1

Two-Dimensional Principal Component Analysis and Its Extensions

Parinya Sanguansat
Faculty of Engineering and Technology, Panyapiwat Institute of Management
Thailand

1. Introduction

Normally in Principal Component Analysis (PCA) (Sirovich & Kirby, 1987; Turk & Pentland, 1991), the 2D image matrices are firstly transformed to 1D image vectors by vectorization. The vectorization of a matrix is the column vector obtain by stacking the columns of the matrix on top of one another. The covariance or scatter matrix are formulated from the these image vectors. The covariance matrix will be well estimated if and only if the number of available training samples is not far smaller than the dimension of this matrix. In fact, it is too hard to collect this the number of samples. Then, normally in 1D subspace analysis, the estimated covariance matrix is not well estimated and not full rank.

Two-Dimensional Principal Component Analysis (2DPCA) was proposed by Yang et al. (2004) to apply with face recognition and representation. Evidently, the experimental results in Kong et al. (2005); Yang & Yang (2002); Yang et al. (2004); Zhang & Zhou (2005) have shown the improvement of 2DPCA over PCA on several face databases. Unlike PCA, the image covariance matrix is computed directly on image matrices so the spatial structure information can be preserved. This yields a covariance matrix whose dimension just equals to the width of the face image. This is far smaller than the size of covariance matrix in PCA. Therefore, the image covariance matrix can be better estimated and will usually be full rank. That means the curse of dimensionality and the Small Sample Size (SSS) problem can be avoided.

In this chapter, the detail of 2DPCA's extensions will be presented as follows: The bilateral projection scheme, the kernel version, the supervised framework, the variation of image alignment and the random approaches.

For the first extension, there are many techniques were proposed in bilateral projection schemes such as $2D^2PCA$ (Zhang & Zhou, 2005), *Bilateral 2DPCA (B2DPCA)* (Kong et al., 2005), *Generalized Low-Rank Approximations of Matrices (GLRAM)* (Liu & Chen, 2006; Liu et al., 2010; Ye, 2004), *Bi-Dierectional PCA (BDPCA)* (Zuo et al., 2005) and *Coupled Subspace Analysis (CSA)* (Xu et al., 2004). The left and right projections are determined by solving two eigenvalue problems per iteration. One corresponds to the column direction and another one corresponds to the row direction of image, respectively. In this way, it is not only consider the image in both directions but also reduce the feature matrix smaller than the original 2DPCA.

As the successful of the kernel method in kernel PCA (KPCA), the kernel based 2DPCA was proposed as *Kernel 2DPCA (K2DPCA)* in Kong et al. (2005). That means the nonlinear mapping can be utilized to improve the feature extraction of 2DPCA.

Since 2DPCA is unsupervised projection method, the class information is ignored. To embed this information for feature extraction, the Linear Discriminant Analysis (LDA) is applied in Yang et al. (2004). Moreover, the 2DLDA was proposed and then applied with 2DPCA in Sanguansat et al. (2006b). Another method was proposed in Sanguansat et al. (2006a) based on class-specific subspace which each subspace is constructed from only the training samples in own class while only one subspace is considered in the conventional 2DPCA. In this way, their representation can provide the minimum reconstruction error.

Because of the image covariance matrix is the key of 2DPCA and it is corresponding to the alignment of pixels in image. Different image covariance matrix will obtain the difference information. To produce alternated version of the image covariance matrix, it can be done by rearranging the pixels. The diagonal alignment 2DPCA and the generalized alignment 2DPCA were proposed in Zhang et al. (2006) and Sanguansat et al. (2007a), respectively.

Finally, the random subspace based 2DPCA were proposed by random selecting the subset of eigenvectors of image covariance matrix as in Nguyen et al. (2007); Sanguansat et al. (2007b; n.d.) to build the new projection matrix. From the experimental results, some subset eigenvectors can perform better than others but it cannot predict by their eigenvalues. However, the mutual information can be used in filter strategy for selecting these subsets as shown in Sanguansat (2008).

2. Two-dimensional principal component analysis

Let each image is represented by a m by n matrix $\mathbf{A}$ of its pixels' gray intensity. We consider linear projection of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is an n dimensional projection axis and $\mathbf{y}$ is the projected feature of this image on $\mathbf{x}$, called *principal component vector*.

In original algorithm of 2DPCA (Yang et al., 2004), like PCA, 2DPCA search for the optimal projection by maximize the total scatter of projected data. Instead of using the criterion as in PCA, the total scatter of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors. From this point of view, the following criterion was adopt as

$$J(\mathbf{x}) = tr(\mathbf{S}_{\mathbf{x}}), \tag{2}$$

where

$$\mathbf{S}_{\mathbf{x}} = E[(\mathbf{y} - E\mathbf{y})(\mathbf{y} - E\mathbf{y})^T]. \tag{3}$$

The total power equals to the sum of the diagonal elements or trace of the covariance matrix, the trace of $\mathbf{S}_{\mathbf{x}}$ can be rewritten as

$$\begin{aligned} tr(\mathbf{S}_x) &= tr\{E[(\mathbf{y} - E\mathbf{y})(\mathbf{y} - E\mathbf{y})^T]\} \\ &= tr\{E[(\mathbf{A} - E\mathbf{A})\mathbf{x}\mathbf{x}^T(\mathbf{A} - E\mathbf{A})^T]\} \\ &= tr\{E[\mathbf{x}^T(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A})\mathbf{x}]\} \\ &= tr\{\mathbf{x}^T E[(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A})]\mathbf{x}\} \\ &= tr\{\mathbf{x}^T\mathbf{G}\mathbf{x}\}. \end{aligned} \tag{4}$$

Giving that

$$\mathbf{G} = E[(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A})]. \tag{5}$$

This matrix $\mathbf{G}$ is called *image covariance matrix*. Therefore, the alternative criterion can be expressed by

$$J(\mathbf{x}) = tr(\mathbf{x}^T\mathbf{G}\mathbf{x}), \tag{6}$$

where the image inner-scatter matrix $\mathbf{Gx}$ is computed in a straightforward manner by

$$\mathbf{G} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k - \bar{\mathbf{A}})^T(\mathbf{A}_k - \bar{\mathbf{A}}), \tag{7}$$

where $\bar{\mathbf{A}}$ denotes the average image,

$$\bar{\mathbf{A}} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{A}_k. \tag{8}$$

It can be shown that the vector $\mathbf{x}$ maximizing Eq. (4) correspond to the largest eigenvalue of $\mathbf{G}$ (Yang & Yang, 2002). This can be done, for example, by using the Eigenvalue decomposition or Singular Value Decomposition (SVD) algorithm. However, one projection axis is usually not enough to accurately represent the data, thus several eigenvectors of $\mathbf{G}$ are needed. The number of eigenvectors (d) can be chosen according to a predefined threshold (θ).

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be eigenvalues of $\mathbf{G}$ which sorted in non-increasing order. We select the d first eigenvectors such that their corresponding eigenvalues satisfy

$$\theta \leq \frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{n}\lambda_i}. \tag{9}$$

For feature extraction, Let $\mathbf{x_1}, \ldots, \mathbf{x_d}$ be d selected largest eigenvectors of $\mathbf{G}$. Each image $\mathbf{A}$ is projected onto these d dimensional subspace according to Eq. (1). The projected image $\mathbf{Y} = [\mathbf{y_1}, \ldots, \mathbf{y_d}]$ is then an m by d matrix given by:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \tag{10}$$

where $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_d}]$ is a n by d projection matrix.

2.1 Column-based 2DPCA

The original 2DPCA can be called the row-based 2DPCA. The alternative way of 2DPCA can be using the column instead of row, column-based 2DPCA (Zhang & Zhou, 2005).

This method can be consider as same as the original 2DPCA but the input images are previously transposed. From Eq. (7), replace the image $\mathbf{A}$ with the transposed image $\mathbf{A}^T$ and call it the column-based image covariance matrix $\mathbf{H}$, thus

$$\mathbf{H} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k^T - \bar{\mathbf{A}}^T)^T(\mathbf{A}_k^T - \bar{\mathbf{A}}^T) \tag{11}$$

$$\mathbf{H} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k - \bar{\mathbf{A}})(\mathbf{A}_k - \bar{\mathbf{A}})^T. \tag{12}$$

Similarly in Eq. (10), the column-based optimal projection matrix can be obtained by computing the eigenvectors of $\mathbf{H}(\mathbf{z})$ corresponding to the q largest eigenvalues as

$$\mathbf{Y}_{col} = \mathbf{Z}^T \mathbf{A}, \tag{13}$$

where $\mathbf{Z} = [\mathbf{z_1}, \ldots, \mathbf{z_q}]$ is a m by q column-based optimal projection matrix. The value of q can also be controlled by setting a threshold as in Eq. (9).

