Matrix FastICA Algorithm and Its Application to Face Recognition

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID:	TPAMI-2007-10-0737
Manuscript Type:	Regular
Keywords:	I.5.1.e Statistical < I.5.1 Models < I.5 Pattern Recognition < I Computing Methodologies, I.5.4.d Face and gesture recognition < I.5.4 Applications < I.5 Pattern Recognition < I Computing Methodologies, I.5.4.m Signal processing < I.5.4 Applications < I.5 Pattern Recognition < I Computing Methodologies

Matrix FastICA Algorithm and Its Application to Face Recognition

Q. Gao, L. Zhang[1], D. Zhang and J. Yang

***Abstract* -- Independent component analysis (ICA) exploits the high-order statistics of data and hence can better extract the structural features than principle component analysis. The conventional ICA, however, suffers from the small sample size (SSS) problem, i.e. the dimensionality of feature space is much higher than the number of available training samples. In this paper, two new measurements of the non-Gaussianity of vector variables, instead of scalar variables, are defined to extract the independent components (ICs) from images rather than vectors. Then the Matrix-FastICA algorithm is developed to compute the demixing matrix of images. In order to reduce the feature storage space and generalize the Matrix-FastICA algorithm to a higher dimensional array, a tensor ICA (T-ICA) algorithm is proposed to extract image ICs in a tensor form. Compared with the conventional ICA, neither Matrix-FastICA nor T-ICA needs the image-to-vector transform and thus they can better preserve the local structural information embedded in images. The SSS problem in conventional ICA is also significantly alleviated because the demixing vectors are directly estimated from image matrices. Extensive experiments on face databases are performed and the results validate that the proposed Matrix-FastICA and T-ICA algorithms outperform many state-of-the-art subspace analysis schemes.**

***Index terms* --** Independent component analysis (ICA), feature extraction, face recognition

1. Introduction

Dimensionality reduction, aiming at revealing meaningful structures and unexpected relationships in multivariate data, is one of the key techniques in cluster analysis, pattern recognition and image classification, especially in biometric authentication applications such as face recognition [1-2]. The most representative approaches to dimensionality reduction may be

[1] Corresponding author. Dr. Lei Zhang is with the Biometric Research Center, Dept. of Computing, The Hong Kong Polytechnic University. Email: cslzhang@comp.polyu.edu.hk. This research is partially supported by the National Science Foundation Council of China (NSFC) under Grant no. 60634030 and partially supported by the NSFC Key Oversea Project under Grant no. 60620160097.
Dr. Quanxue Gao is with the Biometric Research center, Dept. of Computing, The Hong Kong Polytechnic University, and with the School of Telecommunications Engineering, XIDIAN University, Xi'an, China. Email: csqxgao@comp.polyu.edu.hk.
Prof. David Zhang is with the Biometric Research Center, Dept. of Computing, The Hong Kong Polytechnic University. Email: csdzhang@comp.polyu.edu.hk.
Dr. Jian Yang is with the School of Computer Science and Technology, Nanjing University of Science of Technology. Email: csjyang@mail.njust.edu.cn.

the subspace analysis methods (SAM), which include principal component analysis (PCA) [3-4], independent component analysis (ICA) [5], linear discriminant analysis (LDA) [6], etc. PCA and ICA are popular unsupervised SAM schemes and LDA is a popular supervised SAM technique. In general, SAM is to find an optimal subspace or projection via different criteria.

As one of the most popular SAM techniques, PCA is to find an optimal subspace so that the image coordinates are uncorrelated in it. Kirby and Sirovich [3] showed that any face image can be economically represented along the eigenpictures coordinate space and be approximately reconstructed using just a small collection of eigenpictures and their corresponding projections. Based on this work, Turk and Pentland [4] developed the well-known Eigenfaces method for face recognition. Since then, PCA has been extensively investigated and many related face recognition algorithms have been developed [7-8]. PCA, however, exploits only the second-order statistics of the data, and is optimal only if the underlying data follow the Gaussian distribution. It has been observed in many real applications that the natural signals, including speech, EEG signal, and natural images, can be better described as linear combinations of sources with long tailed distributions [5, 9]. In addition, the features extracted via PCA capture the amplitude spectrum of images but not their phase structure spectrum [5].

It has been reported that the high-order statistics can capture the phase spectrum of images [5, 10], which can be very useful for image representation and recognition. For a given image, if we scramble its phase spectrum while maintaining its power spectrum, this will dramatically alter the appearance of the image without changing its second-order statistics. As pointed out by Oppenheim and Lim [11], the phase spectrum contains more image structural information than the power spectrum for human visual perception. For example, if we synthesize a facial image from the amplitude spectrum of face *A* and the phase spectrum of face *B*, then the synthesized image will be perceived as an image of face *B*. To improve the performance of feature extraction

and representation, we need to adopt the high-order statistics into the criterion.

ICA, as an extension of PCA, was proposed to this end and has been widely used in blind source separation, signal processing, medical image analysis, pattern recognition, and texture detection [5, 9, 12]. The objective of ICA is to seek for a better set of basis vectors so that the projection coefficients, which are called independent components (ICs), are statistically as independent as possible in the sense of high-order statistics other than the second-order statistic. Therefore, ICA can remove the high-order statistical dependencies to produce a more sparse and independent code that is useful for subsequent pattern discrimination [13].

By using ICA for face representation and recognition, Bartlett *et al* [5] found that much of the information that perceptually distinguishes faces is contained in the high-order statistics of images, and then they proposed two ICA architectures for face representation. Architecture I is to find a set of spatially independent basis vectors (images) and the coefficients that code each face are not necessarily independent. Architecture II treats the pixels as random variables and the images as outcomes, i.e. it uses ICA to find a set of basis images that make the projection coefficients of image be statistically independent. Their experimental results on FERET database showed that ICA is superior to PCA for face representation and recognition. Since then, a lot of face recognition methods have been developed to improve the representation performance and classification accuracy [14-19]. Liu [14] proposed an enhanced ICA method, which implements ICA in a reduced PCA space. The dimensionality of the PCA space is determined by balancing the representation criterion for adequate face representation and the magnitude criterion for enhanced retrieval performance. Pong *et al* [16] studied the relationship between the number of ICs and the classification accuracy and argued that not all ICs are useful for classification. They then proposed an IC selection algorithm for better classification. Bressan *et al* [17] discussed the selection of ICs and their classification ability by using the global and class-conditional

independence. All these algorithms claim that ICA is superior to PCA for face recognition.

By comparing detailedly the performance of PCA with ICA under different similarity metrics, however, Moghaddam [20] found that ICA is not always better than PCA. Yang *et al* [21] used PCA I and PCA II to evaluate the performance of ICA under Architecture I and Architecture II respectively. Experimental results showed that ICA is not always superior to PCA. Vicente *et al* [22] compared the performance of ICA with PCA and whitened PCA, and argued that ICA can have better recognition performance than PCA when a suitable feature selection step is employed for classification. Li *et al* [23] found that ICA is also superior to PCA in estimating the pose and multi-view subspace. In [24], Kim *et al* modified the kurtosis definition in ICA and proposed an LS-ICA algorithm by using Architecture I. The experimental results showed that LS-ICA performs better than PCA, especially in the case of partial occlusions and local distortions.

Most ICA based algorithms extract ICs via three principles: non-Gaussianity estimation, minimization of mutual information, and maximum likelihood [12]. All the three principles use a vectorized representation of the object. The face images have to be unfolded into 1-D vectors before applying these algorithms. However, the face images are more naturally represented as matrices (second-order tensor) or higher order arrays. The unfolded vectors may lose some structural information embedded in face images, which may be useful for recognition. Another problem is the high dimensionality of the unfolded vector (e.g. 10304 for a 112×92 image), which leads to the small sample size (SSS) problem of SAM: the number of available training samples is much less than the dimension of the underlying vector. This dilemma makes it very difficult to estimate accurately the statistics of the underlying face vector and the estimation error will then deteriorate the accuracy of face recognition.

