Feature Extraction for Classification Problems and Its Application to Face Recognition

Nojun Kwak [1]

Division of Electrical & Computer Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749 KOREA

Abstract

This study investigates a new method of feature extraction for classification problems. The method is based on the independent component analysis (ICA). However, unlike the original ICA, one of the unsupervised learning methods, it is developed for classification problems by utilizing class information. The proposed method is an extension of our previous work on binary-class problems to multi-class problems and it treats the class labels as input features in order to produce two sets of new features: one that carries much information on the class labels and the other that is irrelevant to the class. The learning rule for this method is obtained using the stochastic gradient method to maximize the likelihood of the observed data. Among the new features, using only class-relevant ones, the dimension of the feature space can be greatly reduced in line with the principle of parsimony, resulting better generalization. This method was applied to recognizing face identities and facial expressions using various databases such as the Yale, AT&T (former ORL), Color FERET face databases and so on. The performance of the proposed method was compared with those of conventional methods such as the principal component analysis (PCA), Fisher's linear discriminant (FLD), etc. The experimental results show that the proposed method performs well for face recognition problems.

Key words: ICA, classification, feature extraction, face recognition, facial expression

Email address: nojunk@ieee.org (Nojun Kwak).

[1] Nojun Kwak is an assistant professor of the Division of Electrical & Computer Engineering, Ajou University, Suwon, Korea.

1 Introduction

Many subspace methods have been successfully applied to construct features of an image [1] – [6]. Among these, the Eigenface [1] (based on PCA) and Fisherface [2] (based on FLD) methods are popular, because they allow the efficient characterization of a low-dimensional subspace whilst preserving the perceptual quality of a very high-dimensional raw image.

Though it is the most popular, the Eigenface method [1], by its nature, is not suitable for classification problems since it does not make use of any output class information in computing the principal components (PC). Besides, it extracts features that are not invariant under the transformation. Merely scaling the attributes changes resulting features. In addition, it does not use higher order statistics and it has been reported that the performance of the Eigenface method is severely affected by the level of illumination [2].

Unlike the Eigenface method, the Fisherface method [2] focuses on the classification problems to determine optimal linear discriminating functions for certain types of data whose classes have a Gaussian distribution and the centers of which are well separated. Although it is quite simple and powerful for classification problems, it cannot produce more than $N_c - 1$ features, where N_c is the number of classes. As in the Eigenface method, it only uses second order statistics in representing the images. On the other hand, some researchers have proposed subspace methods using higher order statistics such as the evolutionary pursuit and kernel methods for face recognition [5] [3] [4].

Recently, independent component analysis (ICA), which was originally devised for blind source separation problems, has received a great deal of attention in the neural networks and signal processing societies because of its potential applications in various areas. Bell and Sejnowski [7] developed an unsupervised learning algorithm for performing ICA based on entropy maximization in a single-layer feedforward neural network, and other researchers have shown that ICA is more powerful for face recognition than the PCA [8] [9] [6]. Unlike PCA and FLD, ICA uses higher order statistics and has been applied successfully in recognizing faces with changes in pose [8], and classifying facial actions [9]. Like PCA, it does not utilize the output class information and it leaves plenty of room for improvement.

In our previous works [10] [11] [12], we have proposed a feature extraction method called ICA-FX which utilizes the standard ICA algorithm for binary classification problems. In this method, the binary class label is treated as one of the hidden sources whose linear combinations are considered to constitute the observations. Then, feature extraction problems can be solved by standard ICA algorithms. By maximizing the joint mutual information between the class

labels and the new features, we could find a number of features that carry as much information on the class labels as possible.

However, the application of ICA-FX is limited to two-class problems and it cannot be applied to multi-class problems such as face recognition. As such, in this paper, ICA-FX is extended to multi-class problems. There have been several researches whose focus is to extend binary classification problems to multi-class problems [13] [14] [15] [16]. Most of the researches of this kind dealt with the problem of how to extend binary classifiers such as support vector machines (SVM) to multi-class classification problems and the most popular solution is to decompose multi-class classification problems into several multiple binary classification problems and to use combining schemes afterwards [13] [14] [15] [16]. Regarding feature extraction schemes, FLD which was originally designed for two-class problems has been easily extended to multi-class problems by changing the form of within-scatter and between-scatter matrices.

In this paper, instead of adding only one class node as an input to the structure of ICA, we have added N_c class nodes as inputs to the structure of ICA where N_c denotes the number of classes. In doing so, the 1-out-of-N_c coding scheme is used to code the class label.

The proposed method is applied to face recognition and facial expression problems. It greatly reduces the dimension of feature space while improving the classification performance.

This paper is organized as follows. A brief review of the ICA is carried out in Section 2 and a new feature extraction algorithm is proposed in Section 3. The experimental results for the face recognition problems are given to show the advantages of the proposed algorithm in Section 4. Finally, conclusions follow in Section 5.

2 Review of ICA

The problem of linear independent component analysis for blind source separation was developed in the literature [17] - [19]. In parallel, Bell and Sejnowski [7] developed an unsupervised learning algorithm based on entropy maximization of a feedforward neural network's output layer, which is referred to as the Infomax algorithm. The Infomax approach, the maximum likelihood estimation (MLE) approach, and the negentropy maximization approach were reported to have identical results [20] - [22].

The problem setting of an ICA is as follows: Assume that there is an L-dimensional zero-mean non-Gaussian source vector $\boldsymbol{s}(t) = [s_1(t), \cdots, s_L(t)]^T$,

such that the components $s_i(t)$'s are mutually independent, and an N-dimensional observed data vector $\boldsymbol{x}(t) = [x_1(t), \cdots, x_N(t)]^T$ is composed of a linear combination of sources $s_i(t)$ at each time point t, such that

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t) \tag{1}$$

where A is a full rank $N \times L$ matrix with $L \leq N$. The goal of the ICA is to find a linear mapping W, where each component of an estimate, $\boldsymbol{u}$, of the source vector

$$\boldsymbol{u}(t) = W\boldsymbol{x}(t) = WA\boldsymbol{s}(t) \tag{2}$$

is as independent as possible. The original sources, $\boldsymbol{s}(t)$, are recovered exactly when W is the inverse of A up to some scale changes and permutations. For a derivation of an ICA algorithm, it is usually assumed that $L = N$, because there is little knowledge of the number of sources. In addition, sources are assumed to be independent of time t and be drawn from an independent distribution $p_i(s_i)$.

Bell and Sejnowski [7] used a feed-forward neural processor to develop the Infomax algorithm, one of the most popular algorithms for ICA. The weight update rule of the Infomax algorithm is as follows:

$$\Delta W \propto [I - \boldsymbol{\varphi}(\boldsymbol{u})u^T]W \tag{3}$$

Here, $\boldsymbol{u} = W\boldsymbol{x}$ and $\boldsymbol{\varphi}(\boldsymbol{u}) = \left[-\frac{\frac{\partial p_1(u_1)}{\partial u_1}}{p_1(u_1)}, \cdots, -\frac{\frac{\partial p_N(u_N)}{\partial u_N}}{p_N(u_N)}\right]^T$.

The underlying assumption of the Infomax algorithm is that the sources have a super-Gaussian distribution, which has a sharp peak and longer tails than a normal Gaussian distribution. Some studies [23], [24] relaxed this assumption of the source distribution to be sub-Gaussian or super-Gaussian and [23] presented the extended Infomax learning rule:

$$\Delta W \propto [I - D\tanh(\boldsymbol{u})\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{u}^T]W \tag{4}$$
$$\begin{cases} d_i = 1 & : \text{super-Gaussian} \\ d_i = -1 & : \text{sub-Gaussian.} \end{cases}$$

Here d_i is the ith element of the N-dimensional diagonal matrix D, and it switches between sub- and super-Gaussian using the stability analysis.

In this paper, the extended Infomax algorithm (4) is adopted because it is easy to implement with less strict assumptions on the source distribution.

