Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 26 (2005) 527–532

Pattern Recognition Letters

www.elsevier.com/locate/patrec

2D-LDA: A statistical linear discriminant analysis for image matrix

Ming Li *, Baozong Yuan

Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China

Received 19 August 2004
Available online 18 October 2004

Abstract

This paper proposes an innovative algorithm named 2D-LDA, which directly extracts the proper features from image matrices based on Fisher's Linear Discriminant Analysis. We experimentally compare 2D-LDA to other feature extraction methods, such as 2D-PCA, Eigenfaces and Fisherfaces. And 2D-LDA achieves the best performance.

Keywords: Feature extraction; Image representation; Linear discriminant analysis; Subspace techniques; Face recognition

1. Introduction

Feature extraction is the key to face recognition, as it is to any pattern classification task. The aim of feature extraction is reducing the dimensionality of face image so that the extracted features are as representative as possible. The class of image analysis methods called appearance-based approach has been of wide concern, which relies on statistical analysis and machine learning. Turk and Pentland (1991) presented the well-known Eigenfaces method for face recognition, which uses principal component analysis (PCA) for dimensionality reduction. However, the base physical similarity of the represented images to originals does not provide the best measure of useful information for distinguishing faces from one another (O'Toole, 1993). Belhumeur et al. (1997) proposed Fisherfaces method, which is based on Fisher's Linear Discriminant and produces well separated classes in a low-dimensional subspace. His method is insensitive to large variation in lighting direction and facial expression.

Recently, Yang (2002) investigated the Kernel PCA for learning low dimensional representations for face recognition and found that the Kernel methods provide better representations and achieve lower error rates for face recognition. Bartlett et al. (2002) proposed using ICA for face representation, which is sensitive to the high-order

* Corresponding author. Tel.: +86 10 5168 3149; fax: +51 68 6168 8616.
E-mail address: liming@mail.edu.cn (M. Li).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.09.007

statistics. This method is superior to representations based on PCA for recognizing faces across days and changes in expression. However, Kernel PCA and ICA are both computationally more expensive than PCA. Weng et al. (2003) presented a fast method, called candid covariance-free IPCA (CCIPCA), to obtain the principal components of high-dimensional image vectors. Moghaddan (2002) compared the Bayesian subspace method with several PCA-related methods (PCA, ICA, and Kernel PCA). The experimental results demonstrated its superiority over PCA, ICA and Kernel PCA.

All the PCA-related methods discussed above are based on the analysis of vectors. When dealing with images, we should firstly transform the image matrixes into image vectors. Then based on these vectors the covariance matrix is calculated and the optimal projection is obtained. However, face images are high-dimensional patterns. For example, an image of 112×92 will form a 10304-dimensional vector. It is difficult to evaluate the covariance matrix in such a high-dimensional vector space. To overcome the drawback, Yang proposed a straightforward image projection technique, named as *image principal component analysis* (IMPCA) (Yang et al., 2004), which is directly based on analysis of original image matrices. Different to traditional PCA, 2D-PCA is based on 2D matrices rather than 1D vectors. This means that the image matrix does not need to be converted into a vector. As a result, 2D-PCA has two advantages: easier to evaluate the covariance matrix accurately and lower time-consuming. Liu et al. (1993) proposed an iterative method to calculate the Foley-Sammon optimal discriminant vectors from image matrixes. And he proposed to substitute $D_t = D_b + D_w$ for D_w to overcome the singularity problem. Liu's method was complicate and didn't resolve the singularity problem well.

In this paper, a statistical linear discriminant analysis for image matrix is discussed. Our method proposes to use Fisher linear projection criterion to find out a good projection. This criterion is based on two parameters: the *between-class scatter matrix* and the *within-class scatter matrix*. Because the dimension of between-class and within-class scatter matrix is much low (comparative to number of training samples). So, the problem, that the within-class scatter matrix maybe singular, will be handled. At the same time, the compute-costing is lower than traditional Fisherfaces. Moreover, we discuss about image reconstruction and conduct a series of experiments on the ORL face database.

The organization of this paper is as follows: In Section 2, we propose the idea and describe the algorithm in detail. In Section 3, we compare 2D-LDA with Eigenfaces, Fisherfaces and 2D-PCA on the ORL face database. Finally, the paper concludes with some discussions in Section 4.

2. Two-dimensional linear discriminant analysis

2.1. Principle: The construction of Fisher projection axis

Let $\boldsymbol{A}$ denotes a $m \times n$ image, and $\boldsymbol{x}$ denotes an n-dimensional column vector. $\boldsymbol{A}$ is projected onto $\boldsymbol{x}$ by the following linear transformation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{1}$$

Thus, we get an m-dimensional projected vector $\boldsymbol{y}$, which is called the feature vector of the image $\boldsymbol{A}$.