2.2 The relation of 2DPCA and PCA

As Kong et al. (2005) 2DPCA, performed on the 2D images, is essentially PCA performed on the rows of the images if each row is viewed as a computational unit. That means the 2DPCA of an image can be viewed as the PCA of the set of rows of an image. The relation between 2DPCA and PCA can be proven that by rewriting the image covariance matrix $\mathbf{G}$ in normal covariance matrix as

$$\begin{aligned}
\mathbf{G} &= E\left[(\mathbf{A}-\bar{\mathbf{A}})^T(\mathbf{A}-\bar{\mathbf{A}})\right] \\
&= E\left[\mathbf{A}^T\mathbf{A} - \bar{\mathbf{A}}^T\mathbf{A} - \mathbf{A}^T\bar{\mathbf{A}} + \bar{\mathbf{A}}^T\bar{\mathbf{A}}\right] \\
&= E\left[\mathbf{A}^T\mathbf{A}\right] - f(\bar{\mathbf{A}}) \\
&= E\left[\mathbf{A}^T\left(\mathbf{A}^T\right)^T\right] - f(\bar{\mathbf{A}}),
\end{aligned} \tag{14}$$

where $f(\bar{\mathbf{A}})$ can be neglected if the data were previously centralized. Thus,

$$\begin{aligned}
\mathbf{G} &\approx E\left[\mathbf{A}^T\left(\mathbf{A}^T\right)^T\right] \\
&= \frac{1}{M}\sum_{k=1}^{M}\mathbf{A}_k^T\left(\mathbf{A}_k^T\right)^T \\
&= m\left(\frac{1}{mM}\mathbf{C}\mathbf{C}^T\right),
\end{aligned} \tag{15}$$

where $\mathbf{C} = \left[\,\mathbf{A}_1^T\ \mathbf{A}_2^T\ \cdots\ \mathbf{A}_M^T\,\right]$ and the term $\frac{1}{mM}\mathbf{C}\mathbf{C}^T$ is the covariance matrix of rows of all images.

3. Bilateral projection frameworks

There are two major difference techniques in this framework, i.e. non-iterative and iterative. All these methods use two projection matrices for both row and column. The former computes these projections separately while the latter computes them simultaneously via iterative process.

3.1 Non-iterative method

The non-iterative bilateral projection scheme was applied to 2DPCA via left and right multiplying projection matrices Xu et al. (2006); Zhang & Zhou (2005); Zuo et al. (2005) as follows

$$\mathbf{B} = \mathbf{Z}^T\mathbf{A}\mathbf{X}, \tag{16}$$

where $\mathbf{B}$ is a feature matrix which extracted from image $\mathbf{A}$ and $\mathbf{Z}$ is a left multiplying projection matrix. Similar to the right multiplying projection matrix $\mathbf{X}$ in Section 2, matrix $\mathbf{Z}$ is a m by q projection matrix that obtained by choosing the eigenvectors of image covariance matrix $\mathbf{H}$ corresponding to the q largest eigenvalues. Therefore, the dimension of feature matrix is decreasing from $m \times n$ to $q \times d$ ($q < m$ and $d < n$). In this way, the computation time also be reducing. Moreover, the recognition accuracy of B2DPCA is often better than 2DPCA as the experimental results in Liu & Chen (2006); Zhang & Zhou (2005); Zuo et al. (2005).

3.2 Iterative method

The bilateral projection scheme of 2DPCA with the iterative algorithm was proposed in Kong et al. (2005); Liu et al. (2010); Xu et al. (2004); Ye (2004). Let $\mathbf{Z} \in \mathbb{R}^{m\times q}$ and $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the left and right multiplying projection matrix respectively. For an $m \times n$ image $\mathbf{A}_k$ and $q \times d$ projected image $\mathbf{B}_k$, the bilateral projection is formulated as follows:

$$\mathbf{B}_k = \mathbf{Z}^T\mathbf{A}_k\mathbf{X} \tag{17}$$

where $\mathbf{B}_k$ is the extracted feature matrix for image $\mathbf{A}_k$.

The optimal projection matrices, $\mathbf{Z}$ and $\mathbf{X}$ in Eq. (17) can be computed by solving the following minimization criterion that the reconstructed image, $\mathbf{Z}\mathbf{B}_k\mathbf{X}^T$, gives the best approximation of $\mathbf{A}_k$:

$$J(\mathbf{Z},\mathbf{X}) = \min \sum_{k=1}^{M} \left\| \mathbf{A}_k - \mathbf{Z}\mathbf{B}_k\mathbf{X}^T \right\|_F^2, \tag{18}$$

where M is the number of data samples and $\| \bullet \|_F$ is the Frobenius norm of a matrix.

The detailed iterative scheme designed to compute the optimal projection matrices, $\mathbf{Z}$ and $\mathbf{X}$, is listed in Table 1. The obtained solutions are locally optimal because the solutions are dependent on the initialized $\mathbf{Z}_0$. In Kong et al. (2005), the initialized $\mathbf{Z}_0$ sets to the $m \times m$ identity matrix $\mathbf{I}_m$, while this value is set to $\begin{bmatrix} \mathbf{I}_q \\ 0 \end{bmatrix}$ in Ye (2004), where $\mathbf{I}_q$ is the $q \times q$ identity matrix.

Alternatively, The criterion in Eq. (18) is biquadratic and has no closed-form solution. Therefore, an iterative procedure to obtain the local optimal solution was proposed in Xu et al. (2004). For $\mathbf{X} \in \mathbb{R}^{n\times d}$, the criterion in Eq. (18) can be rewritten as

$$J(\mathbf{X}) = \min \sum_{k=1}^{M} \left\| \mathbf{A}_k - \mathbf{A}_k^{\mathbf{Z}}\mathbf{X}\mathbf{X}^T \right\|_F^2, \tag{19}$$

where $\mathbf{A}_k^{\mathbf{Z}} = \mathbf{Z}\mathbf{Z}^T\mathbf{A}_k$. The solution of Eq. (19) is the eigenvectors of the eigenvalue decomposition of image covariance matrix:

$$\mathbf{G} = \frac{1}{M}\sum_{k=1}^{M} (\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})^T(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}}). \tag{20}$$

Similarly, for $\mathbf{Z} \in \mathbb{R}^{m\times q}$, the criterion in Eq. (18) is changed to

$$J(\mathbf{Z}) = \min \sum_{k=1}^{M} \left\| \mathbf{A}_k - \mathbf{Z}\mathbf{Z}^T\mathbf{A}_k^{\mathbf{X}} \right\|_F^2, \tag{21}$$

```
$S_1$: Initialize $\mathbf{Z}$, $\mathbf{Z} = \mathbf{Z}_0$ and $i = 0$
$S_2$: While not convergent
$S_3$:     Compute $\mathbf{G} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k - \bar{\mathbf{A}})^T\mathbf{Z}_{i-1}\mathbf{Z}_{i-1}^T(\mathbf{A}_k - \bar{\mathbf{A}})$
$S_4$:     Compute the $d$ eigenvectors $\{\mathbf{e}_j^{\mathbf{X}}\}_{j=1}^{d}$ of $\mathbf{G}$
           corresponding to the largest $d$ eigenvalues
$S_5$:     $\mathbf{X}_i = [\mathbf{e}_1^{\mathbf{X}}, \dots, \mathbf{e}_d^{\mathbf{X}}]$
$S_6$:     Compute $\mathbf{H} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k - \bar{\mathbf{A}})\mathbf{X}_i\mathbf{X}_i^T(\mathbf{A}_k - \bar{\mathbf{A}})^T$
$S_7$:     Compute the $q$ eigenvectors $\{\mathbf{e}_j^{\mathbf{Z}}\}_{j=1}^{q}$ of $\mathbf{H}$
           corresponding to the largest $l$ eigenvalues
$S_8$:     $\mathbf{Z}_i = [\mathbf{e}_1^{\mathbf{Z}}, \dots, \mathbf{e}_q^{\mathbf{Z}}]$
$S_9$:     $i = i + 1$
$S_{10}$: End While
$S_{11}$: $\mathbf{Z} = \mathbf{Z}_{i-1}$
$S_{12}$: $\mathbf{X} = \mathbf{X}_{i-1}$
$S_{13}$: Feature extraction: $\mathbf{B}_k = \mathbf{Z}^T\mathbf{A}_k\mathbf{X}$
```

Table 1. The Bilateral Projection Scheme of 2DPCA with Iterative Algorithm.

where $\mathbf{A}_k^{\mathbf{X}} = \mathbf{A}_k\mathbf{X}\mathbf{X}^T$. Again, the solution of Eq. (21) is the eigenvectors of the eigenvalue decomposition of image covariance matrix:

$$\mathbf{H} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})^T. \tag{22}$$

By iteratively optimizing the objective function with respect to $\mathbf{Z}$ and $\mathbf{X}$, respectively, we can obtain a local optimum of the solution. The whole procedure, namely Coupled Subspace Analysis (CSA) Xu et al. (2004), is shown in Table 2.

4. Kernel based frameworks

From Section 2.2, 2DPCA which performed on the 2D images, is basically PCA performed on the rows of the images if each row is viewed as a computational unit.