3 Feature extraction based on ICA for multi-class problems

ICA outputs a set of maximally independent vectors that are linear combinations of the observed data. Although these vectors might have some applications in such areas as blind source separation [7] and data visualization [25], for classification problems, it does not perform as good as supervised methods such as FLD, because it does not make use of class information. The effort to incorporate the standard ICA with supervised learning has been made in our previous works [11], [12]. In those studies, a new feature extraction algorithm, ICA-FX (feature extraction based on ICA), for classification problems with binary class labels was proposed. In this section, this will be extended to multi-class problems. The problem to be solved in this paper is as follows:

> **(Problem statement)** Assume that there are a normalized input feature vector, $\boldsymbol{x} = [x_1, \cdots, x_N]^T$, and an output class, $c \in \{o_1, \cdots, o_{N_c}\}$. The purpose of feature extraction is to extract $M(\leq N)$ new features $\boldsymbol{f_a} = [f_1, \cdots, f_M]^T$ from $\boldsymbol{x}$, by a linear combination of the x_i's, containing the maximum information on class c.

The main idea of the proposed feature extraction algorithm is simple. The application of the standard ICA algorithms to feature extraction for classification problems makes use of the class labels to produce two sets of new features; features that carry as much information on the class labels (these features will be useful for classification) as possible and the others that do not (these will be discarded). The advantage is that the general ICA algorithms can be used for feature extraction by maximizing the joint mutual information between the class labels and new features. The overall derivation of the algorithm takes almost the same steps as that for the binary case reported in [12].

Henceforth, a feature extraction method for classification problems by modifying the standard ICA algorithm is proposed. The main idea of the proposed method is to incorporate the class labels into the structure of standard ICA to extract a set of new features that contains much information about the class label.

In information theoretic view, the aim of feature extraction is to extract M new features $\boldsymbol{f}_a$ from the original N features, $\boldsymbol{x}$, such that $I(\boldsymbol{f}_a; c)$, the mutual information between newly extracted features $\boldsymbol{f}_a$ and the output class c, approaches $I(\boldsymbol{x}; c)$, the mutual information between the original features $\boldsymbol{x}$ and the output class c [12].

This can be satisfied if we can separate the input feature space $\boldsymbol{x}$ into two linear subspaces: one that is spanned by $\boldsymbol{f}_a = [f_1, \cdots, f_M]^T$, which contains the maximum information on the class label c, and the other spanned by

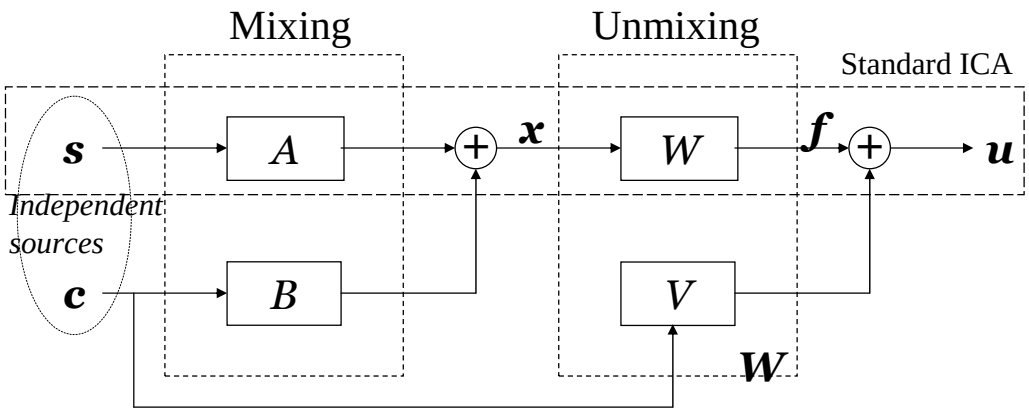


Fig. 1. Feature extraction in the structure of ICA (ICA-FX)

$\boldsymbol{f}_b = [f_{M+1}, \cdots, f_N]^T$, which is nothing to do with the class, i.e., independent of class c as much as possible.

The condition for this separation can be derived as follows. If it is assumed that $\boldsymbol{f} = W\boldsymbol{x}$ where $W \in \Re^{N\times N}$ is nonsingular, then $\boldsymbol{x}$ and $\boldsymbol{f} = [f_1, \cdots, f_N]^T = [\boldsymbol{f}_a^T, \boldsymbol{f}_b^T]^T$ span the same linear space, which can be represented with the direct sum of $\boldsymbol{f}_a \in \Re^M$ and $\boldsymbol{f}_b \in \Re^{N-M}$. Then by the data processing inequality [26],

$$\begin{aligned} I(\boldsymbol{x}; c) =& I(W\boldsymbol{x}; c) = I(\boldsymbol{f}; c) \\ =& I(\boldsymbol{f}_a, \boldsymbol{f}_b; c) \geq I(\boldsymbol{f}_a; c). \end{aligned} \tag{5}$$

The first equality holds because W is nonsingular. The second and the third equalities are from the definitions of $\boldsymbol{f}$, $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$. In the inequality on the last line, the left and the right-hand side are equal if $I(\boldsymbol{f}_b; c) = 0$.

Therefore, if we can make $\boldsymbol{f}_b$ be independent of c, the dimension of the input feature space can be reduced from N to $M(< N)$ by using only $\boldsymbol{f}_a$ instead of $\boldsymbol{x}$, without losing any information on the target class.

To make $\boldsymbol{f}_b$ be independent of the class information c, the structure of independent component analysis is modified as shown in Fig. 1 and this will be denoted by ICA-FX from now on.

In Fig. 1, in the structure of standard ICA drawn in the upper part of the figure, class information c is added as one of the input sources to generate the observation $\boldsymbol{x}$.

To begin with, let us first consider how discrete class labels can be incorporated in the structure of ICA. To enable this, the discrete class labels need to be encoded into numerical variables. Here, if the number of classes is N_c, the 1-out-of-N_c scheme is used in coding classes, i.e., a class vector, $\boldsymbol{c} = [c_1, \cdots, c_{N_c}]^T$, is introduced and if a class label, c, belongs to the lth value, i.e., $c = o_l$, then c_l is activated as 1 and all the other c_i's, $i \neq l$, are set to -1. After all the

training examples are presented, each $c_i, i = 1, \cdots, N_c$, is shifted in order to have zero mean and is scaled to have a unit variance.

Now ICA-FX modifies the structure of standard ICA as follows:

(Mixing) Assume that there are N independent sources $\boldsymbol{s} = [s_1, \cdots, s_N]^T$ which are also independent of the class label c. Assume also that the observed feature vector $\boldsymbol{x}$ is a linear combination of the sources $\boldsymbol{s}$ and $\boldsymbol{c}$ with the mixing matrix $A \in \Re^{N \times N}$ and $B \in \Re^{N \times N_c}$; i.e.,

$$\boldsymbol{x} = A\boldsymbol{s} + B\boldsymbol{c}. \tag{6}$$

In this scenario, for different class labels, the observed data $\boldsymbol{x}$ is assumed to have the same distribution with different means, because $\boldsymbol{c}$ is constant for a specific class label. In face recognition problems, one can think that the features of faces belonging to a person are distributed around a center in the feature space and that such centers are separated from each other by $B\boldsymbol{c}$.

(Unmixing) Because class information $\boldsymbol{c}$ is incorporated in the mixing stage, to reconstruct the sources $\boldsymbol{s}$, $\boldsymbol{c}$ should also be included in the unmixing stage. As shown in Fig. 1, the unmixing equation becomes

$$\boldsymbol{u} = W\boldsymbol{x} + V\boldsymbol{c}. \tag{7}$$

As in standard ICA, our sub-goal is to make $\boldsymbol{u}$ equal to $\boldsymbol{e}$, the scaled and permuted version of the source $\boldsymbol{s}$; i.e.,

$$\boldsymbol{e} \triangleq \Lambda\Pi\boldsymbol{s} \tag{8}$$

where Λ is a diagonal matrix corresponding to an appropriate scale and Π is a permutation matrix.