Recently, matrix-based feature extraction has been attracting much attention in Biometric

authentication. Yang *et al* [7] proposed the two-dimensional PCA (2DPCA) scheme, which directly evaluates the basis vectors from image matrix rather than vectors. Ye [8] proposed the low rank representation of an image in the matrix form. Tao *et al* [25] and Kim *et al* [26] extended this idea to high-order matrix (n-array mode, n>2), and proposed some tensor subspace analysis methods. Compared with traditional SAM schemes, tensor subspace analysis can not only alleviate the SSS problem but also improve the classification accuracy. However, independent feature extraction on image matrix or tensor is rarely investigated yet. Recently, Vasilescu *et al* [27] proposed a multi-linear (tensor) ICA method which uses a tensor to represent the different factors, such as illumination, viewpoint, etc., in facial images. Nonetheless, image matrix is still stretched to a vector in their algorithm.

This paper will present a framework of matrix based ICA by analyzing the measure of non-Gaussianity of facial images. First we will present two new measurements of kurtosis to estimate the non-Gaussianity of vector variables, instead of scalar variables, and propose a novel method, which is called Matrix-FastICA, to calculate the demixing matrix of facial images. The main advantage of Matrix-FastICA is that it avoids transforming image matrix into vector. In order to reduce the feature storage space and apply Matrix-FastICA to higher dimensional array, a tensor ICA (T-ICA) algorithm is then proposed. Both Matrix-FastICA and T-ICA can preserve the local structural information embedded in the images and they alleviate greatly the SSS problem in traditional SAM schemes.

The remainder of this paper is organized as follows. Section 2 briefly reviews ICA. Section 3 presents two new measurements of kurtosis for vector variables and the Matrix-FastICA algorithms are then proposed. Feature extraction by using Matrix-FastICA is discussed in section 4. Section 5 extends Matrix-FastICA to T-ICA. Extensive experiments are performed in Section 6. Finally, the conclusion is made in Section 7.

2. Independent Component Analysis (ICA)

Denote by $\boldsymbol{x} = \left[x_1, x_2, \cdots, x_p\right]^T$ a p-dimensional vector (mix-signal), which is assumed to be a linear combination of several p-dimensional basis vectors $\boldsymbol{q}_i$ with unknown coefficients s_i. It is expected that the coefficients s_i are as statistically independent as possible and they are called the desired ICs of $\boldsymbol{x}$. The mixture model of ICA can be expressed as follows:

$$\boldsymbol{x} = s_1\boldsymbol{q}_1 + s_2\boldsymbol{q}_2 + \cdots + s_l\boldsymbol{q}_l = \boldsymbol{Q}\boldsymbol{s} \tag{1}$$

where $\boldsymbol{Q}$ is a $p \times l$ mixing matrix, $\boldsymbol{s} = \left[s_1, s_2, \cdots, s_l\right]$, $l \le p$, is the independent feature vector that is composed of the desired ICs and l is the number of ICs. The objective of ICA is to estimate the ICs from the observed data $\boldsymbol{x}$ by computing an $l \times p$ demixing matrix $\boldsymbol{W}^T$

$$\boldsymbol{u} = \boldsymbol{W}^T\boldsymbol{x} = \boldsymbol{W}^T\boldsymbol{Q}\boldsymbol{s} \tag{2}$$

such that the estimated IC vector $\boldsymbol{u}$ is a good approximation of the desired IC vector $\boldsymbol{s}$.

Many algorithms have been proposed to estimate the ICs of random vectors by using three different principles: non-Gaussianity estimation, minimization of mutual information, and maximum likelihood [12]. The FastICA algorithm, proposed by Hyvärinen [28], has been widely used in pattern recognition [12, 18, 23, 24]. It uses kurtosis to measure the non-Gaussianity and compute the demixing matrix $\boldsymbol{W}$. Denote by $\boldsymbol{w}$ the column vector of $\boldsymbol{W}$. The kurtosis of $\boldsymbol{w}^T\boldsymbol{x}$ is defined as [28]

$$kurt(\boldsymbol{w}^T\boldsymbol{x}) = E\left[\left(\boldsymbol{w}^T\boldsymbol{x}\right)^4\right] - 3\left(E\left[\left(\boldsymbol{w}^T\boldsymbol{x}\right)^2\right]\right)^2 \tag{3}$$

Let $u = \boldsymbol{w}^T\boldsymbol{x}$, we can see that u is a scalar variable, which represents the elements of $\boldsymbol{u}$. To make the components of $\boldsymbol{u}$ as independent as possible, $\boldsymbol{w}$ can be estimated by maximizing (3) under constraint

$$E\left[\left(\boldsymbol{w}^T\boldsymbol{x}\right)^2\right]=1 \tag{4}$$

$\boldsymbol{w}$ can be estimated iteratively as follows:

$$\boldsymbol{w}(t)=\Omega^{-1}E\left[\boldsymbol{x}\left(\boldsymbol{w}(t-1)^T\boldsymbol{x}\right)^3\right]-3\boldsymbol{w}(t-1) \tag{5}$$

$$\boldsymbol{w}(t)=\frac{\boldsymbol{w}(t)}{\sqrt{\boldsymbol{w}(t)\boldsymbol{H}\boldsymbol{w}(t)}} \tag{6}$$

where $\Omega=E\left[\boldsymbol{x}\boldsymbol{x}^T\right]$. Using (5) and (6), it is readily to calculate all orthogonal column vectors $\boldsymbol{w}_i$, $i=1,\cdots,l$, of $\boldsymbol{W}$ and then $\boldsymbol{W}=\left[\boldsymbol{w}_1 \quad \cdots \quad \boldsymbol{w}_l\right]$ is obtained. For more detailed information about FastICA, please refer to [28].

3. The Matrix-FastICA Algorithm

From Section 2 we can see that the FastICA algorithm calculates the demixing matrix $\boldsymbol{W}$ of the input signal $\boldsymbol{x}$ by maximizing its kurtosis, which is defined for the scalar variable $u=\boldsymbol{w}^T\boldsymbol{x}$. When we apply FastICA to facial images to extract the ICs, we need to stretch the face image into a vector $\boldsymbol{x}$. Therefore, some local structural information, which may be very useful for image representation and classification, may be lost, and the dimensionality of the stretched vector will be very high, which leads to the SSS problem of ICA. To solve those problems, in this section we propose two new definitions of kurtosis for vector variables, instead of a scalar variable as in the conventional FastICA algorithms. Two new ICA algorithms, called Matrix-FastICA I and II, will be consequently derived and they allow us to extract the demixing matrix and ICs directly from images without image-to-vector transformation.

Denote by $\boldsymbol{A}\in R^{m\times n}$ an $m\times n$ face image. Our goal is to find a demixing matrix $\boldsymbol{V}\in R^{n\times l}$ ($l\le n$) such that the independent features of $\boldsymbol{A}$, denoted by $\boldsymbol{S}$, can be directly obtained by

projecting $\boldsymbol{A}$ onto $\boldsymbol{V}$

$$\boldsymbol{S} = \boldsymbol{A}\boldsymbol{V} \tag{7}$$

where $\boldsymbol{S}$ is a $m \times l$ matrix. Each column of $\boldsymbol{S}$ is a m-dimensional vector, which is called the independent component of $\boldsymbol{A}$. Denote by $\boldsymbol{s}$ and $\boldsymbol{v}$, respectively, the column variables of $\boldsymbol{S}$ and $\boldsymbol{V}$ such that $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{v}$. To measure the non-Gaussianity of vector variable $\boldsymbol{s}$, we define two new forms of kurtosis on $\boldsymbol{s}$ as follows

Definition 1

$$\begin{aligned} kurt(\boldsymbol{s}) &= E\left\{\left(\boldsymbol{s}^T\boldsymbol{s}\right)^2\right\} - 3\left(E\left\{\boldsymbol{s}^T\boldsymbol{s}\right\}\right)^2 \\ &= E\left\{\left((\boldsymbol{A}\boldsymbol{v})^T(\boldsymbol{A}\boldsymbol{v})\right)^2\right\} - 3\left(E\left\{(\boldsymbol{A}\boldsymbol{v})^T(\boldsymbol{A}\boldsymbol{v})\right\}\right)^2 \end{aligned} \tag{8}$$

Definition 2

$$\begin{aligned} kurt(\boldsymbol{s}) &= trace\left[E\left\{\left(\boldsymbol{s}\cdot\boldsymbol{s}^T\right)^2\right\} - 3\left(E\left\{\boldsymbol{s}\cdot\boldsymbol{s}^T\right\}\right)^2\right] \\ &= trace\left[E\left\{\left((\boldsymbol{A}\boldsymbol{v})(\boldsymbol{A}\boldsymbol{v})^T\right)^2\right\} - 3\left(E\left\{(\boldsymbol{A}\boldsymbol{v})(\boldsymbol{A}\boldsymbol{v})^T\right\}\right)^2\right] \end{aligned} \tag{9}$$

In definition 1, the non-Gaussianity of $\boldsymbol{s}$ is actually measured by using its inner product, while in definition 2, the high-order cross correlations between the elements in $\boldsymbol{s}$ are also considered in calculating the non-Gaussianity of $\boldsymbol{s}$. Definition 2 is more complex than Definition 1 and then needs more computation.