Similar to 2DPCA, the kernel-based 2DPCA (K2DPCA) can be processed by traditional kernel PCA (KPCA) in the same manner. Let $\mathbf{a}_k^i$ is the i-th row of the k-th image, thus the k-th image can be rewritten as

$$\mathbf{A} = \left[\,\left(\mathbf{a}_k^1\right)^T\ \left(\mathbf{a}_k^2\right)^T\ \cdots\ \left(\mathbf{a}_k^m\right)^T\,\right]^T. \tag{23}$$

From Eq. (15), the covariance matrix $\mathbf{C}$ can be constructed by concatenating all rows of all training images together. Let $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}, m < m'$ be the mapping function that map the the row vectors into a feature space of higher dimensions in which the classes can be linearly

```
$S_1$: Initialize $\mathbf{Z}$, $\mathbf{Z} = \mathbf{I}_m$
$S_2$: For $i = 1, 2, \ldots, T_{\max}$
$S_3$:     Compute $\mathbf{A}_k^{\mathbf{Z}} = \mathbf{Z}_{i-1}\mathbf{Z}_{i-1}^T\mathbf{A}_k$
$S_4$:     Compute $\mathbf{G} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})^T(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})$
$S_5$:     Compute the $d$ eigenvectors $\{\mathbf{e}_j^{\mathbf{X}}\}_{j=1}^{d}$ of $\mathbf{G}$
           corresponding to the largest $d$ eigenvalues
$S_6$:     $\mathbf{X}_i = [\mathbf{e}_1^{\mathbf{X}}, \ldots, \mathbf{e}_d^{\mathbf{X}}]$
$S_7$:     Compute $\mathbf{A}_k^{\mathbf{X}} = \mathbf{A}_k\mathbf{X}_i\mathbf{X}_i^T$
$S_8$:     Compute $\mathbf{H} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})(\mathbf{A}_k^{\mathbf{Z}} - \bar{\mathbf{A}}^{\mathbf{Z}})^T$
$S_9$:     Compute the $q$ eigenvectors $\{\mathbf{e}_j^{\mathbf{Z}}\}_{j=1}^{q}$ of $\mathbf{H}$
           corresponding to the largest $q$ eigenvalues
$S_{10}$:    $\mathbf{Z}_i = [\mathbf{e}_1^{\mathbf{Z}}, \ldots, \mathbf{e}_q^{\mathbf{Z}}]$
$S_{11}$:    If $t > 2$ and $\|\mathbf{Z}_i - \mathbf{Z}_{i-1}\|_F < m\varepsilon$ and $\|\mathbf{X}_i - \mathbf{X}_{i-1}\|_F < n\varepsilon$
$S_{12}$:    Then Go to $S_3$
$S_{13}$:    Else Go to $S_{15}$
$S_{14}$: End For
$S_{15}$: $\mathbf{Z} = \mathbf{Z}_i$
$S_{16}$: $\mathbf{X} = \mathbf{X}_i$
$S_{17}$: Feature extraction: $\mathbf{B}_k = \mathbf{Z}^T\mathbf{A}_k\mathbf{X}$
```

Table 2. Coupled Subspaces Analysis Algorithm.

separated. Therefore the element in the kernel matrix **K** can be computed by

$$\mathbf{K} = \begin{bmatrix} \varphi(\mathbf{a}_1^1)\varphi(\mathbf{a}_1^1)^T & \cdots & \varphi(\mathbf{a}_1^1)\varphi(\mathbf{a}_1^m)^T & \cdots & \varphi(\mathbf{a}_1^1)\varphi(\mathbf{a}_M^1)^T & \cdots & \varphi(\mathbf{a}_1^1)\varphi(\mathbf{a}_M^m)^T \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{a}_1^m)\varphi(\mathbf{a}_1^1)^T & \cdots & \varphi(\mathbf{a}_1^m)\varphi(\mathbf{a}_1^m)^T & \cdots & \varphi(\mathbf{a}_1^m)\varphi(\mathbf{a}_M^1)^T & \cdots & \varphi(\mathbf{a}_1^m)\varphi(\mathbf{a}_M^m)^T \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{a}_M^1)\varphi(\mathbf{a}_1^1)^T & \cdots & \varphi(\mathbf{a}_M^1)\varphi(\mathbf{a}_1^m)^T & \cdots & \varphi(\mathbf{a}_M^1)\varphi(\mathbf{a}_M^1)^T & \cdots & \varphi(\mathbf{a}_M^1)\varphi(\mathbf{a}_M^m)^T \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{a}_M^m)\varphi(\mathbf{a}_1^1)^T & \cdots & \varphi(\mathbf{a}_M^m)\varphi(\mathbf{a}_1^m)^T & \cdots & \varphi(\mathbf{a}_M^m)\varphi(\mathbf{a}_M^1)^T & \cdots & \varphi(\mathbf{a}_M^m)\varphi(\mathbf{a}_M^m)^T \end{bmatrix} \tag{24}$$

which is an mM-by-mM matrix. Unfortunately, there is a critical problem in implementation about the dimension of its kernel matrix. The kernel matrix is $M \times M$ matrix in KPCA, where M is the number of training samples, while it is $mM \times mM$ matrix in K2DPCA, where m is the number of row of each image. Thus, the K2DPCA kernel matrix is m^2 times of KPCA kernel matrix. For example, if the training set has 200 images with dimensions of 100 × 100 then the dimension of kernel matrix shall be 20000 × 20000, that is very big for fitting in memory unit. After that the projection can be formed by the eigenvectors of this kernel matrix as same as the traditional KPCA.

5. Supervised frameworks

Since the 2DPCA is the unsupervised technique, the class information is neglected. This section presents two methods which can be used to embedded class information to 2DPCA.

Firstly, Linear Discriminant Analysis (LDA) is implemented in 2D framework. Secondly, an 2DPCA is performed for each class in class-specific subspace.

5.1 Two-dimensional linear discriminant analysis of principal component vectors

The PCA's criterion chooses the subspace in the function of data distribution while Linear Discriminant Analysis (LDA) chooses the subspace which yields maximal inter-class distance, and at the same time, keeping the intra-class distance small. In general, LDA extracts features which are better suitable for classification task. However, when the available number of training samples is small compared to the feature dimension, the covariance matrix estimated by these features will be singular and then cannot be inverted. This is called singularity problem or Small Sample Size (SSS) problem Fukunaga (1990).

Various solutions have been proposed for solving the SSS problem Belhumeur et al. (1997); Chen et al. (2000); Huang et al. (2002); Lu et al. (2003); Zhao, Chellappa & Krishnaswamy (1998); Zhao, Chellappa & Nandhakumar (1998) within LDA framework. Among these LDA extensions, Fisherface Belhumeur et al. (1997) and the discriminant analysis of principal components framework Zhao, Chellappa & Krishnaswamy (1998); Zhao, Chellappa & Nandhakumar (1998) demonstrates a significant improvement when applying LDA over principal components from the PCA-based subspace. Since both PCA and LDA can overcome the drawbacks of each other. PCA is constructed around the criteria of preserving the data distribution. Hence, it is suited for representation and reconstruction from the projected feature. However, in the classification tasks, PCA only normalize the input data according to their variance. This is not efficient since the between classes relationship is neglected. In general, the discriminant power depends on both within and between classes relationship. LDA considers these relationships via the analysis of within and between-class scatter matrices. Taking this information into account, LDA allows further improvement. Especially, when there are prominent variation in lighting condition and expression. Nevertheless, all of above techniques, the spatial structure information still be not employed.

Two-Dimensional Linear Discriminant Analysis (2DLDA) was proposed in Ye et al. (2005). For overcoming the SSS problem in classical LDA by working with images in matrix representation, like in 2DPCA. In particular, bilateral projection scheme was applied there via left and right multiplying projection matrices. In this way, the eigenvalue problem was solved two times per iteration. One corresponds to the column direction and another one corresponds to the row direction of image, respectively

Because of 2DPCA is more suitable for face representation than face recognition, like PCA. For better performance in recognition task, LDA is still necessary. Unfortunately, the linear transformation of 2DPCA reduces the input image to a vector with the same dimension as the number of rows or the height of the input image. Thus, the SSS problem may still occurred when LDA is performed after 2DPCA directly. To overcome this problem, a simplified version of the 2DLDA is applied only unilateral projection scheme, based on the 2DPCA concept (Sanguansat et al., 2006b;c). Applying 2DLDA to 2DPCA not only can solve the SSS problem and the curse of dimensionality dilemma but also allows us to work directly on the image matrix in all projections. Hence, spatial structure information is maintained and the size of all scatter matrices cannot be greater than the width of face image. Furthermore, computing with this dimension, the face image do not need to be resized, since all information still be preserved.

5.2 Two-dimensional linear discriminant analysis (2DLDA)

Let $\mathbf{z}$ be a q dimensional vector. A matrix $\mathbf{A}$ is projected onto this vector via the similar transformation as Eq. (1):

$$\mathbf{v}{=}\mathbf{A}\mathbf{z}. \tag{25}$$

This projection yields an m dimensional feature vector.

2DLDA searches for the projection axis $\mathbf{z}$ that maximizing the Fisher's discriminant criterion Belhumeur et al. (1997); Fukunaga (1990):

$$J(\mathbf{z}) = \frac{tr\left(\mathbf{S}_b\right)}{tr\left(\mathbf{S}_w\right)}, \tag{26}$$

where $\mathbf{S}_w$ is the *within-class scatter matrix* and $\mathbf{S}_b$ is the *between-class scatter matrix*. In particular, the within-class scatter matrix describes how data are scattered around the means of their respective class, and is given by

$$\mathbf{S}_w = \sum_{i=1}^{K} Pr(\omega_i)E\left[(\mathbf{Hz})(\mathbf{Hz})^T|\omega = \omega_i\right], \tag{27}$$

where K is the number of classes, $Pr(\omega_i)$ is the prior probability of each class, and $\mathbf{H} = \mathbf{A} - E\mathbf{A}$. The between-class scatter matrix describes how different classes. Which represented by their expected value, are scattered around the mixture means by

$$\mathbf{S}_b = \sum_{i=1}^{K} Pr(\omega_i)E\left[(\mathbf{Fz})(\mathbf{Fz})^T\right], \tag{28}$$

where $\mathbf{F} = E[\mathbf{A}|\omega = \omega_i] - E[\mathbf{A}]$.