If this is the case, $\boldsymbol{u} = WA\boldsymbol{s} + (WB + V)\boldsymbol{c} = \Lambda\Pi\boldsymbol{s}$ should hold and $W = \Lambda\Pi A^{-1}$ and $V = -\Lambda\Pi A^{-1}B$. Then, the u_i's $(i = 1, \cdots, N)$ are independent of the class label c by the assumption.

Because our final goal is to make some of the features $\boldsymbol{f}_b$ independent of class information $\boldsymbol{c}$ and force the others $\boldsymbol{f}_a = \boldsymbol{f} \setminus \boldsymbol{f}_b$ contain all the information about the class which was contained in the original features $\boldsymbol{x}$, some restrictions in the form of V is made as shown in Fig. 2 and equation (9).

Here, the original feature vector $\boldsymbol{x}$ is fully connected to $\boldsymbol{u} = [u_1, \cdots, u_N]$ and the class vector $\boldsymbol{c}$ is connected only to $\boldsymbol{u}_a = [u_1, \cdots, u_M]$, but there is no connection from $\boldsymbol{c}$ to $\boldsymbol{u}_b = [u_{M+1}, \cdots, u_N]$. This makes the last $N - M$ rows of V becomes zero vectors.

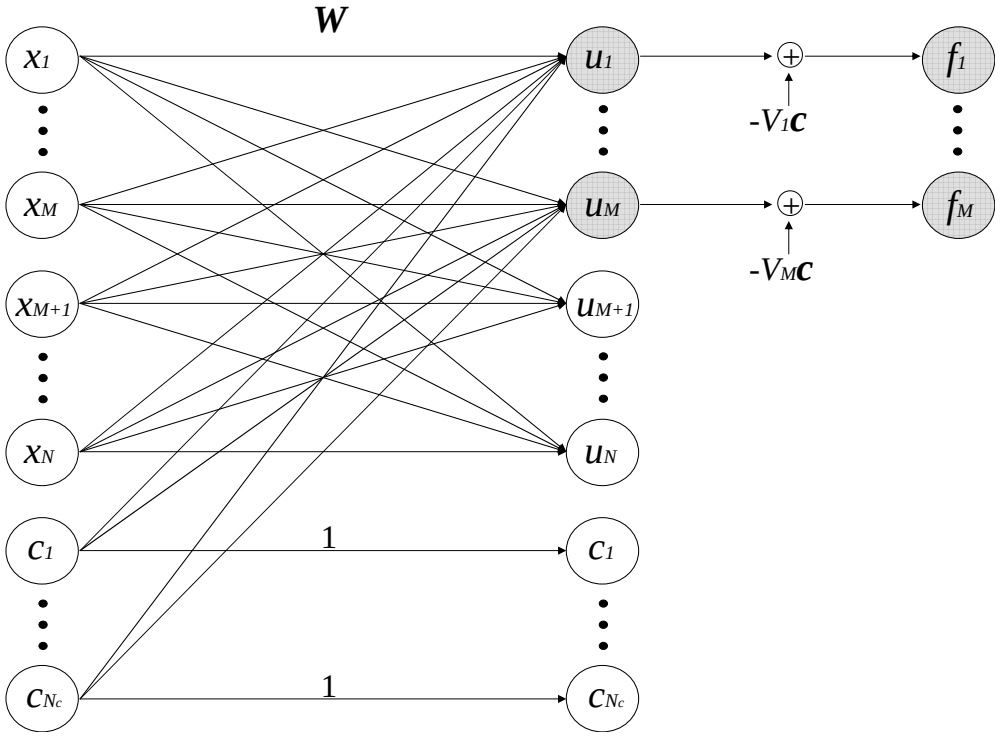


Fig. 2. Feature extraction algorithm based on ICA (ICA-FX)

In the figure, the augmented weight matrix $\boldsymbol{W} \in \Re^{(N+N_c)\times(N+N_c)}$ becomes

$$\boldsymbol{W} = \left(\begin{array}{c|c} W & V \\ \hline \mathbf{0}_{N_c,N} & I_{N_c} \end{array}\right) = \left(\begin{array}{ccc|ccc} w_{1,1} & \cdots & w_{1,N} & v_{1,1} & \cdots & v_{1,N_c} \\ \vdots & & \vdots & \vdots & & \vdots \\ w_{M,1} & \cdots & w_{M,N} & v_{M,1} & \cdots & v_{M,N_c} \\ w_{M+1,1} & \cdots & w_{M+1,N} & & & \\ \vdots & & \vdots & & \mathbf{0}_{N-M,N_c} & \\ w_{N,1} & \cdots & w_{N,N} & & & \\ \hline & \mathbf{0}_{N_c,N} & & & I_{N_c} & \end{array}\right). \quad (9)$$

where $W \in \Re^{N\times N}$ and $V = [V_a^T, \mathbf{0}_{N-M,N_c}^T]^T \in \Re^{N\times N_c}$. Here the first nonzero M rows of V is denoted as $V_a \in \Re^{M\times N_c}$.

In this structure, among the elements of $\boldsymbol{f} = W\boldsymbol{x}(= \boldsymbol{u} - V\boldsymbol{c})$, $\boldsymbol{f}_b = [f_{M+1}, \cdots, f_N]^T$ will be equal to $\boldsymbol{u}_b$ because the ith row of V, V_i is zero for $i = M+1, \cdots, N$. Thus it will be independent of c by the assumption. Therefore, if the relation $\boldsymbol{u} = \boldsymbol{e}$ and the independence assumption on the sources and the class label hold, the $M(< N)$ dimensional new feature vector $\boldsymbol{f}_a$ will contain the whole information about the class label c that was contained in $\boldsymbol{x}$.

Now that the feature extraction problem is set in a similar form as the standard ICA problem, a learning rule for $\boldsymbol{W}$ can be derived in the same way as that for the standard ICA. Below, the MLE approach was used for the derivation.

If it is assumed that $\boldsymbol{u} = [u_1, \cdots, u_N]^T$ is made equal to $\boldsymbol{e}$, a scaled and permutated version of the source, $\boldsymbol{s}$, as in (8), the log likelihood of the data for a given $\boldsymbol{W}$ becomes the following:

$$L(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{W}) = \log|\det \boldsymbol{W}| + \sum_{i=1}^{N} \log p_i(u_i) + \log p(\boldsymbol{c}) \tag{10}$$

because

$$p(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{W}) = |\det \boldsymbol{W}|\ p(\boldsymbol{u}, \boldsymbol{c}) = |\det \boldsymbol{W}|\ \prod_{i=1}^{N} p_i(u_i)\ p(\boldsymbol{c}). \tag{11}$$

The second equality of (11) is from the assumption that each element of $\boldsymbol{s}$ is independent of the other elements of $\boldsymbol{s}$, which is also independent of the class vector $\boldsymbol{c}$.

Now, L can be maximized, and this can be achieved by the steepest ascent method. Because the last term in (10) is a constant, differentiating (10) with respect to $\boldsymbol{W}$ leads to

$$\begin{aligned} \frac{\partial L}{\partial w_{i,j}} &= \frac{adj(w_{j,i})}{|\det \boldsymbol{W}|} - \varphi_i(u_i)x_j \qquad 1 \le i, j \le N \\ \frac{\partial L}{\partial w_{i,N+j}} &= -\varphi_i(u_i)c_j \qquad 1 \le i \le M, 1 \le j \le N_c \end{aligned} \tag{12}$$

where $adj(\cdot)$ is adjoint and $\varphi_i(u_i) = -\frac{dp_i(u_i)}{du_i}/p_i(u_i)$. Note that each c_i has a binary numerical value depending on the class label c.