In order to solve $\boldsymbol{v}$, the column vector of the desired demixing matrix $\boldsymbol{V}$, in the case of definition 1 we could maximize the kurtosis (8) under the constraint

$$E\left\{(\boldsymbol{A}\boldsymbol{v})^T(\boldsymbol{A}\boldsymbol{v})\right\} = 1 \tag{10}$$

and in the case of definition 2 we could maximize kurtosis (9) under the constraint

$$E\left\{(\boldsymbol{A}\boldsymbol{v})(\boldsymbol{A}\boldsymbol{v})^T\right\} = \frac{1}{M}\boldsymbol{H} \tag{11}$$

where $\boldsymbol{H}$ is a $m \times m$ matrix whose elements are all 1 and $M = m \times m$ is a constant.

By calculating the demixing matrices based on the above two definitions, we have the following two Matrix-FastICA algorithms accordingly.

A. Matrix-FastICA algorithm I -- By introducing a Lagrangian coefficient λ to (8), the optimal solution $\boldsymbol{v}$ to (7) under constraint (10) can be obtained by maximizing the following objective function

$$kurt(\boldsymbol{Av}) = E\left\{\left((\boldsymbol{Av})^T(\boldsymbol{Av})\right)^2\right\} - 3\left(E\left\{(\boldsymbol{Av})^T(\boldsymbol{Av})\right\}\right)^2 + \lambda\left(1 - E\left[(\boldsymbol{Av})^T(\boldsymbol{Av})\right]\right) \tag{12}$$

After some calculation, we can derive that

$$\boldsymbol{v} = \frac{2}{\lambda}\left[E\left\{\boldsymbol{v}^T\boldsymbol{A}^T\boldsymbol{Av}\boldsymbol{A}^T\boldsymbol{Av}\right\} - 3\boldsymbol{v}\right] \tag{13}$$

Please refer to **Appendices A** and **B** for the detailed derivation process of (13).

Equation (13) is not an analytic form of $\boldsymbol{v}$. Similar to FastICA, $\boldsymbol{v}$ can be calculated iteratively by using the following update rules:

$$\boldsymbol{v}^*(t) = E\left\{\left(\boldsymbol{v}(t-1)\right)^T\boldsymbol{A}^T\boldsymbol{Av}(t-1)\boldsymbol{A}^T\boldsymbol{Av}(t-1)\right\} - 3\boldsymbol{v}(t-1) \tag{14}$$

$$\boldsymbol{v}(t) = \frac{\boldsymbol{v}^*(t)}{\sqrt{\boldsymbol{v}^*(t)^T\boldsymbol{v}^*(t)}} \tag{15}$$

Once we calculate the first column vector $\boldsymbol{v}_1$ by using (14)~(15), the second column vector $\boldsymbol{v}_2$ can be then calculated by using (14)~(15) under the condition that it is orthogonal to $\boldsymbol{v}_1$. Similarly, more column vectors can be computed and finally the demixing matrix $\boldsymbol{V}$ can be obtained as $\boldsymbol{V} = \left[\boldsymbol{v}_1 \quad \cdots \quad \boldsymbol{v}_n\right]$.

B. Matrix-FastICA algorithm II -- By introducing a Lagrangian coefficient λ to (9), the

optimal solution $\boldsymbol{v}$ to (7) under constraint (11) can be obtained by maximizing the following objective function

$$\begin{aligned} kurt(\boldsymbol{Av}) = trace\Big[& E\left\{\left((\boldsymbol{Av})(\boldsymbol{Av})^T\right)^2\right\} - 3\left(E\left\{(\boldsymbol{Av})(\boldsymbol{Av})^T\right\}\right)^2\Big] \\ & + \lambda\left(1 - trace\left\{E\left[(\boldsymbol{Av})(\boldsymbol{Av})^T\right]\right\}\right) \end{aligned} \tag{16}$$

After some calculation, we can derive that

$$\boldsymbol{v} = \frac{2}{\lambda}\left[E\left\{\boldsymbol{A}^T\boldsymbol{Av}(\boldsymbol{v})^T\boldsymbol{A}^T\boldsymbol{Av}\right\} - 3F(\boldsymbol{v})\right] \tag{17}$$

where $F(\boldsymbol{v}) = \frac{1}{m\times m}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\{E\left[\boldsymbol{A}^T\boldsymbol{E}_{ij}\boldsymbol{Av}\right]\right\}$ and $\boldsymbol{E}_{ij}$ denotes a matrix whose element at the ith row and the jth column is 1 and all other elements are zero.

Please refer to **Appendices A** and **C** for the detailed derivation process of (17).

As in algorithm I, $\boldsymbol{v}$ can be computed iteratively by using the following rules:

$$\boldsymbol{v}^*(t) = E\left\{\boldsymbol{A}^T\boldsymbol{Av}(t-1)\left(\boldsymbol{v}(t-1)\right)^T\boldsymbol{A}^T\boldsymbol{Av}(t-1)\right\} - 3F\left(\boldsymbol{v}(t-1)\right) \tag{18}$$

$$\boldsymbol{v}(t) = \frac{\boldsymbol{v}^*(t)}{\sqrt{\boldsymbol{v}^*(t)^T\boldsymbol{v}^*(t)}} \tag{19}$$

By calculating the vectors one by one as in algorithm I, we can get the desired demixing matrix $\boldsymbol{V}$ as $\boldsymbol{V} = \left[\boldsymbol{v}_1 \quad \cdots \quad \boldsymbol{v}_m\right]$.

4. Feature Extraction and Classification using Matrix-FastICA

Given N training face images $\boldsymbol{A}_i \in R^{m\times n}$ ($i = 1,2,\cdots,N$) and suppose that $\boldsymbol{V} \in R^{n\times l}$ ($l \le n$) is the demixing matrix obtained by the Matrix-FastICA algorithms developed in Section 3, it is easy to obtain the ICs of image $\boldsymbol{A}_i$ by projecting it onto $\boldsymbol{V}$. In practice, however, we will whiten the

facial image dataset before applying Matrix-FastICA in order to reduce the computational load and improve the performance of classification [5, 14-15, 22].

The whitened image $\boldsymbol{X}_i$ of a face image $\boldsymbol{A}_i$ is calculated by

$$\boldsymbol{X}_i = \left(\boldsymbol{A}_i - \overline{\boldsymbol{A}}\right)\boldsymbol{W}_1 \tag{20}$$

where $\overline{\boldsymbol{A}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{A}_i$ is the mean of training images. $\boldsymbol{W}_1$ is the whiten matrix of the training images, which is calculated by

$$\boldsymbol{W}_1 = \boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}} \tag{21}$$

where $\boldsymbol{P}$ is composed of the eigenvectors of the covariance matrix $\boldsymbol{G} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{A}_i - \overline{\boldsymbol{A}}\right)^T\left(\boldsymbol{A}_i - \overline{\boldsymbol{A}}\right)$ of the training dataset and $\boldsymbol{\Lambda}$ is a diagonal matrix with the diagonal element being the eigenvalues of $\boldsymbol{G}$.