With the linearity properties of both the trace function and the expectation, $J(\mathbf{z})$ may be rewritten as

$$\begin{aligned}
J(\mathbf{z}) &= \frac{tr(\sum_{i=1}^{K} Pr(\omega_i)E\left[(\mathbf{Fz})(\mathbf{Fz})^T\right])}{tr(\sum_{i=1}^{K} Pr(\omega_i)E\left[(\mathbf{Hz})(\mathbf{Hz})^T|\omega = \omega_i\right])} \\
&= \frac{\sum_{i=1}^{K} Pr(\omega_i)E\left[tr((\mathbf{Fz})(\mathbf{Fz})^T)\right]}{\sum_{i=1}^{K} Pr(\omega_i)\, E\left(tr\left[(\mathbf{Hz})(\mathbf{Hz})^T|\omega = \omega_i\right]\right)} \\
&= \frac{\sum_{i=1}^{K} Pr(\omega_i)E\left[tr((\mathbf{Fz})^T(\mathbf{Fz}))\right]}{\sum_{i=1}^{K} Pr(\omega_i)\, E\left(tr\left[(\mathbf{Hz})^T(\mathbf{Hz})|\omega = \omega_i\right]\right)} \\
&= \frac{tr\left(\mathbf{z}^T\left(\sum_{i=1}^{K} Pr(\omega_i)E\left[\mathbf{F}^T\mathbf{F}\right]\right)\mathbf{z}\right)}{tr\left(\mathbf{z}^T\left(\sum_{i=1}^{K} Pr(\omega_i)E\left[\mathbf{H}^T\mathbf{H}|\omega = \omega_i\right]\right)\mathbf{z}\right)} \\
&= \frac{tr(\mathbf{z}^T\tilde{\mathbf{S}}_b\mathbf{z})}{tr(\mathbf{z}^T\tilde{\mathbf{S}}_w\mathbf{z})}.
\end{aligned} \tag{29}$$

Furthermore, $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ can be evaluated as follows:

$$\tilde{\mathbf{S}}_b = \sum_{i=1}^{K} \frac{n_i}{K} (\bar{\mathbf{A}}_i - \bar{\mathbf{A}})^T (\bar{\mathbf{A}}_i - \bar{\mathbf{A}}) \tag{30}$$

$$\tilde{\mathbf{S}}_w = \sum_{i=1}^{K} \frac{n_i}{K} \sum_{\mathbf{A}_k \in \omega_i} (\mathbf{A}_k - \bar{\mathbf{A}}_i)^T (\mathbf{A}_k - \bar{\mathbf{A}}_i), \tag{31}$$

where n_i and $\bar{\mathbf{A}}_i$ are the number of elements and the expected value of class ω_i respectively. $\bar{\mathbf{A}}$ denotes the overall mean.

Then the optimal projection vector can be found by solving the following generalized eigenvalue problem:

$$\tilde{\mathbf{S}}_b \mathbf{z} = \lambda \tilde{\mathbf{S}}_w \mathbf{z}. \tag{32}$$

Again the SVD algorithm can be applied to solve this eigenvalue problem on the matrix $\tilde{\mathbf{S}}_w^{-1} \tilde{\mathbf{S}}_b$. Note that, in this size of scatter matrices involved in eigenvalue decomposition process is also become n by n. Thus, with the limited the training set, this decomposition is more reliably than the eigenvalue decomposition based on the classical covariance matrix.

The number of projection vectors is then selected by the same procedure as in Eq. (9). Let $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_q]$ be the projection matrix composed of q largest eigenvectors for 2DLDA. Given a m by n matrix $\mathbf{A}$, its projection onto the principal subspace spanned by $\mathbf{z}_i$ is then given by

$$\mathbf{V} = \mathbf{A}\mathbf{Z}. \tag{33}$$

The result of this projection $\mathbf{V}$ is another matrix of size m by q. Like 2DPCA, this procedure takes a matrix as input and outputs another matrix. These two techniques can be further combined, their combination is explained in the next section.

5.3 2DPCA+2DLDA

In this section, we apply an 2DLDA within the well-known frameworks for face recognition, the LDA of PCA-based feature (Zhao, Chellappa & Krishnaswamy, 1998). This framework consists of 2DPCA and 2DLDA steps, namely 2DPCA+2DLDA. From Section 2, we obtain a linear transformation matrix $\mathbf{X}$ on which each input face image $\mathbf{A}$ is projected. At the 2DPCA step, a feature matrix $\mathbf{Y}$ is obtained. The matrix $\mathbf{Y}$ is then used as the input for the 2DLDA step. Thus, the evaluation of within and between-class scatter matrices in this step will be slightly changed. From Eqs. (30) and (31), the image matrix $\mathbf{A}$ is substituted for the 2DPCA feature matrix $\mathbf{Y}$ as follows

$$\tilde{\mathbf{S}}_b^Y = \sum_{i=1}^{K} \frac{n_i}{K} (\bar{\mathbf{Y}}_i - \bar{\mathbf{Y}})^T (\bar{\mathbf{Y}}_i - \bar{\mathbf{Y}}) \tag{34}$$

$$\tilde{\mathbf{S}}_w^Y = \sum_{i=1}^{K} \frac{n_i}{K} \sum_{\mathbf{Y}_k \in \omega_i} (\mathbf{Y}_k - \bar{\mathbf{Y}}_i)^T (\mathbf{Y}_k - \bar{\mathbf{Y}}_i) \tag{35}$$

where $\mathbf{Y}_k$ is the feature matrix of the k-th image matrix $\mathbf{A}_k$, $\bar{\mathbf{Y}}_i$ be the average of $\mathbf{Y}_k$ which belong to class ω_i and $\bar{\mathbf{Y}}$ denotes a overall mean of $\mathbf{Y}$,

$$\bar{\mathbf{Y}} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{Y}_k. \tag{36}$$

The 2DLDA optimal projection matrix **Z** can be obtained by solving the eigenvalue problem in Eq. (32). Finally, the composite linear transformation matrix, **L=XZ**, is used to map the face image space into the classification space by,

$$\mathbf{D} = \mathbf{AL}. \tag{37}$$

The matrix **D** is 2DPCA+2DLDA feature matrix of image **A** with dimension m by q. However, the number of 2DLDA feature vectors q cannot exceed the number of principal component vectors d. In general case ($q < d$), the dimension of **D** is less than **Y** in Section 2. Thus, 2DPCA+2DLDA can reduce the classification time compared to 2DPCA.

5.4 Class-specific subspace-based two-dimensional principal component analysis

2DPCA is a unsupervised technique that is no information of class labels are considered. Therefore, the directions that maximize the scatter of the data from all training samples might not be as adequate to discriminate between classes. In recognition task, a projection that emphasize the discrimination between classes is more important. The extension of Eigenface, PCA-based, was proposed by using alternative way to represent by projecting to Class-Specific Subspace (CSS) (Shan et al., 2003). In conventional PCA method, the images are analyzed on the features extracted in a low-dimensional space learned from all training samples from all classes. While each subspaces of CSS learned from training samples from one class. In this way, the CSS representation can provide a minimum reconstruction error. The reconstruction error is used to classify the input data via the Distance From CSS (DFCSS). Less DFCSS means more probability that the input data belongs to the corresponding class.

This extension was based on Sanguansat et al. (2006a). Let $\mathbf{G}_k$ be the image covariance matrix of the k^{th} CSS. Then $\mathbf{G}_k$ can be evaluated by

$$\mathbf{G}_k = \frac{1}{M} \sum_{\mathbf{A}_c \in \omega_k} (\mathbf{A}_c - \bar{\mathbf{A}}_k)^T (\mathbf{A}_c - \bar{\mathbf{A}}_k), \tag{38}$$

where $\bar{\mathbf{A}}_k$ is the average image of class ω_k. The k^{th} projection matrix $\mathbf{X}_k$ is a n by d_k projection matrix which composed by the eigenvectors of $\mathbf{G}_k$ corresponding to the d_k largest eigenvalues. The k^{th} CSS of 2DPCA was represented as a 3-tuple:

$$\Re_k^{2DPCA} = \{\mathbf{X}_k, \bar{\mathbf{A}}_k, d_k\} \tag{39}$$

Let **S** be a input sample and $\mathbf{U}_k$ be a feature matrix which projected to the k^{th} CSS, by

$$\mathbf{U}_k = \mathbf{W}_k \mathbf{X}_k, \tag{40}$$

where $\mathbf{W}_k = \mathbf{S} - \bar{\mathbf{A}}_k$. Then the reconstruct image $\mathbf{W}_k^r$ can be evaluates by

$$\mathbf{W}_k^r = \mathbf{U}_k \mathbf{X}_k^T. \tag{41}$$

Therefore, the DFCSS is defined by reconstruction error as follows

$$\varepsilon_k(\mathbf{W}_k^r, \mathbf{S}) = \sum_{m=1}^{n_{row}} \sum_{n=1}^{n_{col}} \left| \mathbf{w}^r_{(m,n)_k} - \mathbf{s}_{(m,n)} \right|. \tag{42}$$

If $\varepsilon_t = \min_{1 \le k \le K} (\varepsilon_k)$ then the input sample **S** is belong to class ω_t.

