M. Yang, Extended isomap for pattern classification, Proc. National Conf. on Artificial Intelligence, 2002, pp. 224–229.
[21] M. Yang, Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods, Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition, 2002, pp. 215–220.
[22] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 328–340.
[23] Y. Pang, A. Teoh, E. Wong, F. Abas, Supervised locally linear embedding in face recognition, Proc. Int. Symp. on Biometrics and Security Technologies, 2008, pp. 1–6.

[24] L. Han, X. Wu, W. Liang, G. Hou, Y. Jia, Discriminative human action recognition in the learned hierarchical manifold space, Image Vis. Comput. 28 (2010) 836–849.
[25] T. Kohonen (Ed.), Self-Organizing Maps, 2nd edition, Springer, Berlin, Germany, 1997.
[26] H. Yin, ViSOM—a novel method for multivariate data projection and structure visualization, IEEE Trans. Neural Netw. 13 (2002) 237–243.
[27] H. Yin, On multidimensional scaling and the embedding of self-organising maps, Neural Netw. 21 (2008) 160–169.
[28] S. Lawrence, C.L. Giles, A. Tsoi, A. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1997) 98–113.
[29] X. Tan, S. Chen, Z. Zhou, F. Zhang, Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft *k*-nn ensemble, IEEE Trans. Neural Netw. 16 (2005) 875–886.
[30] H. Yin, Nonlinear dimensionality reduction and data visualization: a review, Int. J. Autom. Comput. 3 (2007) 294–303.
[31] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 40–51.
[32] E. Murphy-Chutorian, M. Trivedi, Head pose estimation in computer vision: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 607–626.
[33] M. Yang, Face recognition using kernel eigenfaces, Proc. IEEE Int. Conf. on Image Processing, 2000, pp. 37–40.
[34] M. Kim, D. Kim, S. Lee, Face recognition using the embedded HMM with second-order block specific observations, Pattern Recognit. 36 (2003) 2723–2735.
[35] J. Yang, D. Zhang, A.F. Frangi, J. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1–7.
[36] D.L. Donoho, C. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, Proc. Natl. Acad. Sci. 100 (2003) 5591–5596.
[37] E. Kokiopoulou, Y. Saad, Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 2143–2156.
[38] Y. Weiss, Segmentation using eigenvectors: a unifying view, Proc. IEEE Int. Conf. on Computer Vision, 1999, pp. 975–982.
[39] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.
[40] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, Proc. IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142.
[41] A. Martínez, R. Benavente, The AR face database, Tech. Rep. 24, CVC Technical Report, June 1998.
[42] T. Sim, S. Baker, M. Bsat, The cmu pose, illuminlation, and expression database, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1615–1618.
[43] C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167.
[44] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.
[45] S. Ji, J. Ye, Generalized linear discriminant analysis: a unified framework and efficient model selection, IEEE Trans. Neural Netw. 19 (2008) 1768–1782.
[46] J. Lu, K. Plataniotis, A. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, IEEE Trans. Neural Netw. 14 (2003) 117–129.
[47] L. Chen, H. Liao, M. Ko, J. Lin, G. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognit. 33 (2000) 1713–1726.
[48] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data with application to face recognition, Pattern Recognit. 34 (2001) 2067–2070.