Applying the proposed Matrix-FastICA algorithms to the whitened data $\boldsymbol{X}_i$, it is readily to get the demixing matrix $\boldsymbol{V}_1 \in R^{n\times l}$. The independent features, denoted by $\boldsymbol{S}_i$, of $\boldsymbol{X}_i$ can be obtained by projecting $\boldsymbol{X}_i$ onto $\boldsymbol{V}_1$

$$\boldsymbol{S}_i = \left[\boldsymbol{s}_1^i\ \boldsymbol{s}_2^i \cdots \boldsymbol{s}_l^i\right] = \boldsymbol{X}_i\boldsymbol{V}_1 \tag{22}$$

Substituting (20) into (22), we have

$$\boldsymbol{S}_i = \left(\boldsymbol{A}_i - \overline{\boldsymbol{A}}\right)\times\boldsymbol{W}_1\times\boldsymbol{V}_1 \tag{23}$$

For the convenience of expression and the consistency of the ICA model, we denote by $\boldsymbol{V}$ the product of $\boldsymbol{W}_1$ and $\boldsymbol{V}_1$ and call it the demixing matrix of the original dataset, i.e. $\boldsymbol{V} = \boldsymbol{W}_1\times\boldsymbol{V}_1$, then $\boldsymbol{S}_i = \left(\boldsymbol{A}_i - \overline{\boldsymbol{A}}\right)\times\boldsymbol{V}$.

In ICA, however, the discriminability of each column in the demixing matrix $\boldsymbol{V}$ is not known in prior. In order to improve the classification accuracy and reduce the dimensionality of features, we measure the discriminability of each column of $\boldsymbol{V}$ via class discriminability [5]

$$r = \frac{\sigma_b}{\sigma_w} \tag{24}$$

where $\sigma_b = \sum_j \left(\bar{\boldsymbol{c}}_j - \bar{\boldsymbol{c}}\right)^T \left(\bar{\boldsymbol{c}}_j - \bar{\boldsymbol{c}}\right)$ and $\sigma_w = \sum_j \sum_{\boldsymbol{s} \in c_j} \left(\boldsymbol{s} - \bar{\boldsymbol{c}}_j\right)^T \left(\boldsymbol{s} - \bar{\boldsymbol{c}}_j\right)$ represent the between-class variability and the within-class variability, respectively, of the projection coefficients of the training images. $\bar{\boldsymbol{c}}$ denotes the global mean of the training images and $\bar{\boldsymbol{c}}_j$ denotes the mean of the images in the jth class. $\boldsymbol{s}$ denotes the ICs of the images. Based on the magnitude of r, the discriminabilities of the columns of $\boldsymbol{V}$ can be measured and then a sub-matrix of $\boldsymbol{V}$, denoted by $\boldsymbol{V}_r \in R^{n \times k}$, can be determined, where k is the number of selected columns for feature extraction.

After obtaining the reduced demixing matrix $\boldsymbol{V}_r$, we can get the reduced independent features $\boldsymbol{S}_i$ from each face image $\boldsymbol{A}_i$. Then given a probe face image $\boldsymbol{A}^* \in R^{m \times n}$, its independent features S^* can be obtained by projecting it onto $\boldsymbol{V}_r$. The task of identification can be realized by using the nearest neighbor classifier, which calculates the similarity measurement δ between $\boldsymbol{S}^*$ and $\boldsymbol{S}_i$. In this paper, the Euclidean distance (L_2) is used to measure the similarity, which is defined as

$$\delta_{L_2}\left(\boldsymbol{S}^*, \boldsymbol{S}_i\right) = \sqrt{\sum_{j=1}^{k} \left\| \boldsymbol{s}_j^* - \boldsymbol{s}_j^i \right\|} \tag{25}$$

where $\boldsymbol{s}_j^*$ and $\boldsymbol{s}_j^i$ denote the jth column of feature matrices $\boldsymbol{S}^*$ and $\boldsymbol{S}_i$, respectively. Note that $\boldsymbol{s}_j^*$ or $\boldsymbol{s}_j^i$ is an m-dimensional vector, i.e. the ICs extracted by Matrix-FastICA are vectors rather

than scalars as in the conventional ICA. As we can see in the experiments, though Matrix-FastICA can achieve much higher recognition accuracy than conventional ICA schemes, it may require a large feature storage space. To overcome this shortcoming, the tensor independent component analysis introduced next can be a good solution to reducing the storage space while keeping high recognition accuracy.

5. Extension of Matrix-FastICA to Tensor ICA

Although Matrix-FastICA can extract independent features directly from image matrices without image-to-vector stretching, it may need a large feature storage space because each feature in Matrix-FastICA is an *m*-dimensional vector. In addition, the natural images may be represented in a form of high-order tensors, e.g. color image as third-order tensor. Recently, tensor-based subspace analysis has been extracting more and more attention in biometric authentication [25, 26]. In order to overcome the problem of Matrix-FastICA and extend it to tensor subspace analysis, in this section we propose a tensor independent component analysis (TICA) technique, which estimates the demixing matrix by using the previously developed Matrix-FastICA algorithms.

Before presenting the TICA algorithm, we first briefly review some basic terminologies and concepts of tensor analysis in section 5.1. For more detailed information about tensor algebra, please refer to [25-26, 29-30].

5.1 Tensor algebra

Denote by $\mathcal{A}$ a tensor of size $I_1 \times I_2 \times \cdots \times I_k \times \cdots \times I_K$. The order of $\mathcal{A}$ is K and the kth dimension (or mode) of $\mathcal{A}$ is of size I_k. An element of $\mathcal{A}$ is denoted by $\mathcal{A}_{i_1, i_2, \ldots, i_K}$, where

$1 \le i_k \le I_k$ and $1 \le k \le K$. We have the following definitions [29-30].

Definition 3. (Mode-k Matrixizing or Matrix Unfolding) The mode-k matrixizing or matrix unfolding of a Kth order tensor $\mathcal{A}$ is a matrix $\boldsymbol{D}_{(k)} \in R^{I_k \times I_{\bar{k}}}$, $I_{\bar{k}} = \left(\prod_{j \ne k} I_j\right)$, which is the ensemble of vectors in space R^{I_k} obtained by keeping index i_k fixed and varying the other indices. We call $\boldsymbol{D}_{(k)}$ the mode-k matrixizing of $\mathcal{A}$.

Definition 4. (Mode-k product) The mode-k product $\mathcal{A} \times_k \boldsymbol{U}$ of a tensor $\mathcal{A}$ and a matrix $\boldsymbol{U} \in \mathrm{R}^{I_k \times I_k'}$ is an $I_1 \times I_2 \times \ldots \times I_{k-1} \times I_k' \times I_{k+1} \times \ldots \times I_K$ tensor defined by

$$\left(\mathcal{A} \times_k \boldsymbol{U}\right)_{i_1 \times i_2 \times \ldots \times i_{k-1} \times j \times i_{k+1} \times \ldots \times i_K} = \sum_{i_k} \left(\mathcal{A}_{i_1 \times i_2 \times \ldots \times i_{k-1} \times i_k \times i_{k+1} \ldots \times i_K} \boldsymbol{U}_{j \times i_k}\right) \tag{26}$$

for all index values. The mode-k product is a type of contraction.

5.2 TICA by using Matrix-FastICA

Given an arbitrary K-array ($K > 2$) matrix, i.e. tensor, $\mathcal{A} \in R^{I_1 \times I_2 \times \cdots \times I_K}$, it can be expressed as follows [29]

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_K \boldsymbol{U}_K \quad \text{and} \quad \mathcal{S} = \mathcal{A} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_K \boldsymbol{U}_K \tag{27}$$

where tensor $\mathcal{S}$, called the core tensor, governs the interaction between the mode matrices $\boldsymbol{U}_k$ for $k = 1, \ldots, K$. Matrix $\boldsymbol{U}_k$ contains the orthogonal vectors spanning the column space of matrix $\boldsymbol{D}_{(k)}$ that resulted from the mode-k flatting of $\mathcal{A}$. The goal of the TICA to be developed is to find K transformation matrices $\boldsymbol{U}_k \in R^{L_k \times I_k}$ ($L_k < I_k$, $k = 1, 2, \cdots, K$) such that the columns of $\mathcal{S}$ are as independent as possible by maximizing the non-Gaussianity of the input data. Here we use the previously developed Matrix-FastICA algorithms to evaluate and

maximize the non-Gaussianity.