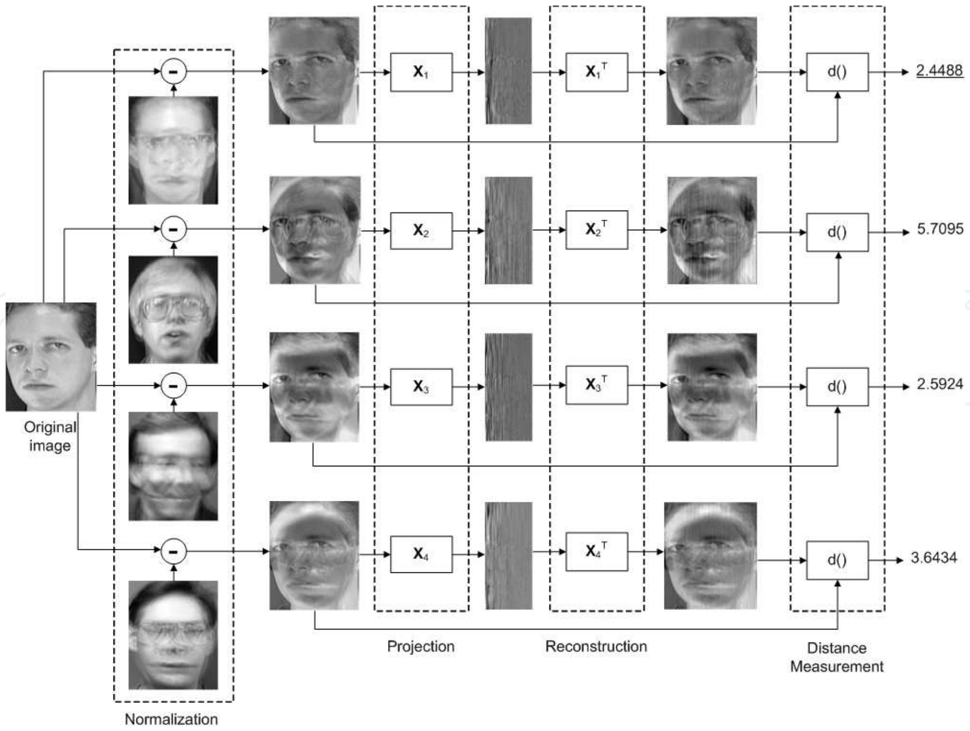


Fig. 1. CSS-based 2DPCA diagram.

For illustration, we assume that there are 4 classes, as shown in Fig. 1. The input image must be normalized with the averaging images of all 4 classes. And then project to 2DPCA subspaces of each class. After that the image is reconstructed by the projection matrices (**X**) in each class. The DFCSS is used now to measure the similarity between the reconstructed image and the normalized original image on each CSS. From Fig. 1, the DFCSS of the first class is minimum, thus we decide this input image is belong to the first class.

6. Alignment based frameworks

Since 2DPCA can be viewed as the row-based PCA, that means the information contains only in row direction. Although, combining it with the column-based 2DPCA can consider the information in both row and column directions. But there still be other directions which should be considered.

6.1 Diagonal-based 2DPCA (DiaPCA)

The motivation for developing the DiaPCA method originates from an essential observation on the recently proposed 2DPCA (Yang et al., 2004). In contrast to 2DPCA, DiaPCA seeks the optimal projective vectors from diagonal face images and therefore the correlations between variations of rows and those of columns of images can be kept. Therefore, this problem can solve by transforming the original face images into corresponding diagonal face images, as shown in Fig. 2 and Fig. 3. Because the rows (columns) in the transformed diagonal face images simultaneously integrate the information of rows and columns in original images, it can reflect both information between rows and those between columns. Through the entanglement of row and column information, it is expected that DiaPCA may find some useful block or structure information for recognition in original images. The sample diagonal face images on Yale database are displayed in Fig. 4.

Experimental results on a subset of FERET database (Zhang et al., 2006) show that DiaPCA is more accurate than both PCA and 2DPCA. Furthermore, it is shown that the accuracy can be further improved by combining DiaPCA and 2DPCA together.

6.2 Image cross-covariance analysis

In PCA, the covariance matrix provides a measure of the strength of the correlation of all pixel pairs. Because of the limit of the number of training samples, thus this covariance cannot be well estimated. While the performance of 2DPCA is better than PCA, although all of the correlation information of pixel pairs are not employed for estimating the image covariance matrix. Nevertheless, the disregard information may possibly include the useful information. Sanguansat et al. (2007a) proposed a framework for investigating the information which was neglected by original 2DPCA technique, so called Image Cross-Covariance Analysis (ICCA). To achieve this point, the *image cross-covariance matrix* is defined by two variables, the first variable is the original image and the second one is the shifted version of the former. By our shifting algorithm, many image cross-covariance matrices are formulated to cover all of the information. The Singular Value Decomposition (SVD) is applied to the image cross-covariance matrix for obtaining the optimal projection matrices. And we will show that these matrices can be considered as the orthogonally rotated projection matrices of traditional 2DPCA. ICCA is different from the original 2DPCA on the fact that the transformations of our method are generalized transformation of the original 2DPCA.

First of all, the relationship between 2DPCA's image covariance matrix $\mathbf{G}$, in Eq. (5), and PCA's covariance matrix $\mathbf{C}$ can be considered as

$$\mathbf{G}\left(i,j\right)=\sum_{k=1}^{m}\mathbf{C}\left(m\left(i-1\right)+k,m\left(j-1\right)+k\right) \tag{43}$$

where $\mathbf{G}(i,j)$ and $\mathbf{C}(i,j)$ are the i^{th} row, j^{th} column element of matrix $\mathbf{G}$ and matrix $\mathbf{C}$, respectively. And m is the height of the image.

For illustration, let the dimension of all training images are 3 by 3. Thus, the covariance matrix of these images will be a 9 by 9 matrix and the dimension of image covariance matrix is only 3 by 3, as shown in Fig. 5.

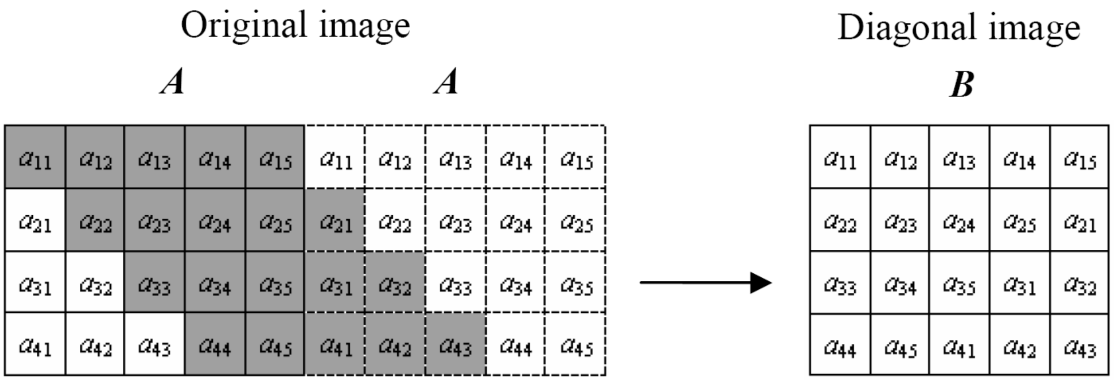
From Eq. (43), each elements of $\mathbf{G}$ is the sum of all the same label elements in $\mathbf{C}$, for example:

$$\begin{aligned}\mathbf{G}(\mathbf{1},\mathbf{1}) &= \mathbf{C}(\mathbf{1},\mathbf{1})+\mathbf{C}(\mathbf{2},\mathbf{2})+\mathbf{C}(\mathbf{3},\mathbf{3}),\\ \mathbf{G}(\mathbf{1},\mathbf{2}) &= \mathbf{C}(\mathbf{1},\mathbf{4})+\mathbf{C}(\mathbf{2},\mathbf{5})+\mathbf{C}(\mathbf{3},\mathbf{6}),\\ \mathbf{G}(\mathbf{1},\mathbf{3}) &= \mathbf{C}(\mathbf{1},\mathbf{7})+\mathbf{C}(\mathbf{2},\mathbf{8})+\mathbf{C}(\mathbf{3},\mathbf{9}).\end{aligned} \tag{44}$$

It should be note that the total power of image covariance matrix equals and traditional covariance matrix $\mathbf{C}$ are identical,

$$tr(\mathbf{G})=tr(\mathbf{C}). \tag{45}$$

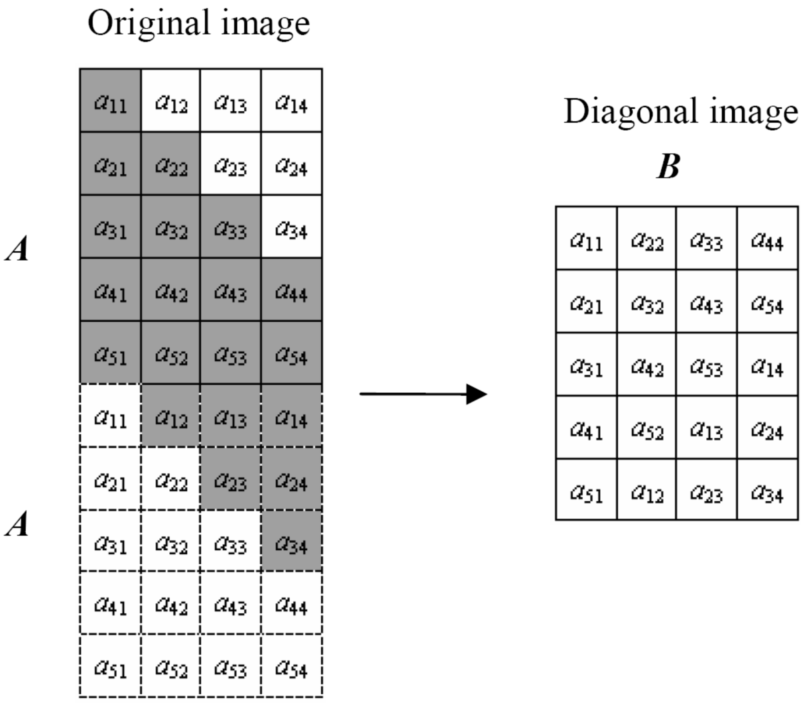
From this point of view in Eq. (43), we can see that image covariance matrix is collecting the classification information only $1/m$ of all information collected in traditional covariance matrix. However, there are the other $(m-1)/m$ elements of the covariance matrix still be not



a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{22}	a_{23}	a_{24}	a_{25}	a_{21}
a_{33}	a_{34}	a_{35}	a_{31}	a_{32}
a_{44}	a_{45}	a_{41}	a_{42}	a_{43}