It can be seen that $|\det \boldsymbol{W}| = |\det W|$ and $adj(w_{j,i})/|\det \boldsymbol{W}| = W_{i,j}^{-T}$. Thus the learning rule becomes

$$\begin{aligned} \Delta W &\propto\ W^{-T} - \boldsymbol{\varphi}(\boldsymbol{u})\boldsymbol{x}^T \\ \Delta V_a &\propto\ -\boldsymbol{\varphi}(\boldsymbol{u}_a)\boldsymbol{c}^T. \end{aligned} \tag{13}$$

Here $\boldsymbol{\varphi}(\boldsymbol{u}) \triangleq [\varphi_1(u_1), \cdots, \varphi_N(u_N)]^T$ and $\boldsymbol{\varphi}(\boldsymbol{u}_a) \triangleq [\varphi_1(u_1), \cdots, \varphi_M(u_M)]^T$.

Since the two terms in (13) have different tasks regarding the update of the separate matrices W and V_a, the learning process can be divided. Applying a natural gradient on updating W, by multiplying W^TW on the right side of the first equation of (13), the following is obtained.

$$\begin{aligned} W^{(t+1)} &= W^{(t)} + \mu_1[I_N - \boldsymbol{\varphi}(\boldsymbol{u})\boldsymbol{f}^T]W^{(t)} \\ V_a^{(t+1)} &= V_a^{(t)} - \mu_2\boldsymbol{\varphi}(\boldsymbol{u}_a)\boldsymbol{c}^T. \end{aligned} \tag{14}$$

Here μ_1 and μ_2 are the learning rates that can be set differently.

By this weight update rule, the resulting u_i's will be not only independent of one another but also independent of the class label c, if the initial point (W^0, V_a^0) is sufficiently near the optimal point.

Note that the learning rule for W is the same as the original ICA learning rule [7], and also note that $\boldsymbol{f}_a$ corresponds to the first M elements of $W\boldsymbol{x}$. Therefore, the optimal features $\boldsymbol{f}_a$ can be extracted by the proposed algorithm when it finds the optimal solution for W by (14). Note also that although the 1-of-N_c scheme was used in coding the discrete class labels into numerical ones, the derivation of the algorithm can be easily applied to other coding schemes.

The stability condition of the learning rule (14) for multi-class ICA-FX can be easily derived in the same way as that for binary classification problems in [12] as follows:

Theorem: The local asymptotic stability of the ICA-FX around the stationary point $(W = \Lambda\Pi A^{-1}, V = -\Lambda\Pi A^{-1}B)$ is governed by the nonlinear moment

$$\kappa_i = E\{\dot{\varphi}_i(e_i)\}E\{e_i^2\} - E\{\varphi_i(e_i)e_i\} \tag{15}$$

and it is stable if

$$1 + \kappa_i > 0, \quad 1 + \kappa_j > 0, \quad (1 + \kappa_i)(1 + \kappa_j) > 1 \tag{16}$$

for all $1 \leq i, j \leq N$. Therefore, the sufficient condition is

$$\kappa_i > 0, \quad 1 \leq i \leq N. \tag{17}$$

Proof: See Appendix.

Note that the condition for the stability of the ICA-FX in Theorem 1 is identical to that of the standard ICA in [27]. Therefore, the interpretation of the nonlinear moment κ_i can be consulted to [27]. It is important that the local stability be preserved when the activation function $\varphi_i(e_i)$ is chosen to be positively correlated with the true activation function $\varphi_i^*(e_i) \triangleq -\dot{p}_i(e_i)/p_i(e_i)$.

The computational complexity of the algorithm increases approximately in the order of $(N^2 + MN_c)$, which is the number of elements in $\boldsymbol{W}$.

As in standard ICA, applying PCA before conducting the ICA-FX can enhance the performance of the ICA-FX much more. Therefore, the PCA was used in all the following experimental results before applying the ICA-FX.

4 Experimental Results

In this section, the ICA-FX was applied to face recognition problems and the performance was compared with those of the other methods such as PCA, ICA, and FLD. This is an extension of [28] where face recognition problems were viewed as multiple binary classification problems and the binary version of the ICA-FX [11] [12] was used to tackle the multi-class classification problems.

To apply the ICA-FX to face recognition problems, we first need to determine the original input features $\boldsymbol{x}$ of an image which will be used to obtain the new features $\boldsymbol{f}_a$. There are several methods for determining the features of an image, such as wavelets, Fourier analysis, fractal dimensions, and many other methods [29]. Among them, one can easily come up with an idea of using each pixel as one feature.

Though this is the simplest, directly applicable to PCA and FLD, without loss of information of an image, the dimension of the input space of this method becomes too large to be handled easily. Moreover, images are vulnerable to noise. Thus, in this paper, each image was downsampled into a manageable size in order to reduce the computational complexity as in [4].

Subsequently, each downsampled pixel was transformed to have a zero mean and a unit variance over the training images, and PCA was then performed both as a whitening process of the ICA-FX and for the purpose of further reducing the dimension of the input space which leads to a significant reduction of time for training in ICA-FX.

Therefore, in this setting, x_i corresponds to the coefficient of the ith principal component of a given image. Finally, the main routine of the ICA-FX was applied to extract the valuable features for classification. Figure 3 shows the experimental procedure used in this paper. For comparison, ICA and FLD were also used after PCA was performed, as shown in the figure. The downsampling step was identical to that used in [4] and we could compare the performance of the proposed method with those of kernel methods reported in [4].

If not explicitly stated otherwise, the performances were tested with the leave-one-out scheme and the classification was performed using the one nearest neighborhood classifier. That is to say, to test the ith image among the total n images, all the other $(n-1)$ images were used for training and the ith image was classified as the identity of the image whose Euclidean distance from the ith image was the closest among the $(n-1)$ images. For the Color FERET dataset and the CMU-PIE dataset, training and test data were set differently and the classification was performed using the one nearest neighborhood classifier.

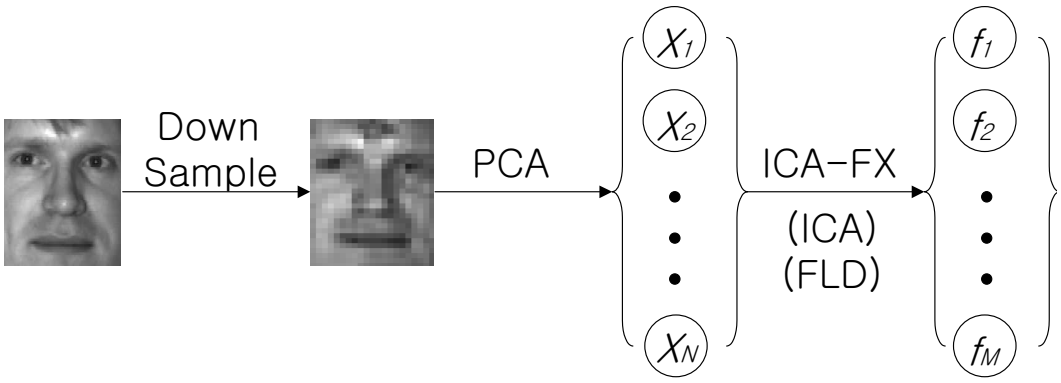


Fig. 3. Experimental procedure

Fig. 4. Yale Database

The ICA-FX was applied to the Yale [2], the AT&T [30], the Color FERET [31] [32] , and the CMU-PIE [33] face databases for face recognition, and to the Japanese Female Facial Expression (JAFFE) [34] database for classifying facial expressions. Throughout the experiments, the learning rates μ_1 and μ_2 for the ICA-FX were set to 0.002 and 0.1 respectively and the number of iterations for learning was set to 300. The results of ICA were obtained by the extended Infomax algorithm with a learning rate of 0.002 and 300 iterations.

4.1 Yale Database

The Yale face database consists of 165 grayscale images of 15 individuals. There are 11 images per subject with different facial expressions or configurations. In [2], the authors report two types of databases: a closely cropped set and a full face set. In this paper, the closely cropped set was used and the images were downsampled into 21 $\times$ 30 pixels as in [4]. Figure 4 shows the downsampled images of the first three individuals of the dataset.