Note that direct computation of the transformation matrices $\boldsymbol{U}_k$ ($k = 1, \cdots, K$) is infeasible [25, 26, 29, 30]. In general, this problem can be solved approximately by employing an iterative scheme which was originally proposed for low-rank approximation of the second-order tensors and was later extended for high-order tensors [25-26, 29]. In what follows, we discuss how to compute the transformation matrix $\boldsymbol{U}_k$ in (27) by using such an iterative scheme.

Assuming that $\boldsymbol{U}_1, ..., \boldsymbol{U}_{k-1}, \boldsymbol{U}_{k+1}, ..., \boldsymbol{U}_K$ are known, we denote by $\mathcal{Y}_k$ the tensor

$$\mathcal{Y}_k = \mathcal{A} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_{k-1} \boldsymbol{U}_{k-1} \times_{k+1} \boldsymbol{U}_{k+1} \cdots \times_K \boldsymbol{U}_K \tag{28}$$

Denote by $\boldsymbol{Y}_{(k)}$ the corresponding k-mode matrix unfolding of $\mathcal{Y}_k$ and by $\mathcal{S}_k$ the tensor

$$\mathcal{S}_k = \mathcal{Y}_k \times_k \boldsymbol{U}_k \tag{29}$$

Denote by $\boldsymbol{S}_{(k)}$ the corresponding k-mode unfolding of $\mathcal{S}_k$. Then (29) can be expressed as

$$\boldsymbol{S}_{(k)} = \boldsymbol{U}_k \boldsymbol{Y}_{(k)} \tag{30}$$

Now the problem is to seek for a $\boldsymbol{U}_k$ such that the columns of $\boldsymbol{S}_{(k)}$, which are the ICs of $\boldsymbol{Y}_{(k)}$, are as independent as possible. $\boldsymbol{Y}_{(k)}$, instead of the face images $\mathcal{A}$ in the original dataset, are viewed as the available training samples now. After such an operation, the Matrix-FastICA algorithms developed in Sections 3 and 4 can be used to calculate the demixing matrix $\boldsymbol{U}_k$ of $\boldsymbol{Y}_{(k)}$. The other transformation matrices $\boldsymbol{U}_i$ ($i \neq k$) can be estimated using the similar procedure.

5.3 TICA algorithm

The TICA algorithm described in Section 5.2 can be summarized as follows:

Step 1. Input the training dataset $\mathcal{A}_i \in R^{I_1 \times I_2 \times \cdots \times I_K}$, $i = 1, 2, \cdots, N$. Set the dimensionality of the

output tensor $\mathcal{S}_i \in R^{l_1 \times l_2 \times \cdots \times l_K}$ and set the maximum training iteration number $T_{\max}$.

Step 2. Initialize $\boldsymbol{U}_1^0 = \boldsymbol{I}_{l_1}, \boldsymbol{U}_2^0 = \boldsymbol{I}_{l_2}, ..., \boldsymbol{U}_K^0 = \boldsymbol{I}_{l_K}$, where $\boldsymbol{I}_{l_j}$ denote the $I_j \times l_j$ unit matrix.

Step 3. Training iteration.

for $t = 1, 2, \cdots, T_{\max}$

for $k = 1, 2, \cdots, K$

Calculating $\mathcal{Y}_k^i = \mathcal{A}_i \times_1 \boldsymbol{U}_1^t \cdots \times_{k-1} \boldsymbol{U}_{k-1}^t \times_{k+1} \boldsymbol{U}_{k+1}^t \cdots \times_K \boldsymbol{U}_K^t$

Calculating $\boldsymbol{Y}_{(k)}^i \leftarrow \mathcal{Y}_k^i$ via definition (30)

Calculating $\boldsymbol{U}_k^t$ by using (14)-(15) or (18)-(19).

end

end

Step 4. Extracting ICs $\mathcal{S}_i = \mathcal{A}_i \times_1 \boldsymbol{U}_1 \cdots \times_K \boldsymbol{U}_K$.

Step 5. Extracting ICs of probe image $\mathcal{A}^*$: $\mathcal{S}^* = \mathcal{A}^* \times_1 \boldsymbol{U}_1 \cdots \times_K \boldsymbol{U}_K$.

Step 6. Classification based on the similarity between $\mathcal{S}_i$ and $\mathcal{S}^*$.

6. Experimental Results

In this section, we will verify the performance of the proposed Matrix-FastICA and T-ICA algorithms in comparison with some popular unsupervised SAM schemes, including ICA [5], PCA [4], 2DPCA [7], R-ICA[2] and popular supervised SAM schemes FLD [6] and 2DFLD [31]. For the convenience of expression, we denote by MICA-1 the proposed Matrix-FastICA algorithm 1, by MICA-2 the proposed Matrix-FastICA algorithm 2, by TICA-1 the proposed

[2] Here R-ICA refers to that we take the rows of images as training samples, and then perform ICA on these samples to compute the demixing matrix. The procedures are similar to those in 2DPCA.

TICA with MICA-1 and by TICA-2 the proposed TICA with MICA-2.

Four well-known face databases are used here: the Yale database [32] is used to evaluate the performance of the proposed methods under illumination and expression variations; the FERET database [33] is used to test the performance under expression and pose variations; the UMIST database [34] is used to test the performance under pose and rotation variations; and the ORL database is used to evaluate the performance under variations of pose and scale [35]. In the Yale database, the original images were normalized (in scale and orientation) such that the two eyes are aligned at the same position. The facial images were then cropped (32×32) for training and matching in the experiments[3]. For the FERET database, the facial portion of each image was manually cropped and then normalized to 80×80. For the other databases, no further preprocessing was made.

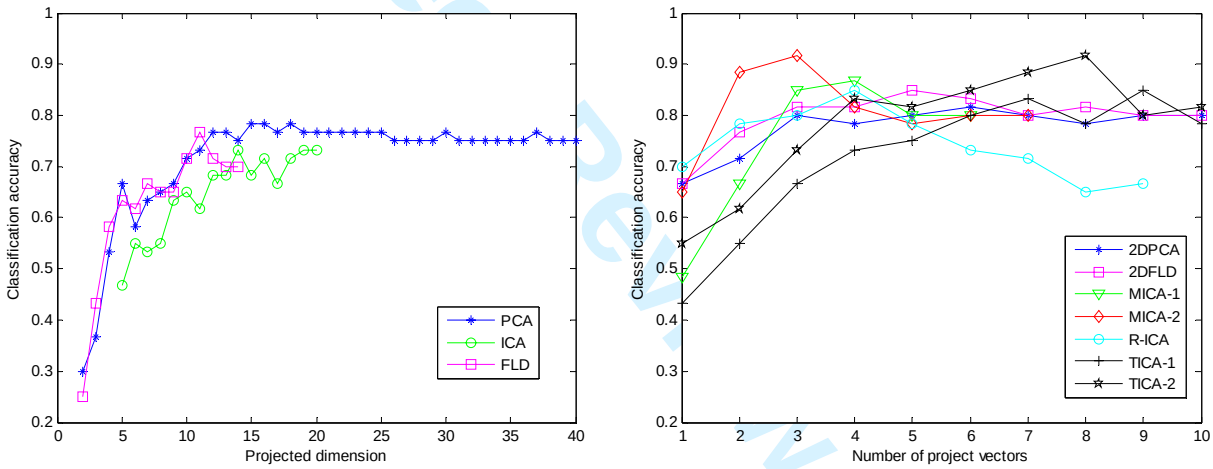
6.1 Yale database

The Yale face database was established at the Yale Center [32]. It consists of images from 15 different persons, with 11 images each person, and has 165 images in total. The images demonstrate variations in lighting conditions and facial expressions. Therefore this database is used to evaluate the performance of the proposed methods under the variations of illumination and expression. Fig.1 shows some cropped images of one subject in the Yale database.