Suppose there are L known pattern classes in the training set, and M denotes the size of the training set. The jth training image is denoted by an $m \times n$ matrix A_j ($j = 1, 2, \ldots, M$), and the mean image of all training sample is denoted by $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{A}}_i$ $(i = 1, 2, \ldots, L)$ denoted the mean image of class T_i and N_i is the number of samples in class T_i, the projected class is P_i. After the projection of training image onto $\boldsymbol{x}$, we get the *projected feature vector*

$$\boldsymbol{y}_j = \boldsymbol{A}_j\boldsymbol{x}, \quad j = 1, 2, \ldots, M. \tag{2}$$

How do we judge a projection vector $\boldsymbol{x}$ is good? In fact, the total scatter of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors (Turk and Pentland, 1991). From this point of view, we introduced a criterion at first,

$$J(\boldsymbol{x}) = \frac{P_{\mathrm{B}}}{P_{\mathrm{W}}}. \tag{3}$$

There were two parameters

$$P_{\mathrm{B}} = \mathrm{tr}(\boldsymbol{TS}_{\mathrm{B}}), \tag{4}$$

$$P_{\mathrm{W}} = \mathrm{tr}(\boldsymbol{TS}_{\mathrm{W}}), \tag{5}$$

where $\boldsymbol{TS}_{\mathrm{B}}$ denotes the between-class scatter matrix of projected feature vectors of training images, and $\boldsymbol{TS}_{\mathrm{W}}$ denotes the within-class scatter matrix of projected feature vectors of training images. So,

$$\begin{aligned}\boldsymbol{TS}_{\mathrm{B}} &= \sum_{i=1}^{L} N_i(\bar{\boldsymbol{y}}_i - \bar{\boldsymbol{y}})(\bar{\boldsymbol{y}}_i - \bar{\boldsymbol{y}})^{\mathrm{T}} \\ &= \sum_{i=1}^{L} N_i[(\overline{\boldsymbol{A}}_i - \overline{\boldsymbol{A}})\boldsymbol{x}][(\overline{\boldsymbol{A}}_i - \overline{\boldsymbol{A}})\boldsymbol{x}]^{\mathrm{T}},\end{aligned} \tag{6}$$

$$\begin{aligned}\boldsymbol{TS}_{\mathrm{W}} &= \sum_{i=1}^{L} \sum_{\boldsymbol{y}_k \in P_i} (\boldsymbol{y}_k - \bar{\boldsymbol{y}}_i)(\boldsymbol{y}_k - \boldsymbol{y}_i)^{\mathrm{T}} \\ &= \sum_{i=1}^{L} \sum_{\boldsymbol{y}_k \in P_i} [(\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i)\boldsymbol{x}][(\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i)\boldsymbol{x}]^{\mathrm{T}}.\end{aligned} \tag{7}$$

So

$$\begin{aligned}\mathrm{tr}(\boldsymbol{TS}_{\mathrm{B}}) &= \boldsymbol{x}^{\mathrm{T}} \left(\sum_{i=1}^{L} N_i(\overline{\boldsymbol{A}}_i - \overline{\boldsymbol{A}})^{\mathrm{T}}(\overline{\boldsymbol{A}}_i - \overline{\boldsymbol{A}}) \right) \boldsymbol{x} \\ &= \boldsymbol{x}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{B}} \boldsymbol{x},\end{aligned} \tag{8}$$

$$\begin{aligned}\mathrm{tr}(\boldsymbol{TS}_{\mathrm{W}}) &= \boldsymbol{x}^{\mathrm{T}} \left(\sum_{i=1}^{L} \sum_{\boldsymbol{A}_k \in T_i} (\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i)^{\mathrm{T}}(\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i) \right) \boldsymbol{x} \\ &= \boldsymbol{x}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{W}} \boldsymbol{x}.\end{aligned} \tag{9}$$

We could evaluate $\boldsymbol{TS}_{\mathrm{B}}$ and $\boldsymbol{TS}_{\mathrm{W}}$ directly using the training image samples.