For the data, PCA was first performed on 630 downsampled pixels and various

Fig. 5. Weights of various subspace methods for Yale dataset. (1st row: PCA (Eigenfaces), 2nd row: ICA, 3rd row: FLD (Fisherfaces), 4th row: ICA-FX)

numbers of principal components were used as the inputs of the ICA, FLD and ICA-FX. Figure 5 represents the typical weights of PCA, ICA, FLD, and ICA-FX. The top row is the first 10 principal components (PC) among 165 PC's, which are generally referred to as Eigenfaces. The third row is the first 10 out of 14 Fisherfaces that are the weights of FLD. The second and the fourth rows are the weights of ICA and ICA-FX respectively. Here, the first 30 principal components were used as inputs to ICA, FLD, and ICA-FX and ten features were extracted using the ICA-FX.

Figure 6 shows the performances of PCA, ICA, FLD, and ICA-FX when different numbers of principal components were used in the face classification. The performances were obtained by the leave-one-out scheme. Note that the number of features produced by the FLD is 14, because there are 15 subjects in this dataset, while the number of features by ICA is the same as that of PCA. In the ICA-FX, the number of features was set to 10. Because ICA and the ICA-FX can have different results according to the initial weight randomization, the results of ICA and ICA-FX are averages of two experiments. From the figure, it can be seen that the performance of ICA-FX is better than those of the other methods regardless of the number of principal components that are used as inputs to the ICA-FX.

Note that the error rate decreases as the number of principal components increases in the ICA-FX. In other methods, the error rates decrease in the beginning as the number of features increases but they increase as the number of features further increases. Regarding PCA and ICA, the reason for this can be attributed to '*Occam's razor*' or '*the law of parsimony*' which states that

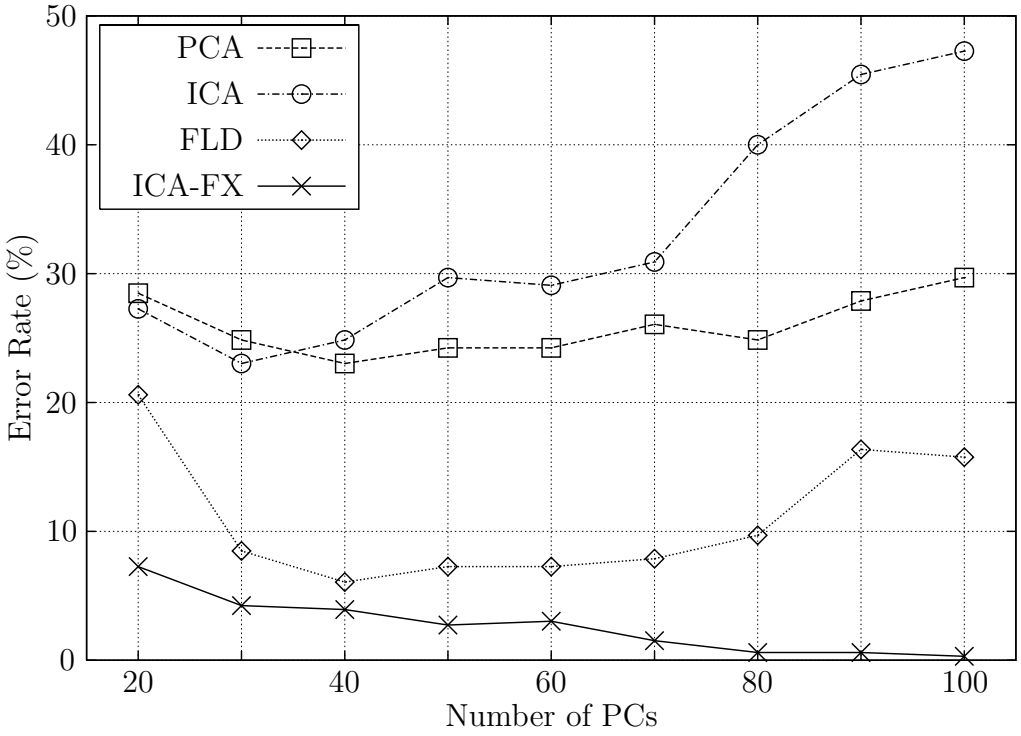


Fig. 6. Comparison of performances of PCA, ICA, FLD, and ICA-FX on Yale database with various number of PC's. (The numbers of features for FLD and ICA-FX are 14 and 10 respectively. The number of features for ICA is the same as that of PCA)

simple decision rule has better generalization performances [35]. On the other hand, because the number of features used in FLD is fixed to 14 regardless of the number of principal components, the reason for this phenomenon on FLD can be sought for differently. As the dimension of input space increases the within-scatter matrix in FLD becomes close to a singular matrix and FLD may suffer from the '*small sample size*' problem which results in a poor generalization performance [36].

Figure 7 shows the performance of the ICA-FX with various numbers of extracted features (M in Section III) when the number of principal components (N in Section III) was fixed to 30, 40, and 50. In the figure, it can be seen that the performances are better when $10 \sim 20$ features are extracted and the error rates tend to grow as the number of extracted features increases for all the three cases. This phenomenon can again be explained by '*Occam's razor*' [35]. The unnecessarily large number of features degrades the generalization performance.

To provide insights on how the ICA-FX simplifies the face pattern distribution, each face pattern is projected into the two dimensional feature space in Figure 8. This figure provides a low-dimensional representation of the data, which can be used to capture the structure of the data. In the figure, the PCA,

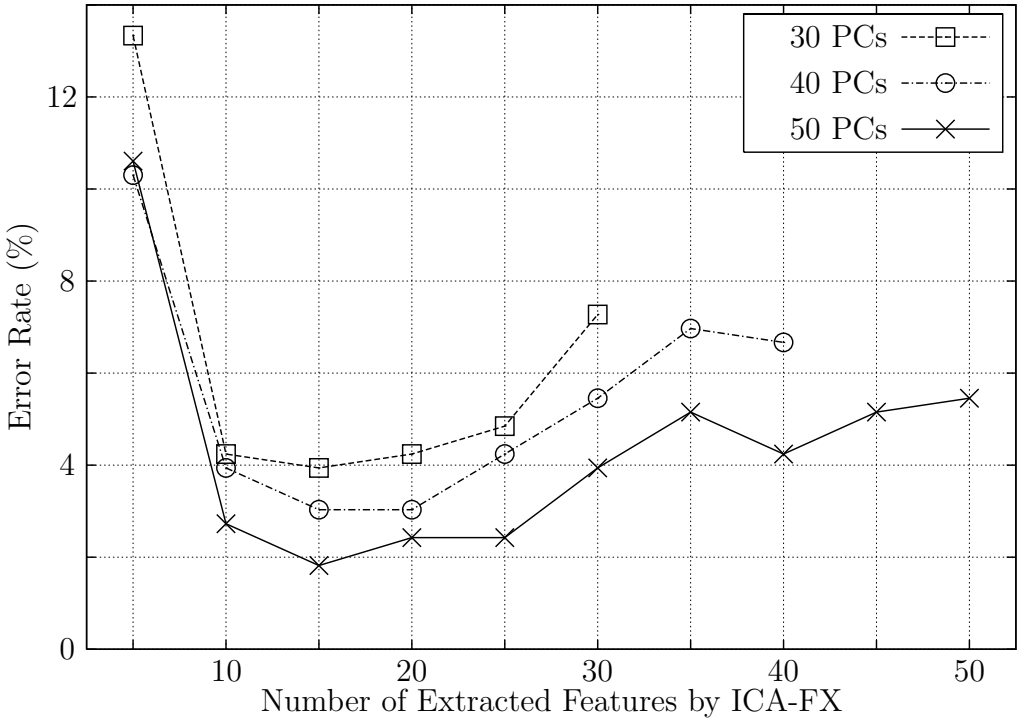


Fig. 7. Performances of ICA-FX on Yale database with various number of features used. (30, 40, and 50 principal components were used as inputs to ICA-FX.)