Figure 1. Some sample images of one subject in the Yale database.

[3] We thank Dr. He Xiaofei for providing the cropped images.

A subset with the first seven images per individual was used as the training set. The rest images were used for testing. Thus, the training and testing datasets have 105 and 60 images respectively. The different schemes were applied to the training images to compute the projection matrices, which were then used to project the testing images onto lower dimensional subspaces for classification. The classification was performed by using the nearest-neighbor classifier. The recognition accuracy varies with the number of dimensions of the extracted features. Fig. 2 plots the curves of recognition accuracy versus number of features by different schemes. In Table 1 we listed the top recognition accuracies of different schemes.



(a) (b)

Figure 2. The recognition accuracies of different vector-based representation and matrix-based representation algorithms on the Yale database. (a) Vector-based algorithms, i.e. PCA, FLD, ICA; (b) matrix-based algorithms, i.e. 2DPCA, 2DFLD, R-ICA, MICA-1, MICA-2, TICA-1, and TICA-2.

Table 1. Top recognition accuracies (%) and the associated dimensionalities on the Yale database by different schemes.

Method	PCA	ICA	FLD	2DPCA	2DFLD	R-ICA	MICA-1	MICA-2	TICA-1	TICA-2
Accuracy	78.33	68.33	76.67	81.67	85.00	85.00	86.67	91.67	90.00	91.67
Dimension	30	26	11	192	160	128	128	96	27	32

From Fig. 2 we can see that the proposed MICA and TICA schemes, especially MICA-2 and TICA-2, can achieve much higher accuracies than the other schemes, even including the supervised learning techniques FLD and 2D-FLD. From Table 1, we see that the top accuracies

of PCA, FLD, ICA, 2DPCA, 2DFLD, R-ICA, MICA-1, MICA-2, TICA-1 and TICA-2 are 78.33%, 76.67%, 68.33%, 81.67%, 85.00%, 85.00%, 86.67%, 91.67%, 90% and 91.67%, respectively. The associated numbers of features, i.e. dimensions, with those recognition accuracies are 30, 11, 19, 192, 160, 128, 128, 96, 27 and 32, respectively. We see that MICA and TICA schemes achieve higher accuracy than other schemes. MICA-2 has better recognition accuracy than that of MICA-1 and TICA-2 works better than TICA-1. On this database, the classification accuracy of TICA is better than the associated MICA scheme. Particularly, the TICA schemes need only a small amount of features while having high recognition accuracy.

Table 2. Top recognition accuracies (%) on the Yale database by different schemes. The values in the parentheses are the associated dimensionality of features.

Training number	1	2	3	4	5	6	7	8	9
PCA	32 (11)	52.59 (25)	59.17 (40)	63.81 (47)	66.67 (14)	66.67 (40)	78.33 (30)	82.22 (26)	86.67 (46)
2DPCA	34.67 (128)	57.04 (128)	61.67 (128)	68.57 (96)	68.57 (128)	72.22 (128)	81.67 (192)	86.67 (128)	86.67 (96)
FLD	-------	49.63 (3)	61.67 (6)	62.8571 (13)	68.89 (12)	70.67 (10)	76.67 (11)	86.67 (12)	76.67 (10)
2DFLD	-------	46.67 (64)	56.67 (224)	63.81 (96)	73.33 (96)	73.33 (128)	85.00 (160)	88.89 (160)	86.67 (96)
ICA	35.33 (11)	56.30 (28)	54.17 (30)	56.19 (27)	61.11 (20)	64.00 (19)	73.33 (19)	77.78 (22)	83.33 (19)
R-ICA	33.33 (96)	56.30 (64)	65.00 (64)	69.52 (128)	76.00 (64)	73.33 (64)	85.00 (128)	84.44 (96)	83.33 (64)
MICA-1	36.00 (96)	57.78 (128)	61.67 (96)	71.43 (96)	72.22 (96)	70.67 (96)	86.67 (128)	88.89 (128)	90.00 (128)
MICA-2	38.67 (32)	57.78 (96)	65.83 (64)	76.19 (96)	78.89 (96)	78.67 (96)	91.67 (96)	93.33 (96)	93.33 (96)
TICA-1	43.33 (12)	59.17 (22)	66.67 (20)	70.48 (24)	71.11 (24)	70.67 (20)	90.00 (32)	84.44 (28)	90.00 (26)
TICA-2	48.00 (12)	59.78 (20)	66.67 (24)	76.19 (21)	77.78 (24)	80.00 (24)	91.67 (32)	91.11 (21)	93.33 (4*8)
Gabor-MICA-1	55.33 (7680)	71.11 (5120)	82.50 (2560)	80.95 (3840)	85.56 (6400)	86.67 (6400)	93.33 (2560)	95.56 (2560)	96.67 (2560)
Gabor-MICA-2	56.00 (6400)	72.59 (2560)	84.17 (2560)	81.90 (2560)	88.89 (3840)	89.33 (3860)	93.33 (2560)	97.78 (5120)	100 (3840)
Gabor-TICA-1	54.00 (640)	74.07 (480)	85.83 (320)	83.81 (640)	88.89 (960)	88.00 (960)	93.33 (640)	97.78 (1000)	100 (600)
Gabor-TICA-2	54.67 (800)	76.29 (480)	85.83 (480)	83.81 (600)	91.11 (720)	93.33 (480)	96.67 (600)	100 (720)	100 (480)

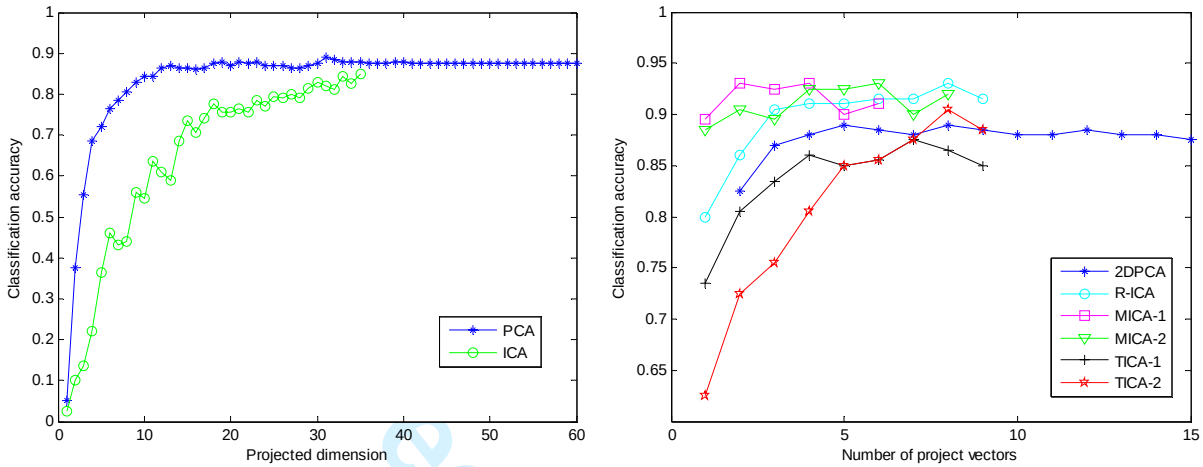
In order to evaluate more sufficiently the performance of the proposed methods in comparison with other algorithms, we made the following experiments. We used the first 1 to 9 images per person for training and used the remaining images for testing. Table 2 lists the best recognition accuracies of different algorithms and the associated numbers of features. The Gabor-MICA and Gabor-TICA refer to the algorithms that applying MICA and TICA to the Gabor filtering enhanced images respectively, like in the Gabor filter based ICA scheme proposed by Liu *et al* [17]. In the experiments, 5 scales and 8 directions are used as in [17]. It can be seen that Gabor-TICA and Gabor-MICA algorithms have much better results than the other schemes, and the Gabor-TICA methods achieve the best classification accuracies but with the highest numbers of features.