So, the criterion could be expressed by

$$J(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{B}} \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{W}} \boldsymbol{x}}, \tag{10}$$

where $\boldsymbol{x}$ is a unitary column vector. This criterion is called *Fisher linear projection criterion*. The unitary vector $\boldsymbol{x}$ that maximizes $J(\boldsymbol{x})$ is called the *Fisher optimal projection axis*. The optimal projection $\boldsymbol{x}_{\mathrm{opt}}$ is chosen when the criterion is maximized, i.e.,

$$\boldsymbol{x}_{\mathrm{opt}} = \arg\max_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}). \tag{11}$$

If $\boldsymbol{S}_{\mathrm{W}}$ is nonsingular, the solution to above optimization problem is to solve the generalized eigenvalue problem (Turk and Pentland, 1991):

$$\boldsymbol{S}_{\mathrm{B}} \boldsymbol{x}_{\mathrm{opt}} = \lambda \boldsymbol{S}_{\mathrm{W}} \boldsymbol{x}_{\mathrm{opt}}. \tag{12}$$

In the above equation, λ is the maximal eigenvalue of $\boldsymbol{S}_{\mathrm{W}}^{-1} \boldsymbol{S}_{\mathrm{B}}$.

The traditional LDA must face to the singularity problem. However, 2D-LDA overcomes this problem successfully. This is because: For each training image, $\boldsymbol{A}_j$ $(j = 1,2,\ldots,M)$, we have $\mathrm{rank}(\boldsymbol{A}_j) = \min(m, n)$.

From (9), we have

$$\begin{aligned}\mathrm{rank}(\boldsymbol{S}_{\mathrm{W}}) &= \mathrm{rank} \left(\sum_{i=1}^{L} \sum_{\boldsymbol{A}_k \in T_i} (\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i)^{\mathrm{T}}(\boldsymbol{A}_k - \overline{\boldsymbol{A}}_i) \right) \\ &\leqslant (M - L) \cdot \min(m, n).\end{aligned} \tag{13}$$

So, in 2D-LDA, $\boldsymbol{S}_{\mathrm{W}}$ is nonsingular when

$$M \geqslant L + \frac{n}{\min(m, n)}. \tag{14}$$

In real situation, (14) is always satisfied. So, $\boldsymbol{S}_{\mathrm{W}}$ is always nonsingular.

In general, it is not enough to have only one Fisher optimal projection axis. We usually need to select a set of projection axis, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$, subject to the orthonormal constraints. That is,

$$\begin{cases} \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d\} = \arg\max J(\boldsymbol{x}) \\ \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j = 0, \quad i \neq j, \ i, j = 1, \ldots, d. \end{cases} \tag{15}$$

In fact, the optimal projection axes, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$, are the orthonormal eigenvectors of $\boldsymbol{S}_{\mathrm{W}}^{-1} \boldsymbol{S}_{\mathrm{B}}$ corresponding to the first d largest eigenvalues.

Using these projection axes, we could form a new *Fisher projection matrix* $\boldsymbol{X}$, which is a $n \times d$ matrix,

$$\boldsymbol{X} = [\, \boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_d \,]. \tag{16}$$

2.2. Feature extraction

We will use the optimal projection vectors of 2D-LDA, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$, for feature extraction. For a given image $\boldsymbol{A}$, we have

$$\boldsymbol{y}_k = \boldsymbol{A} \boldsymbol{x}_k, \quad k = 1, 2, \ldots, d. \tag{17}$$

Then, we have a family of *Fisher feature vectors* $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d$, which formed a $m \times d$ matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d]$. We called this matrix $\boldsymbol{Y}$ as the *Fisher feature matrix* of the image $\boldsymbol{A}$.

2.3. Reconstruction

In the 2D-LDA method, we can use the Fisher feature matrixes and Fisher optimal projection axes to reconstruct a image by following steps.

For a given image $\boldsymbol{A}$, the Fisher feature matrix is $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d]$ and the Fisher optimal projection axes $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d]$, then we have

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}. \tag{18}$$

Because $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ are orthonormal, it is easy to obtain the reconstructed image of $\boldsymbol{A}$:

$$\widetilde{\boldsymbol{A}} = \boldsymbol{Y}\boldsymbol{X}^{\mathrm{T}} = \sum_{k=1}^{d} \boldsymbol{y}_k \boldsymbol{x}_k^{\mathrm{T}}. \tag{19}$$

We called $\widetilde{\boldsymbol{A}}_k = \boldsymbol{y}_k \boldsymbol{x}_k^{\mathrm{T}}$ as a *reconstructed subimage* of $\boldsymbol{A}$, which have the same size as image $\boldsymbol{A}$. This means that we use a set of 2D Fisherfaces to reconstruct the original image. If we select $d = n$, then we can completely reconstruct the images in the training set: $\widetilde{\boldsymbol{A}} = \boldsymbol{A}$. If $d < n$, the reconstructed image $\widetilde{\boldsymbol{A}}$ is an approximation for $\boldsymbol{A}$.