ICA, FLD, and ICA-FX were used to generate features using all the 165 face images. Thirty principal components are used as inputs of ICA, FLD, and ICA-FX. For the ICA-FX, ten features are extracted. The most significant two features are selected as bases for PCA and FLD cases, and the first two features are selected as bases for ICA and ICA-FX. For the sake of simplicity in visualization, the first seven identities among the total fifteen identities are shown in the figure, i.e., total 77 images are used for the plots. Before plotting, features are normalized to have zero means and unit variances. Seven different symbols such as '+', and '*' are used to represent different identities. Note that the same symbols cluster more closely in the cases of FLD and ICA-FX than those of PCA and ICA as expected.

In Table 1, the performance of ICA-FX was compared with those of the other algorithms: PCA (Eigenface), FLD (Fisherface), ICA, and the kernel methods presented in [4]. In the table, the hold-out test was also used as well as the leave-one-out test. For hold-out test, the first 8 images of each person is used for training and the other 3 images were reserved for test data. Therefore, the training and the test data consist of 120 and 45 samples respectively in hold-out test. The experimental procedure for this work was set to be the same as that in [4]. In the experiments, PCA was initially conducted on 630 pixels and the first 30 principal components were used as in [2]. Subsequently, the FLD, ICA, and ICA-FX were applied to these 30 principal components. Table

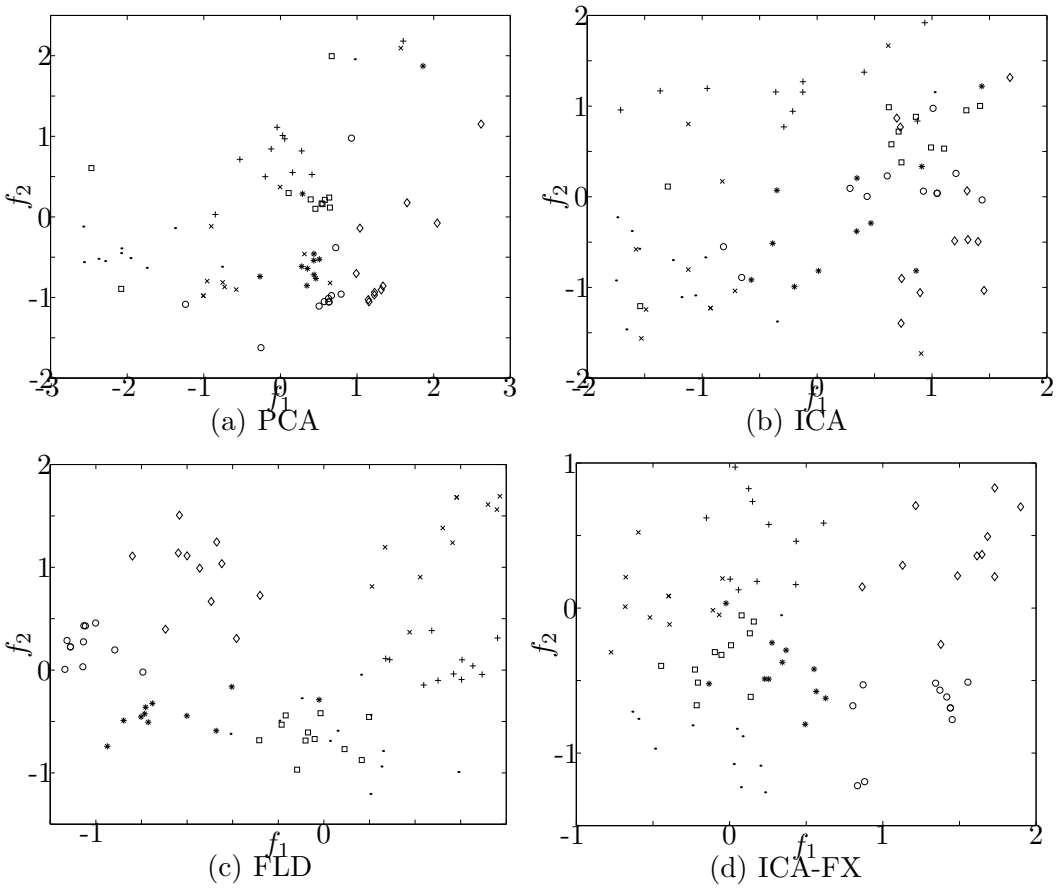


Fig. 8. Distribution of seven identities $(\cdot, \circ, *, \times, +, \diamond, \square)$ of Yale data in two dimensional subspaces of PCA, ICA, FLD, and ICA-FX.

Table 1
Experimental results on Yale database: the leave-one-out and the hold-out tests were both reported. The training and the test data consist of 120 and 45 samples respectively in hold-out test. The results for kernel methods are from [4].

Method	Dim. of Reduced Space	Leave-one-out (165)		Hold-out (120/45)	
		No. of Error	Error Rate (%)	No. of Error	Error Rate (%)
Eigenface (PCA)	30	41	24.85	10	22.22
ICA	30	38	23.03	10	22.22
Fisherface (FLD)	14	14	8.48	5	11.11
Kernel Eigenface (d=3)	60	40	24.24	–	–
Kernel Fisherface (G)	14	10	6.06	–	–
ICA-FX	**10**	**7**	**4.24**	**3**	**6.66**

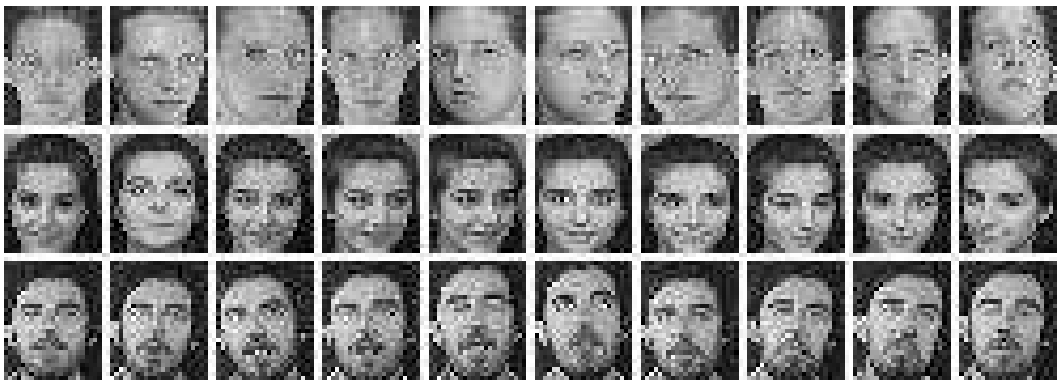


Fig. 9. AT&T Database

1 shows the classification error rates of each methods. In the left columns, the error rates were determined by the 'leave-one-out' strategy, while in the right columns, they were obtained by the 'hold-out' scheme. The recognition was performed using the one nearest neighbor classifier as in [2]. In the leave-one-out case, the performances of the kernel Eigenface, and the kernel Fisherface are from [4]. From the table it can be seen that the ICA-FX outperforms the other methods using a smaller number of features. Also note that the performance of 'leave-one-out' scheme is better than 'hold-out' scheme for supervised feature extractors, FLD and ICA-FX. This may due to the smaller number of training examples in hold-out test compared to those in leave-one-out test.

4.2 AT&T Database

The AT&T database of faces (formerly 'The ORL Database of Faces') [30], consists of 400 images, which are ten different images for 40 distinct individuals. It includes various lighting conditions, facial expressions, and facial details. The images were downsampled into 23×28 pixels for computational efficiency as in [4]. Figure 9 shows the downsampled images of the first three individuals.