6.2 FERET database

The partial FERET face database used in this section comprises 400 gray-level frontal view face images from 200 persons. Each person has two images (**fa** and **fb**), which were obtained at different times and with different facial expressions. All the images were cropped manually to the size of 80×80. In the experiment, the **fa** images were used as gallery for training while the **fb** images as probes for testing.

Fig. 3 plots the recognition accuracies by different methods under different numbers of features. Table 3 lists the top recognition accuracy results by different methods and the associated dimensions of features. As can be seen from Fig. 3 and Table 3, the MICA schemes have better results than the other methods. Though R-ICA and MICA methods have higher top recognition accuracies than TICA, they require much higher dimensions of feature than TICA. By using Gabor filters to enhance the face images, the Gabor-TICA schemes achieve the best classification

accuracies among all the algorithms but with the largest numbers of features. This is consistent with the results on the Yale database.



(a) (b)

Figure 3. The recognition accuracies of different vector-based representation and matrix-based representation algorithms on the FERET fafb database. (a) Vector-based algorithms, i.e. PCA, and ICA; (b) matrix-based algorithms, i.e. 2DPCA, R-ICA, MICA-1, MICA-2, TICA-1 and TICA-2.

Table 3. Top recognition accuracies (%) and the associated dimensionalities on the FERET fafb database by different schemes.

Methods	PCA	2DPCA	ICA	R-ICA	MICA-1	MICA-2	TICA-1	TICA-2	Gabor-ICA-1	Gabor-TICA-2
Accuracy	89.00	89.00	85.00	93.00	93.00	93.00	87.50	90.50	96.00	95.50
Dimension	31	400	35	640	480	560	42	40	800	720

6.3 ORL database

The ORL database contains samples from 40 individuals, each providing 10 sample images. For some subjects, the images were taken at different times. The facial expressions (open or closed eyes, smiling or non-smiling) and occlusion (glasses or no glasses) also vary. The images were taken with a tolerance for tilting and rotation up to 20 degrees. There is also some variation in the scale of up to 10 percent. All images are grayscale and normalized to a resolution of 112×92 pixels. Fig. 4 shows five sample images of one subject in the ORL database.

Figure 4. Some images of one subject in the ORL database.

In the experiments, the first 1 to 5 images per person were used to form the training set and the remaining images were used for testing. Fig. 5 shows the classification accuracy versus number of features by different schemes when five images per person were selected for training, and the remaining images for testing. Table 4 lists the top recognition results of the different methods and the associated dimensions of features. As can be seen from Fig. 5 and Table 4, the MICA and TICA schemes have significantly better classification performance than PCA, ICA and R-ICA schemes, and achieve better or similar results to 2DPCA and FLD methods. TICA outperforms MICA in classification and has less number of features. The recognition accuracy of TICA-2 is higher than 2DPCA and the supervised FLD and only slightly less than 2DFLD. Again, the Gabor filter enhanced Gabor-TICA methods have the best classification accuracy among all the algorithms.

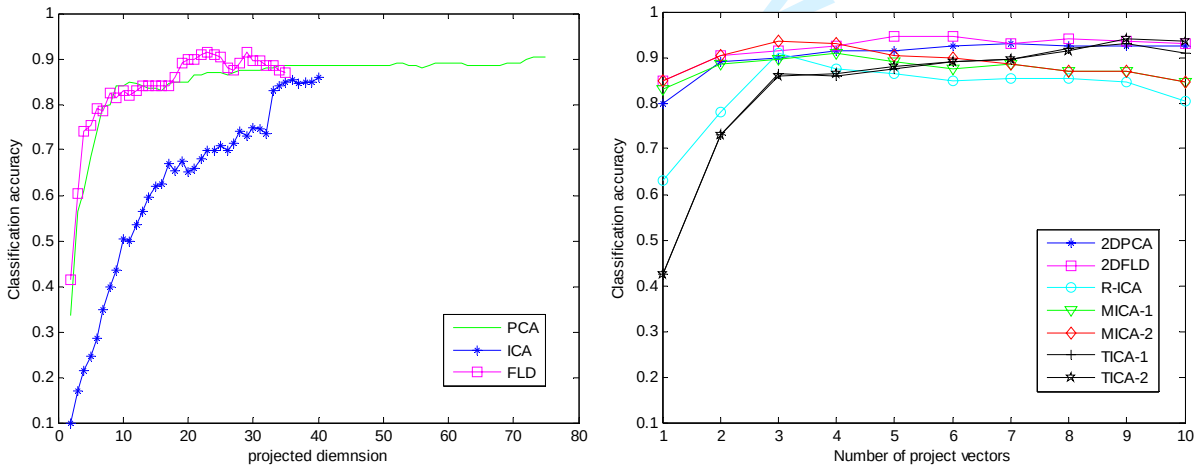


Figure 5. The recognition accuracies of different vector-based representation and matrix-based representation algorithms on the ORL database. (a) Vector-based algorithms, i.e. PCA, and ICA; (b) matrix-based algorithms, i.e. 2DPCA, R-ICA, MICA-1, MICA-2, TICA-1 and TICA-2.

Table 4. Top recognition accuracies (%) and the associated dimensionalities on the ORL database by different schemes. The values in the parentheses are the associated dimensionality of features.

Training number	1	2	3	4	5
PCA	70.00(36)	82.50(68)	85.00(56)	88.33(65)	90.50(73)
2DPCA	75.56(560)	86.88(448)	88.57(672)	91.67(672)	93.00(560)
ICA	64.44(10)	76.56(35)	78.93(40)	82.92(50)	86.50(40)
FLD	----------	81.25(33)	86.43(36)	90.00(29)	91.50(23)
2DFLD	----------	88.13(224)	91.07(448)	94.17(560)	94.50(560)
R-ICA	74.44(224)	85.31(224)	86.79(224)	89.58(336)	91.00(336)
MICA-1	75.28(224)	85.00(224)	86.79(224)	89.58(448)	91.00(448)
MICA-2	75.28(224)	85.00(224)	88.93(336)	92.92(336)	93.50(336)
TICA-1	75.56(10)	86.25(30)	88.93(24)	91.25(25)	93.00(27)
TICA-2	75.56(18)	87.81(32)	89.29(21)	92.50(30)	94.00(27)
Gabor-TICA-1	85.83(600)	92.81(640)	93.93(600)	97.50(1120)	98.00(960)
Gabor-TICA-2	85.00(720)	93.44(600)	94.29(600)	97.92(600)	98.00(640)

6.4 UMIST database

The UMIST database was established at the University of Manchester Institute of Science and Technology [34]. It is a multi-view database, consisting of 575 images from 20 people and covering a wide range of poses from profile to frontal views. The image size was cropped to 112×92. Fig. 6 shows some sample images of one subject.

In the experiments, we used the first 1, 3, 6, 9 images per person for training and used the remaining images for testing. Table 5 shows the top classification accuracies of different algorithms and the associated numbers of features. Fig. 7 plots the classification accuracy versus number of features by different schemes when 9 images per person are selected for training, and the remaining images for testing. It can be seen from Table 5 and Fig. 7 that the proposed MICA and TICA schemes, especially the MICA-2 and TICA-2 methods, perform much than the other unsupervised algorithms and even the supervised 2DFLD algorithm.