Fig. 2. Illustration of the ways for deriving the diagonal face images: If the number of columns is more than the number of rows



a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}
a_{51}	a_{52}	a_{53}	a_{54}
a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}
a_{51}	a_{52}	a_{53}	a_{54}

a_{11}	a_{22}	a_{33}	a_{44}
a_{21}	a_{32}	a_{43}	a_{54}
a_{31}	a_{42}	a_{53}	a_{14}
a_{41}	a_{52}	a_{13}	a_{24}
a_{51}	a_{12}	a_{23}	a_{34}

Fig. 3. Illustration of the ways for deriving the diagonal face images: If the number of columns is less than the number of rows

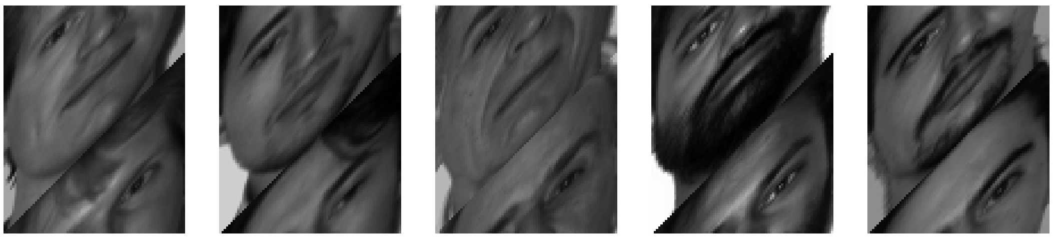


Fig. 4. The sample diagonal face images on Yale database.

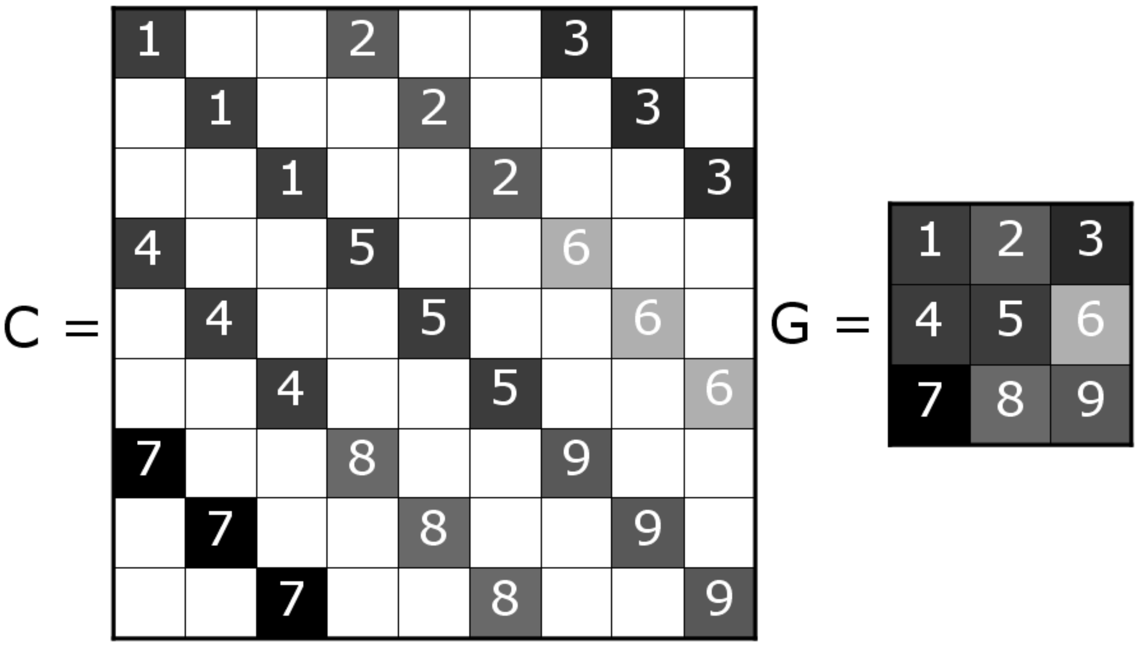


Fig. 5. The relationship of covariance and image covariance matrix.

considered. By the experimental results in Sanguansat et al. (2007a). For investigating how the retaining information in 2D subspace is rich for classification, the new **G** is derived from the PCA's covariance matrix as

$$\mathbf{G}_L(i,j)=\sum_{k=1}^{m}\mathbf{C}\left(f\left(m\left(i-1\right)+k\right),m\left(j-1\right)+k\right), \tag{46}$$

$$f(x)=\begin{cases}x+L-1, & 1\le x\le mn-L+1\\ x-mn+L-1, & mn-L+2\le x\le mn\end{cases} \tag{47}$$

where $1 \le L \le mn$.

The $\mathbf{G}_L$ can also be determined by applying the shifting to each images instead of averaging certain elements of covariance matrix. Therefore, the $\mathbf{G}_L$ can alternatively be interpreted as the *image cross-covariance matrix* or

$$\mathbf{G}_L = E[(\mathbf{B}_L - E[\mathbf{B}_L])^T(\mathbf{A} - E[\mathbf{A}])] \tag{48}$$

where $\mathbf{B}_L$ is the L^{th} shifted version of image **A** that can be created via algorithm in Table 3. The samples of shifted images $\mathbf{B}_L$ are presented in Fig. 6.

In 2DPCA, the columns of the projection matrix, **X**, are obtained by selection the eigenvectors which corresponding to the d largest eigenvalues of image covariance matrix, in Eq. (5). While in ICCA, the eigenvalues of image cross-covariance matrix, $\mathbf{G}_L$, are complex number with non-zero imaginary part. The Singular Value Decomposition (SVD) is applied to this matrix instead of Eigenvalue decomposition. Thus, the ICCA projection matrix contains a set of orthogonal basis vectors which corresponding to the d largest singular values of image cross-covariance matrix.

For understanding the relationship between the ICCA projection matrix and the 2DPCA projection matrix, we will investigate in the simplest case, i.e. there are only one training

```
S_1:  Input $m \times n$ original image $\mathbf{A}$
      and the number of shifting $L$ ($2 \leq L \leq mn$).
S_2:  Initialize the row index, $irow = [2, \ldots, n, 1]$,
      and output image $\mathbf{B}$ = $m \times n$ zero matrix.
S_3:  For $i = 1, 2, \ldots, L-1$
S_4:      Sort the first row of $\mathbf{A}$ by the row index, $irow$.
S_5:      Set the last row of $\mathbf{B}$ = the first row of $\mathbf{A}$.
S_6:      For $j = 1, 2, \ldots, m-1$
S_7:          Set the $j^{th}$ row of $\mathbf{B}$ = the $(j+1)^{th}$ row of $\mathbf{A}$.
S_8:      End For
S_9:      Set $\mathbf{A} = \mathbf{B}$
S_10: End For
```

Table 3. The Image Shifting Algorithm for ICCA

image. Therefore, the image covariance matrix and image cross-covariance matrix are simplified to $\mathbf{A}^T\mathbf{A}$ and $\mathbf{B}_L^T\mathbf{A}$, respectively.

The image $\mathbf{A}$ and $\mathbf{B}_L$ can be decomposed by using Singular Value Decomposition (SVD) as

$$\mathbf{A} = \mathbf{U_A}\mathbf{D_A}\mathbf{V_A^T}, \tag{49}$$

$$\mathbf{B}_L = \mathbf{U}_{\mathbf{B}_L}\mathbf{D}_{\mathbf{B}_L}\mathbf{V}_{\mathbf{B}_L}^T. \tag{50}$$

Where $\mathbf{V_A}$ and $\mathbf{V}_{\mathbf{B}_L}$ contain a set of the eigenvectors of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{B}_L^T\mathbf{B}_L$, respectively. And $\mathbf{U_A}$ and $\mathbf{U}_{\mathbf{B}_L}$ contain a set of the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{B}_L\mathbf{B}_L^T$, respectively. And $\mathbf{D_A}$ and $\mathbf{D}_{\mathbf{B}_L}$ contain the singular values of $\mathbf{A}$ and $\mathbf{B}_L$, respectively. If all eigenvectors of $\mathbf{A}^T\mathbf{A}$ are selected then the $\mathbf{V_A}$ is the 2DPCA projection matrix, i.e. $\mathbf{X} = \mathbf{V_A}$.

Let $\mathbf{Y} = \mathbf{A}\mathbf{V_A}$ and $\mathbf{Z} = \mathbf{B}_L\mathbf{V}_{\mathbf{B}_L}$ are the projected matrices of $\mathbf{A}$ and $\mathbf{B}$, respectively. Thus,

$$\mathbf{B}_L^T\mathbf{A} = \mathbf{V}_{\mathbf{B}_L}\mathbf{Z}^T\mathbf{Y}\mathbf{V}_\mathbf{A}^T. \tag{51}$$

Denoting the SVD of $\mathbf{Z}^T\mathbf{Y}$ by

$$\mathbf{Z}^T\mathbf{Y} = \mathbf{P}\mathbf{D}\mathbf{Q}^T, \tag{52}$$

and substituting into Eq. (51) gives

$$\begin{aligned}\mathbf{B}_L^T\mathbf{A} &= \mathbf{V}_{\mathbf{B}_L}\mathbf{P}\mathbf{D}\mathbf{Q}^T\mathbf{V}_\mathbf{A}^T \\ &= \mathbf{R}\mathbf{D}\mathbf{S}^T,\end{aligned} \tag{53}$$

where $\mathbf{RDS}^T$ is the singular value decomposition of $\mathbf{B}_L^T\mathbf{A}$ because of the unique properties of the SVD operation. It should be note that $\mathbf{B}_L^T\mathbf{A}$ and $\mathbf{Z}^T\mathbf{Y}$ have the same singular values. Therefore,

$$\mathbf{R} = \mathbf{V}_{\mathbf{B}_L}\mathbf{P}, \tag{54}$$

$$\mathbf{S} = \mathbf{V_A}\mathbf{Q} = \mathbf{X}\mathbf{Q} \tag{55}$$

can be thought of as orthogonally rotated of projection matrices $\mathbf{V_A}$ and $\mathbf{V}_{\mathbf{B}_L}$, respectively.