2.4. Classification

Given two images $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ represented by 2D-LDA feature matrix $\boldsymbol{Y}_1 = [\boldsymbol{y}_1^1, \ldots, \boldsymbol{y}_d^1]$ and $\boldsymbol{Y}_2 = [\boldsymbol{y}_1^2, \ldots, \boldsymbol{y}_d^2]$. So the similarity $d(\boldsymbol{Y}_1, \boldsymbol{Y}_2)$ is defined as

$$d(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \sum_{k=1}^{d} \|\boldsymbol{y}_k^1 - \boldsymbol{y}_k^2\|_2, \tag{20}$$

where $\|\boldsymbol{y}_k^1 - \boldsymbol{y}_k^2\|_2$ denotes the Euclidean distance between the two Fisher feature vectors $\boldsymbol{y}_k^1$ and $\boldsymbol{y}_k^2$.

If the Fisher feature matrix of training images are $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_M$ (M is the total number of training images), and each image is assigned to a class T_i. Then, for a given test image $\boldsymbol{Y}$, if $d(\boldsymbol{Y}, \boldsymbol{Y}_l) = \min_j d(\boldsymbol{Y}_1, \boldsymbol{Y}_j)$ and $\boldsymbol{Y}_l \in T_i$, then the resulting decision is $\boldsymbol{Y} \in T_i$.

3. Experiment and analysis

We evaluated our 2D-LDA algorithm on the ORL face image database. The ORL database (http://www.cam-orl.co.uk) contains images of 40 individuals, each person have 10 different images. For some individuals, the images were taken at different times. The facial expression (open or closed eyes, smiling or nonsmiling) and facial details (glasses or no glasses) also vary. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The images were taken with a tolerance for some titling and rotation of the face of up to 20°. Moreover, there is also some variation in the scale of up to about 10%. The size of each image is 92 × 112 pixels, with 256 grey levels per pixel. Five samples of one person in ORL database are shown in Fig. 1. So, we could use the ORL: database to evaluate 2D-LDA's performance under conditions where pose and the size of sample are varied.

Using 2D-LDA, we could project the test face image onto the Fisher optimal projection axis, then we could use the Fisher feature vectors set to reconstruct the image. In Fig. 2, some reconstructed images and the original image of one person were given out. In Fig. 2, the variance d denotes the number of dimension used to map and reconstruct the face image. As observed these images, we could find that the reconstructed images are very like obtained by sample the original image on the spacing vertical scanning line. The reconstructed image $\widetilde{\boldsymbol{A}}$ is more and more like to the original image $\boldsymbol{A}$ as the value of d increased.

We have done an experiment on the ORL database to evaluate performance of 2D-LDA, 2D-

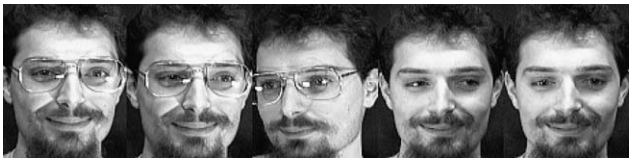

Fig. 1. Five images in ORL database.

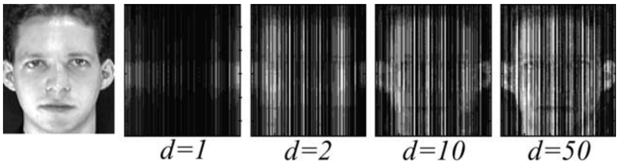

Fig. 2. Some reconstructed images of one person.

PCA (Yang et al., 2004), Eigenfaces (Turk and Pentland, 1991), Fisherfaces (Belhumeur et al., 1997). To evaluate the pure ability of these four in the fair environment, we did not do any preprocess on the face images, and we did not utilize any optimized algorithm. We just realized the algorithm appeared in the literature (Turk and Pentland, 1991; Belhumeur et al., 1997 and Yang et al., 2004) without any modification. In our experiment, we select first five images samples per person for training, and the left five images samples for testing. So, in our experiment, the size of training set and testing set were both 200. So in the 2D-LDA, the size of between-class scatter matrix $\boldsymbol{S}_{\mathrm{B}}$ and within-class scatter matrix $\boldsymbol{S}_{\mathrm{W}}$ are both 92 × 92. Fig. 3 shows the classification result. From Fig. 3, we find that the recognition rate of 2D-LDA have achieved the best performance in the four methods. And the best result of 2D-LDA 94.0% is much better than the best result of 2D-PCA 92.5%. And from Fig. 3, we could find that the two 2D feature extraction methods have outstanding performance in the low-dimension condition, but the conventional ones' ability is very poor.