Figure 6. Some images for one subject of UMIST database

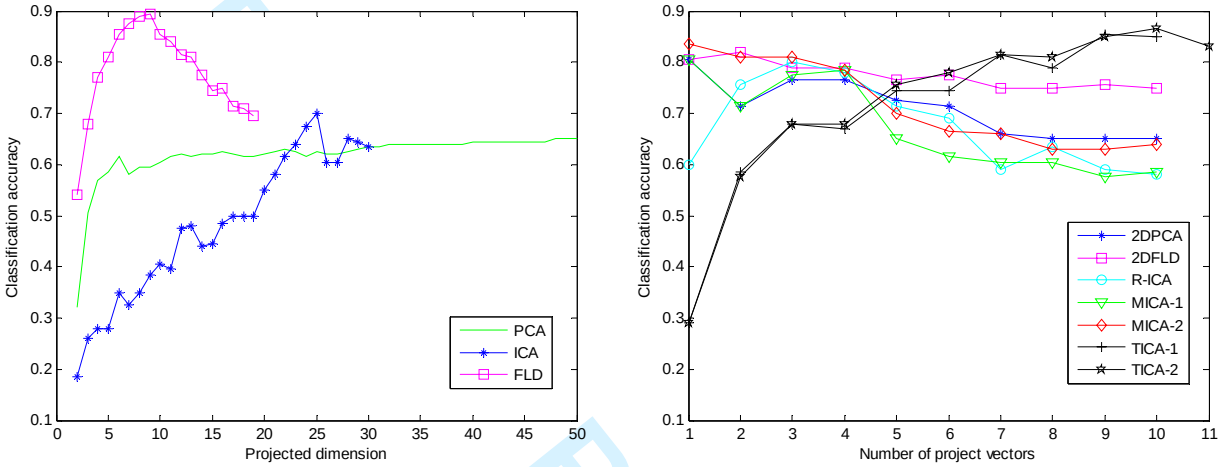


Figure 7. The recognition accuracies of different vector-based representation and matrix-based representation algorithms on the UMIST database. (a) Vector-based algorithms, i.e. PCA, FLD, ICA; (b) matrix-based algorithms, i.e. 2DPCA, 2DFLD, R-ICA, MICA-1, MICA-2, TICA-1, and TICA-2.

Table 5. The recognition accuracies (%) of different schemes on the UMIST database. The values in parentheses are the corresponding number of features.

Training number	PCA	ICA	2DPCA	FLD	2DFLD	R-ICA	MICA-1	MICA-2	TICA-1	TICA-2
1	57.50 (18)	52.78 (17)	59.72 (672)	-----	-----	61.11 (672)	60.56 (336)	60.56 (336)	61.39 (30)	62.50 (24)
3	58.75 (16)	47.81 (18)	62.19 (336)	66.56 (8)	67.81 (224)	71.87 (224)	62.81 (336)	76.25 (224)	71.56 (27)	72.31 (10)
6	58.85 (46)	57.69 (50)	66.92 (336)	79.23 (10)	76.54 (448)	76.15 (224)	72.31 (224)	79.23 (224)	77.31 (12)	77.31 (30)
9	65.00 (48)	70.00 (25)	76.50 (336)	89.50 (9)	82.00 (224)	80.00 (336)	80.50 (112)	83.50 (224)	85.50 (18)	86.50 (20)

7. Conclusion

Independent feature extraction based on image matrix was discussed in this paper and two new measurements of the kurtosis for vector variables were presented. Then two Matrix-FastICA (MICA) algorithms were developed to estimate the demixing matrix directly from the images.

The MICA algorithms were consequently extended for high array image matrices so that the tensor ICA (TICA) schemes were proposed. Both TICA and MICA work directly on image matrix without image-to-vector stretching and hence they significantly alleviate the small sample size problem in subspace analysis. Extensive experiments on Yale, FERET, ORL and UMIST databases were conducted to validate the performance of the proposed schemes and it can be concludes that the matrix and tensor representation based MICA and TICA schemes perform better than the conventional vector representation based algorithms because the intrinsic structural information embedded in the images can be better preserved by MICA and TICA. Especially, the TICA schemes only require a small amount of features for classification, which means that the storage space can be greatly reduced while achieving high accuracy.

Reference

[1] Y. Koren and L. Carmel, "Robust linear discrimensionality", *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 4, pp. 459-470, July 2004.

[2] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40-51, January 2007.

[3] M. Kirby and L. Sirovich, "Application of the KL Procedure for the Characterization of Human Faces", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.

[4] M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 72-86, 1991.

[5] M. S. Bartlett, J. R. Movellan and T. J. Sejnowski, "Face recognition by independent component analysis", *IEEE Trans. on Neural Networks*, vol.13, no. 6, pp. 1450-1464, 2002.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July1997.

[7] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004.

[8] J. Ye, "Generalized low rank approximations of matrices," in *Proc. Int. Conf. Machine Learning (ICML)*, Banff, AB, Canada, 2004, 887-894.

[9] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.

[10] A. J. Bell and T. J. Senjowski, "The independent components of natural scenes are edge filters", *Vision Res.*, vol. 37, no. 23, pp. 3327-3338, 1997.

[11] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals", *Proc. IEEE*, vol. 69. pp. 529-541, 1981.

[12] A. Hyvärinen, "Survey on independent component analysis", *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[13] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 13, pp. 607–609, 1996.
[14] C. Liu and H. Wechsler, "Independent component analysis of Gabor feature for face recognition", *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 919-928, 2003.
[15] C. Liu, "Enhanced independent component analysis and its application to content based face image retrieval", *IEEE Trans. on System, man and Cybernetics, Part B*, vol. 34, no. 2, pp. 1117-1127, 2004.
[16] C. Y. Pong and J. H. Lai, "Face representation using independent component analysis", *Pattern Recognition*, vol,35, pp.1247-1257, 2002.
[17] M. Bressan and J. Vitrià, "On the selection and classification of independent features", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1312-1317, October 2003.
[18] I. Dagher and R. Nachar, "Face Recognition Using IPCA-ICA Algorithm", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 996-1000, June 2006.
[19] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA", Comput. Vist. Image Underst., vol. 91, no. 1/2, pp. 115-137, Jul. 2003.
[20] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 780–788, 2002.
[21] J. Yang, D. Zhang, and J. Y. Yang, "Constructing PCA Baseline algorithms to reevaluate ICA-based face recognition performance", *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 4, pp. 1015-1021, Aug. 2007.
[22] M. A. Vicente, P. O. Hoyer, and A. Hyvärinen, "Equivalence of some common linear feature extraction techniques for appearance-based object recognition tasks", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp.896-900, May 2007.
[23] S. Z. Li, X. Lu, X. Hou, X. Peng, and Q. Chen, "Learning Multiview Face Subspaces and Facial Pose Estimation Using Independent Component analysis", *IEEE Trans. Image Processing*, vol. 14, no. 6, June pp. 705-712, 2005.
[24] J. Kim, J. M. Choi, J. Yi, M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion", *IEEE Trans. Pattern Analysis and Machine Intelligence*. vol. 27, no. 12, pp. 1977-1981, 2005.
[25] D. Tao, X. Li, X. Wu, "General Tensor Discriminant Analysis and Gabor Features for Gait Recognition", *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 29, no. 10, pp. 1700-1715, Oct. 2007.
[26] T. Kim, S. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification", *In proc. Of IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
[27] M. Alex O. Vasilescu and D. Terzopoulos, "Multilinear Independent Component Analysis", *In Proc. Of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 547-553, June 2005.
[28] A. Hyvärinen and E. Oja, "Fast and Robust Fixed-Point Algorithms for independent component analysis", *IEEE Trans. Neural Networks*, vol. 10, no. 3, pp. 626-634, 1999.
[29] L. Lathauwer, B. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
[30] L. Lathauwer, "Signal Processing Based on Multilinear Algebra," PhD dissertation, Katholike Universiteit Leuven, 1997.
[31] K. Liu, Y. Cheng, and J. Yang, "Algebraic feature extraction for image recognition based on an optimal discriminant criterion," *Pattern Recognit.*, vol. 26, no. 6, pp. 903 – 911, Jun. 1993.
[32] Yale Univ. Face Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html, 2002.
[33] P. J. Phillips, H.Wechsler, J. Juang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vision Comput. J.*, vol. 16, no. 5, pp. 295–306, 1998.
[34] UMIST Face Database, Daniel Graham, http://images.ee.umist.ac.uk/danny/database.html, 2005.
[35] ORL Database, http://www.cam-orl.co.uk/facedatabase.html, 2002.
[36] G. S. Wang, M. X. Wu and Z. Jia, *Matrix Inequality*, Science Press, Beijing, China, 2006.