As a result in Eq. (55), the ICCA projection matrix is the orthogonally rotated of original 2DPCA projection matrix.

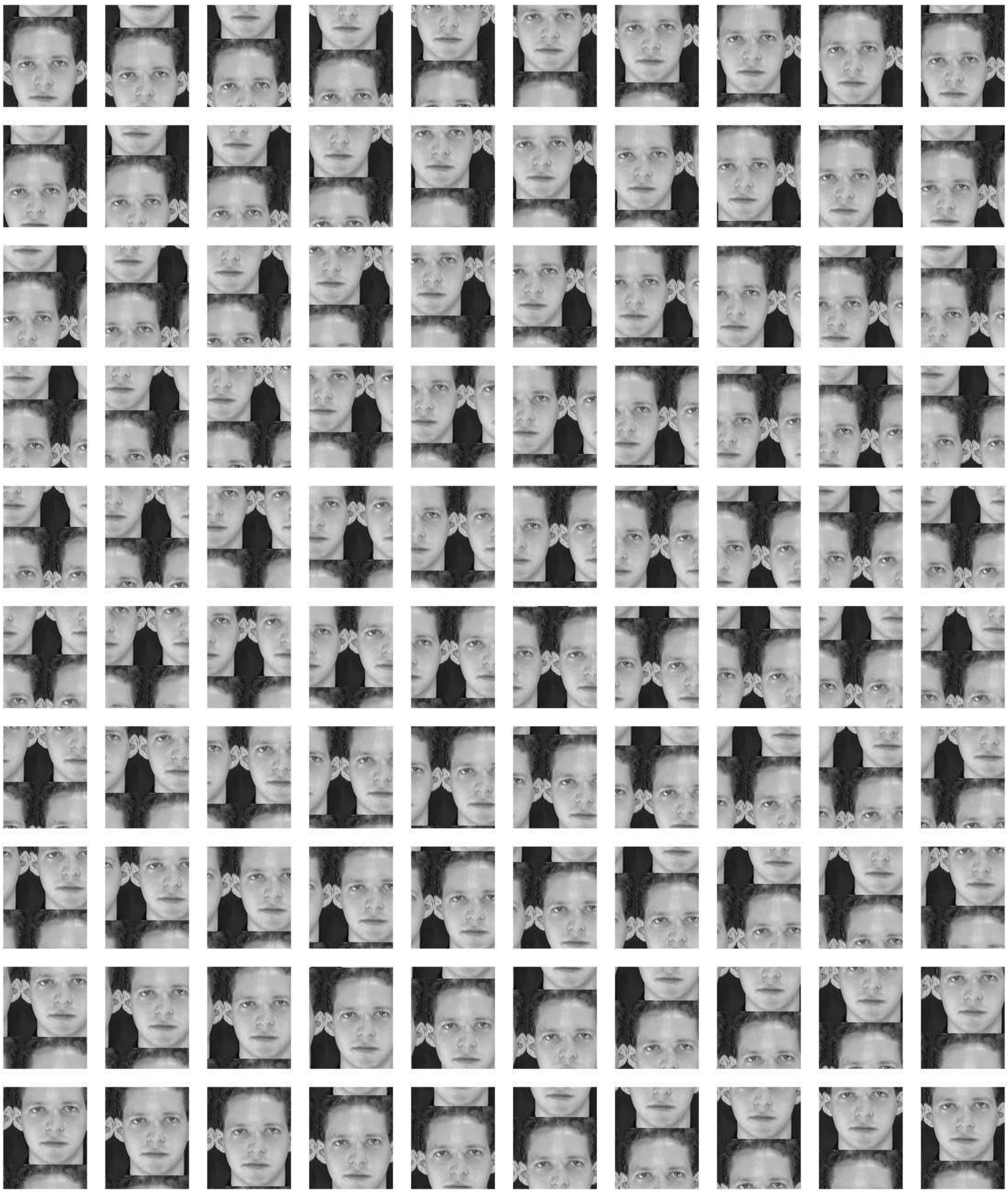


Fig. 6. The samples of shifted images on the ORL database.

7. Random frameworks

In feature selection, the random subspace method can improve the performance by combining many classifiers which corresponds to each random feature subset. In this section, the random method is applied to 2DPCA in various ways to improve its performance.

7.1 Two-dimensional random subspace analysis (2DRSA)

The main disadvantage of 2DPCA is that it needs many more coefficients for image representation than PCA. Many works try to solve this problem. In Yang et al. (2004), PCA is used after 2DPCA for further dimensional reduction, but it is still unclear how the dimension of 2DPCA could be reduced directly. Many methods to overcome this problem were proposed by applied the bilateral-projection scheme to 2DPCA. In Zhang & Zhou (2005); Zuo et al. (2005), the right and left multiplying projection matrices are calculated independently while the iterative algorithm is applied to obtain the optimal solution of these projection matrices in Kong et al. (2005); Ye (2004). And the non-iterative algorithm for optimization was proposed in Liu & Chen (2006). In Xu et al. (2004), they proposed the iterative procedure which the right projection is calculated by the reconstructed images of the left projection and the left projection is calculated by the reconstructed images of the right projection. Nevertheless, all of above methods obtains only the local optimal solution.

Another method for dealing with high-dimensional space was proposed in Ho (1998b), called Random Subspace Method (RSM). This method is the one of ensemble classification methods, like Bagging Breiman (1996) and Boosting Freund & Schapire (1995). However, Bagging and Boosting are not reduce the high-dimensionality. Bagging randomly select a number of samples from the original training set to learn an individual classifier while Boosting specifically weight each training sample. The RSM can effectively exploit the high-dimensionality of the data. It constructs an ensemble of classifiers on independently selected feature subsets, and combines them using a heuristic such as majority voting, sum rule, etc.

There are many reasons the Random Subspace Method is suitable for face recognition task. Firstly, this method can take advantage of high dimensionality and far away from the curse of dimensionality (Ho, 1998b). Secondly, the random subspace method is useful for critical training sample sizes (Skurichina & Duin, 2002). Normally in face recognition, the dimension of the feature is extremely large compared to the available number of training samples. Thus applying RSM can avoid both of the curse of dimensionality and the SSS problem. Thirdly, The nearest neighbor classifier, a popular choice in the 2D face-recognition domain (Kong et al., 2005; Liu & Chen, 2006; Yang et al., 2004; Ye, 2004; Zhang & Zhou, 2005; Zuo et al., 2005), can be very sensitive to the sparsity in the high-dimensional space. Their accuracy is often far from optimal because of the lack of enough samples in the high-dimensional space. The RSM brings significant performance improvements compared to a single classifier Ho (1998a); Skurichina & Duin (2002). Finally, since there is no hill climbing in RSM, there is no danger of being trapped in local optima Ho (1998b).

The RSM was applied to PCA for face recognition in Chawla & Bowyer (2005). They apply the random selection directly to the feature vector of PCA for constructing the multiple subspaces. Nevertheless, the information which contained in each element of PCA feature vector is not equivalent. Normally, the element which corresponds to the larger eigenvalue, contains more useful information. Therefore, applying RSM to PCA feature vector is seldom appropriate.

```
$S_1$: Project image, $\mathbf{A}$, by Eq. (10).
$S_2$: For $i = 1$ to the number of classifiers
$S_3$:     Randomly select a $r$ dimensional random subspace, $\mathbf{Z}_i^r$,
           from $\mathbf{Y}$ ($r < m$).
$S_4$:     Construct the nearest neighbor classifier, $\mathbf{C}_i^r$.
$S_5$: End For
$S_6$: Combine the output of each classifiers by using majority voting.
```

Table 4. Two-Dimensional Random Subspace Analysis Algorithm

Different from PCA, the 2DPCA feature is a matrix form. Thus, RSM is more suitable for 2DPCA, because the column direction does not depend on the eigenvalue.

A framework of Two-Dimensional Random Subspace Analysis (2DRSA) (Sanguansat et al., n.d.) is proposed to extend the original 2DPCA. The RSM is applied to feature space of 2DPCA for generating the vast number of feature subspaces, which be constructed by an autonomous, pseudorandom procedure to select a small number of dimensions from a original feature space. For a m by n feature matrix, there are 2^m such selections that can be made, and with each selection a feature subspace can be constructed. And then individual classifiers are created only based on those attributes in the chosen feature subspace. The outputs from different individual classifiers are combined by the uniform majority voting to give the final prediction.

The Two-Dimensional Random Subspace Analysis consists of two parts, 2DPCA and RSM. After data samples was projected to 2D feature space via 2DPCA, the RSM are applied here by taking advantage of high dimensionality in these space to obtain the lower dimensional multiple subspaces. A classifier is then constructed on each of those subspaces, and a combination rule is applied in the end for prediction on the test sample. The 2DRSA algorithm is listed in Table 4, the image matrix, $\mathbf{A}$, is projected to feature space by 2DPCA projection in Eq. (10). In this feature space, it contains the data samples in matrix form, the $m \times d$ feature matrix, $\mathbf{Y}$ in Eq. (10). The dimensions of feature matrix $\mathbf{Y}$ depend on the height of image (m) and the number of selected eigenvectors of the image covariance matrix $\mathbf{G}$ (d). Therefore, only the information which embedded in each element on the row direction was sorted by the eigenvalue but not on the column direction. It means this method should randomly pick up some rows of feature matrix $\mathbf{Y}$ to construct the new feature matrix $\mathbf{Z}$. The dimension of $\mathbf{Z}$ is $r \times d$, normally r should be less than m. The results in Ho (1998b) have shown that for a variety of data sets adopting half of the feature components usually yields good performance.