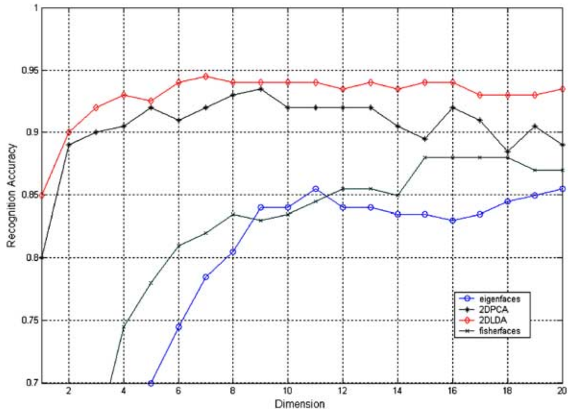

Fig. 3. Comparison of 2D-LDA and 2D-PCA on ORL Database.

Table 1 showed out the comparison of the training time of the four algorithms (CPU: Pentium IV 2.66 GHz, RAM: 256 M). The four algorithms are realized in the Matlab environment. We could see that the 2D-LDA and 2D-PCA's computing-cost is very low compared with Eigenfaces and Fisherfaces. This is because in the 2D condition, we only need to handle a 92 × 92 matrix. But using the Eigenfaces and Fisherfaces, we must face to a 10 304 × 10 304 matrix. It is a hard work. At last, It must be mentioned that when used the Fisherfaces, we must reduced the dimension of image data to avoid that $\boldsymbol{S}_{\mathrm{W}}$ is singular (Belhumeur et al., 1997). Mapping the original data onto how many dimensions space is a hard problem. We must select the proper number of dimension through experiment. Considering this situation, Fisherfaces is very time-costing.

Table 2 showed that the memory cost of 2D feature extraction is much larger than the 1D ones. This is because 2D methods used a $n \times d$ matrix to present a face image. At the same time the 1D techniques reconstructed face images by a d-dimension vector.

Table 1
Comparison of CPU Time (s) for feature extraction using ORL database (15 dimensions)

2D-LDA	2D-PCA	Eigenfaces	Fisherfaces
0.4210	0.4210	28.5000	32.5310

Table 2
Comparison of memory cost (bytes) to present a 92 × 112 image using different techniques (15 dimensions)

2D-LDA	2D-PCA	Eigenfaces	Fisherfaces
6720	6720	60	60

4. Conclusion

In this paper, a new algorithm for image feature extraction and selection was proposed. This method uses the Fisher Linear Discriminant Analysis to enhance the effect of variation caused by different individuals, other than by illumination, expression, orientation, etc. 2D-LDA uses the image matrix instead of the image vector to compute the between-class scatter matrix and the within-class scatter matrix.

From our experiments, we can see that the 2D-LDA have many advantages over other methods. 2D-LDA achieves the best recognition accuracy in the four algorithms. And this technique's computing cost is very low compared with Eigenfaces and Fisherfaces, and close to 2D-PCA. At the same time, this method shows powerful performance in the low dimension. From Fig. 2, we can see that this new projection method is very like to select spacing vertical scanning lines to present a image. Maybe this is the reason that this algorithm is so effective in image classification.

2D-LDA still has its shortcoming. It needs more memory to store a image than the Eigenfaces and Fisherfaces.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 60441002) and the University Key Research Project (No. 2003SZ002).

References

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

Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Machine Intell. 19 (7), 711–720.

Lui, K., Cheng, Y., Yang, J., 1993. Algebraic feature extraction for image recognition based on an optimal discriminant criterion. Pattern Recognition 26 (6), 903–911.

Moghaddan, B., 2002. Principal manifolds and probabilistic subspaces for visual recognition. IEEE Trans. Pattern Anal. Machine Intell. 24 (6), 780–788.

O'Toole, A., 1993. Low-dimensional representation of faces in higher dimensions of the face space. J. Opt. Soc. Amer. 10 (3).

Turk, M., Pentland, A., 1991. Eigenfaces for recognition. J. Cognitive Neurosci. 3 (1), 71–86.

Weng, J., Zhang, Y., Hwang, W., 2003. Candid covariance-free incremental principal component analysis. IEEE Trans. Pattern Anal. Machine Intell. 25 (8), 1034–1040.

Yang, M.H., 2002. Kernel eigenfaces vs. Kernel Fisherfaces: Face recognition using Kernel methods. In: Proc. 5th Internat. Conf. on Automatic Face and Gesture Recognition (RGR'02). pp. 215–220.

Yang, J., Zhang, D., Frangi, A.F., Yang, J.-y., 2004. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Anal. Machine Intell. 26 (1), 131–137.