The experiments were performed exactly the same way as in the Yale database. The results in the forthcoming figures are from the leave-one-out test with the one nearest neighborhood classifier. Averages of two experiments for the ICA and ICA-FX are reported here. Figure 10 shows the weights of the PCA, ICA, FLD, and ICA-FX for this dataset respectively.

Figure 11 shows the error rates of the PCA, ICA, FLD, and ICA-FX when different numbers of principal components were used. Note that the number of extracted features by FLD is 39, because there are 40 classes. Because there must be at least 40 PC's to get 39 Fisherfaces, the error rates of the FLD for

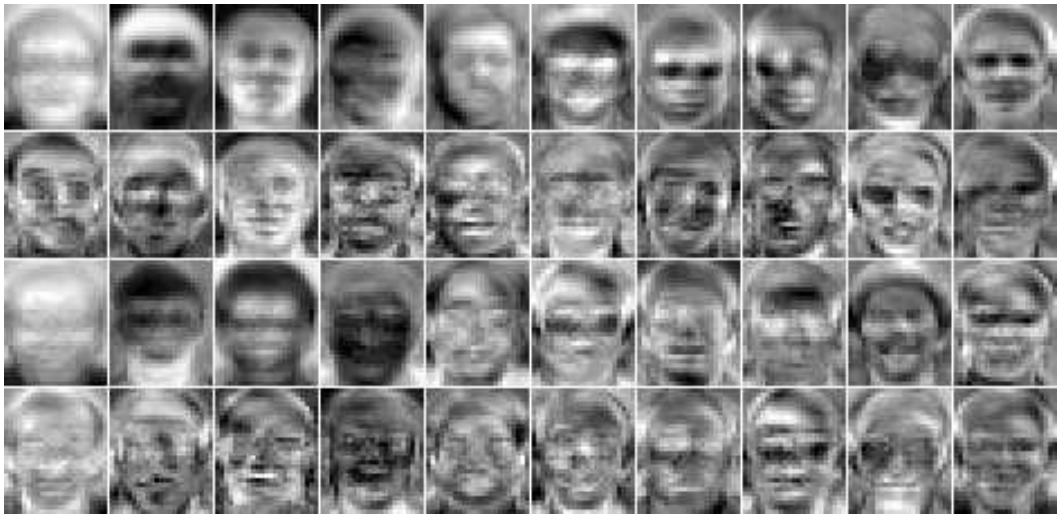


Fig. 10. Weights of various subspace methods for AT&T dataset. (1st row: PCA (Eigenfaces), 2nd row: ICA, 3rd row: FLD (Fisherfaces), 4th row: ICA-FX)

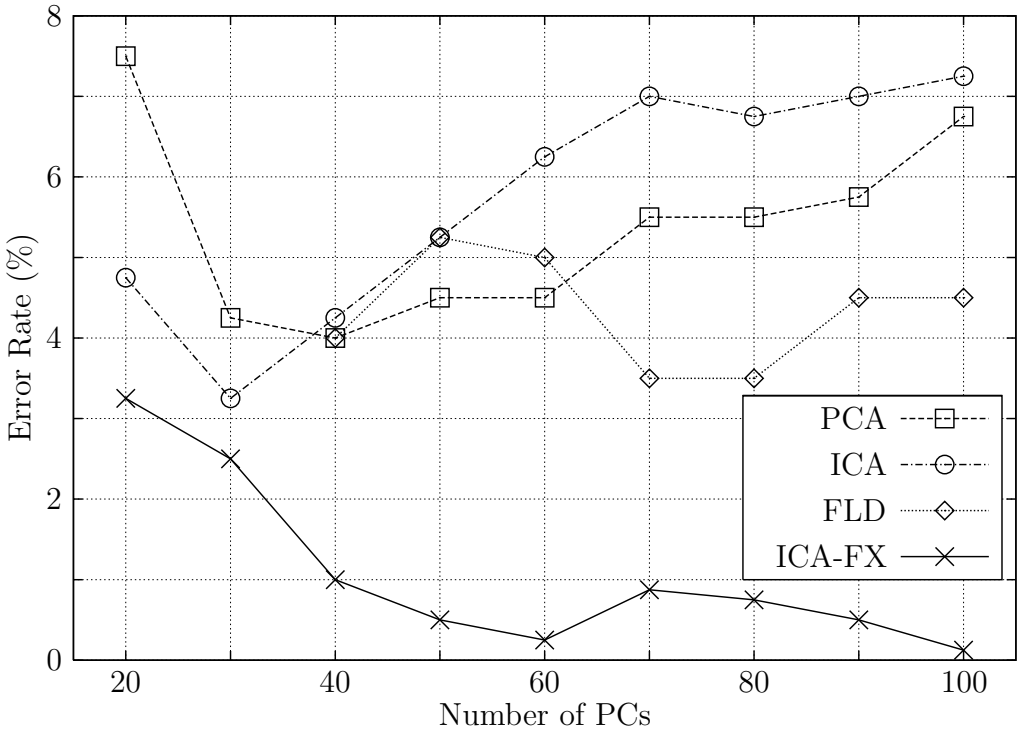


Fig. 11. Comparison of performances of PCA, ICA, FLD, and ICA-FX on AT&T database with various number of PC's. (The numbers of features for FLD and ICA-FX are 39 and 10 respectively. The number of features for ICA is the same as that of PCA)

20 and 30 PC's are not reported. The number of extracted features for the ICA-FX was set to ten.

Figure 12 shows the error rates of the ICA-FX when different numbers of

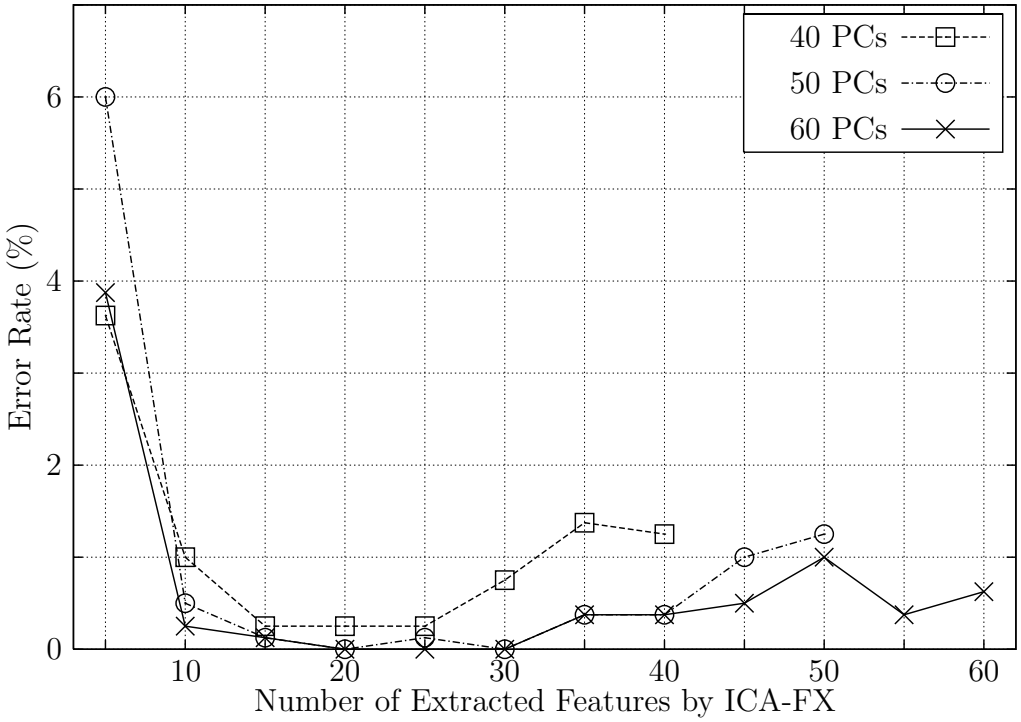


Fig. 12. Performances of ICA-FX on AT&T database with various number of features used. (40, 50, and 60 principal components were used as inputs to ICA-FX.)

features were used with 40, 50, or 60 principal components. It can be seen that there are little differences in the performance when ten or more features are extracted and the error rates gradually increase as the number of features increases. This phenomenon is the same as that for the Yale database.