7.2 Two-dimensional diagonal random subspace analysis (2D^{2}RSA)

The extension of 2DRSA was proposed in Sanguansat et al. (2007b), namely the Two-Dimensional Diagonal Random Subspace Analysis. It consists of two parts i.e. DiaPCA and RSM. Firstly, all images are transformed into the diagonal face images as in Section 6.1. After that the transformed image samples was projected to 2D feature space via DiaPCA, the RSM are applied here by taking advantage of high dimensionality in these space to obtain the lower dimensional multiple subspaces. A classifier is then constructed on each of those subspaces, and a combination rule is applied in the end for prediction on the test sample. Similar to 2DRSA, the 2D^{2}RSA algorithm is listed in Table 5.

```
$S_1$: Transforming images into diagonal images.
$S_2$: Project image, $\mathbf{A}$, by Eq. (10).
$S_3$: For $i = 1$ to the number of classifiers
$S_4$:     Randomly select a $r$ dimensional random subspace,
           $\mathbf{Z}_i^r$, from $\mathbf{Y}$ ($r < m$).
$S_5$:     Construct the nearest neighbor classifier, $\mathbf{C}_i^r$.
$S_6$: End For
$S_7$: Combine the output of each classifiers by using
       majority voting.
```

Table 5. Two-Dimensional Diagonal Random Subspace Analysis Algorithm.

7.3 Random subspace method-based image cross-covariance analysis

As discussed in Section 6.2, not all elements of the covariance matrix is used in 2DPCA. Although, the image cross-covariance matrix can be switching these elements to formulate many versions of image cross-covariance matrix, the $(m-1)/m$ elements of the covariance matrix are still not advertent in the same time. For integrating this information, the Random Subspace Method (RSM) can be using here via randomly select the number of shifting L to construct a set of multiple subspaces. That means each subspace is formulated from difference versions of image cross-covariance matrix. And then individual classifiers are created only based on those attributes in the chosen feature subspace. The outputs from different individual classifiers are combined by the uniform majority voting to give the final prediction. Moreover, the RSM can be used again for constructing the subspaces which are corresponding to the difference number of basis vectors d. Consequently, the number of all random subspaces of ICCA reaches to $d \times L$. That means applying the RSM to ICCA can be constructed more subspaces than 2DRSA. As a result, the RSM-based ICCA can alternatively be apprehended as the generalized 2DRSA.

8. Conclusions

This chapter presents the extensions of 2DPCA in several frameworks, i.e. bilateral projection, kernel method, supervised based, alignment based and random approaches. All of these methods can improve the performance of traditional 2DPCA for image recognition task. The bilateral projection can obtain the smallest feature matrix compared to the others. The class information can be embedded in the projection matrix by supervised frameworks that means the discriminant power should be increased. The alternate alignment of pixels in image can reveal the latent information which is useful for the classifier. The kernel based 2DPCA can achieve to the highest performance but the appropriated kernel's parameters and a huge of memory are required to manipulate the kernel matrix while the random subspace method is good for robustness.

9. References

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

Breiman, L. (1996). Bagging predictors, *Machine Learning* 24(2): 123–140.

Chawla, N. V. & Bowyer, K. (2005). Random subspaces and subsampling for 2D face recognition, *Computer Vision and Pattern Recognition*, Vol. 2, pp. 582–589.

Chen, L., Liao, H., Ko, M., Lin, J. & Yu, G. (2000). A new LDA based face recognition system which can solve the small sample size problem, *Pattern Recognition* 33(10): 1713–1726.

Freund, Y. & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting, *European Conference on Computational Learning Theory*, pp. 23–37.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, second edn, Academic Press.

Ho, T. K. (1998a). Nearest neighbors in random subspaces, *Proceedings of the 2nd Int'l Workshop on Statistical Techniques in Pattern Recognition*, Sydney, Australia, pp. 640–648.

Ho, T. K. (1998b). The random subspace method for constructing decision forests, *IEEE Trans. Pattern Anal. and Mach. Intell.* 20(8): 832–844.

Huang, R., Liu, Q., Lu, H. & Ma, S. (2002). Solving the small sample size problem of LDA, *Pattern Recognition* 3: 29–32.

Kong, H., Li, X., Wang, L., Teoh, E. K., Wang, J.-G. & Venkateswarlu, R. (2005). Generalized 2D principal component analysis, *IEEE International Joint Conference on Neural Networks (IJCNN)* 1: 108–113.

Liu, J. & Chen, S. (2006). Non-iterative generalized low rank approximation of matrices, *Pattern Recognition Letters* 27: 1002–1008.

Liu, J., Chen, S., Zhou, Z.-H. & Tan, X. (2010). Generalized low-rank approximations of matrices revisited, *Neural Networks, IEEE Transactions on* 21(4): 621 –632.

Lu, J., Plataniotis, K. N. & Venetsanopoulos, A. N. (2003). Regularized discriminant analysis for the small sample size problem in face recognition, *Pattern Recogn. Lett.* 24(16): 3079–3087.

Nguyen, N., Liu, W. & Venkatesh, S. (2007). Random subspace two-dimensional pca for face recognition, *Proceedings of the multimedia 8th Pacific Rim conference on Advances in multimedia information processing*, PCM'07, Springer-Verlag, Berlin, Heidelberg, pp. 655–664.
URL: *http://portal.acm.org/citation.cfm?id=1779459.1779555*

Sanguansat, P. (2008). 2dpca feature selection using mutual information, *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, pp. 578 –581.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (2006a). Class-specific subspace-based two-dimensional principal component analysis for face recognition, *International Conference on Pattern Recognition*, Vol. 2, Hong Kong, China, pp. 1246–1249.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (2006b). Two-dimensional linear discriminant analysis of principle component vectors for face recognition, *IEICE Trans. Inf. & Syst. Special Section on Machine Vision Applications* E89-D(7): 2164–2170.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (2006c). Two-dimensional linear discriminant analysis of principle component vectors for face recognition, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, Toulouse, France, pp. 345–348.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (2007a). Image cross-covariance analysis for face recognition, *IEEE Region 10 Conference on Convergent Technologies for the Asia-Pacific*, Taipei, Taiwan.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (2007b). Two-dimensional diagonal random subspace analysis for face recognition, *International Conference on Telecommunications, Industry and Regulatory Development*, Vol. 1, pp. 66–69.

Sanguansat, P., Asdornwised, W., Jitapunkul, S. & Marukatat, S. (n.d.). Two-dimensional random subspace analysis for face recognition, *7th International Symposium on Communications and Information Technologies*.

Shan, S., Gao, W. & Zhao, D. (2003). Face recognition based on face-specific subspace, *International Journal of Imaging Systems and Technology* 13(1): 23–32.

Sirovich, L. & Kirby, M. (1987). Low-dimensional procedure for characterization of human faces, *J. Optical Soc. Am.* 4: 519–524.

Skurichina, M. & Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linear classifiers, *Pattern Anal. Appl.* 5(2): 121–135.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition, *J. of Cognitive Neuroscience* 3(1): 71–86.

Xu, A., Jin, X., Jiang, Y. & Guo, P. (2006). Complete two-dimensional PCA for face recognition, *International Conference on Pattern Recognition*, Vol. 3, pp. 481–484.

Xu, D., Yan, S., Zhang, L., Liu, Z. & Zhang, H. (2004). Coupled subspaces analysis, *Technical report*, Microsoft Research.

Yang, J. & Yang, J. Y. (2002). From image vector to matrix: A straightforward image projection technique IMPCA vs. PCA, *Pattern Recognition* 35(9): 1997–1999.

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

Ye, J. (2004). Generalized low rank approximations of matrices, *International Conference on Machine Learning*, pp. 887–894.

Ye, J., Janardan, R. & Li, Q. (2005). Two-dimensional linear discriminant analysis, *in* L. K. Saul, Y. Weiss & L. Bottou (eds), *Advances in Neural Information Processing Systems 17*, MIT Press, Cambridge, MA, pp. 1569–1576.

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

Zhang, D., Zhou, Z.-H. & Chen, S. (2006). Diagonal principal component analysis for face recognition, *Pattern Recognition* 39(1): 133–135.

Zhao, W., Chellappa, R. & Krishnaswamy, A. (1998). Discriminant analysis of principle components for face recognition, *IEEE 3rd Inter. Conf. on Automatic Face and Gesture Recognition*, Japan.

Zhao, W., Chellappa, R. & Nandhakumar, N. (1998). Empirical performance analysis of linear discriminant classifiers, *Computer Vision and Pattern Recognition*, IEEE Computer Society, pp. 164–171.

Zuo, W., Wang, K. & Zhang, D. (2005). Bi-dierectional PCA with assembled matrix distance metric, *International Conference on Image Processing*, Vol. 2, pp. 958–961.

Principal Component Analysis

Edited by Dr. Parinya Sanguansat

ISBN 978-953-51-0195-6
Hard cover, 300 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

This book is aimed at raising awareness of researchers, scientists and engineers on the benefits of Principal Component Analysis (PCA) in data analysis. In this book, the reader will find the applications of PCA in fields such as image processing, biometric, face recognition and speech processing. It also includes the core concepts and the state-of-the-art methods in data analysis and feature extraction.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821