In Figure 13, each face pattern is projected into the two dimensional feature space. In the figure, the PCA, ICA, FLD, and ICA-FX were used to generate features using all the 400 face images. Forty principal components are used as inputs of ICA, FLD, and ICA-FX. For the ICA-FX, ten features are extracted. The most significant two features are selected as bases for PCA and FLD cases, and the first two features are selected as bases for ICA and ICA-FX. As in Yale databases, for the sake of simplicity in visualization, the first seven identities among the total 40 identities are shown in the figure, i.e., total 70 images are used for the plots. Before plotting, features are normalized to have zero means and unit variances. Seven different symbols are used to represent different identities. In the figure, one can rather easily separate one cluster of a symbol from another in the cases of FLD and ICA-FX than in the cases of PCA and ICA.

Table 2 shows the error rates of the PCA, ICA, FLD, the kernel methods, and ICA-FX. Both the leave-one-out and the hold-out tests were shown. For hold-out test, the first 5 images of each person were used for training and the rest 5 images were used for test. Therefore, the numbers of training and test

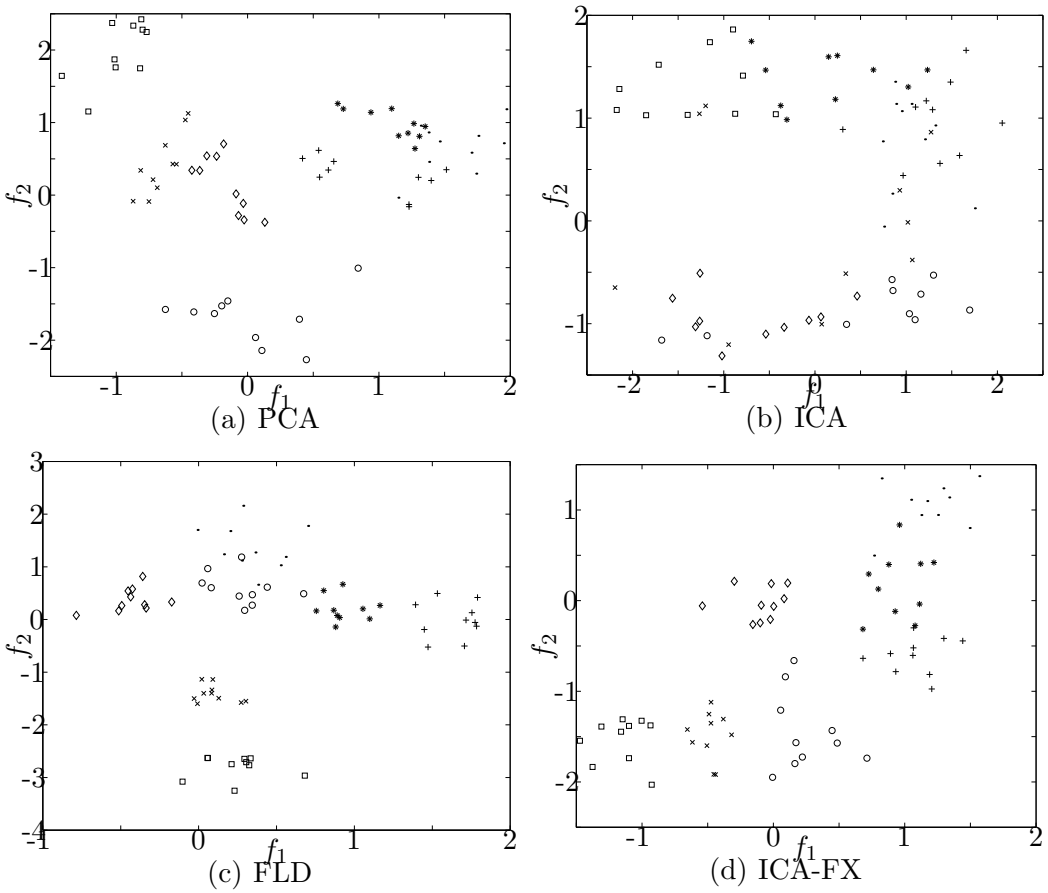


Fig. 13. Distribution of seven identities $(\cdot, \circ, *, \times, +, \diamond, \square)$ of AT&T data in two dimensional subspaces of PCA, ICA, FLD, and ICA-FX.

Table 2
Experimental results on AT&T database: the leave-one-out and the hold-out tests were both reported. The training and the test data consist of 200 and 200 samples respectively in hold-out test. The results for kernel methods are from [4].

Method	Dim. of Reduced Space	Leave-one-out (400)		Hold-out (200/200)	
		No. of Error	Error Rate (%)	No. of Error	Error Rate (%)
Eigenface (PCA)	40	16	4.00	22	11.0
ICA	40	17	4.25	23	11.5
Fisherface (FLD)	39	16	4.00	19	9.5
Kernel Eigenface (d=3)	40	8	2.00	–	–
Kernel Fisherface (G)	39	5	1.25	–	–
ICA-FX	**10**	**4**	**1.00**	**13**	**6.5**

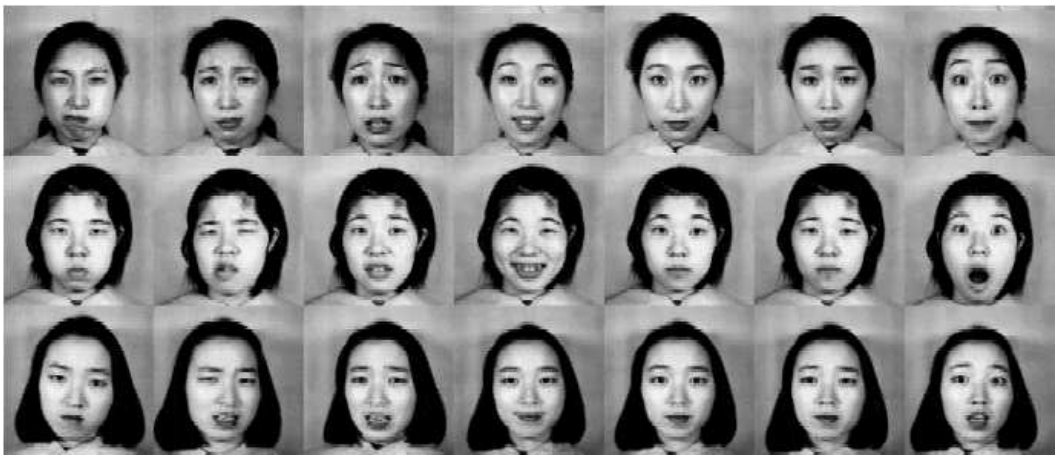


Fig. 14. JAFFE Database

Table 3
Distribution of JAFFE database

Category	No. of Images	Total Images
Angry	30	
Disappointed	29	
Fearful	32	
Happy	31	213
Sad	31	
Surprised	30	
Neutral	30	

samples are 200 each. For ICA, FLD and ICA-FX, 40 principal components were used for the input vector as in [4]. The performances of the kernel methods are those from [4]. As shown in the table, it can be seen that the ICA-FX outperforms the other methods with significantly less features. Also note that the performances of leave-one-out test are better than hold-out test. This may due to the smaller number of training examples in the hold-out test.

4.3 JAFFE Database

This database consists of 213 images of seven facial expressions (angry, disappointed, fearful, happy, sad, surprised, and neutral) posed by ten Japanese female models [34]. The number of images belonging to each category is shown in Table 3. Figure 14 shows samples of the images. For the experiments, each image was downsampled to 16×16 and a total of 256 pixels were used. The PCA, ICA, FLD, and ICA-FX were used for recognizing seven facial expressions and the weights for each method are shown in Figure 15. Note that there

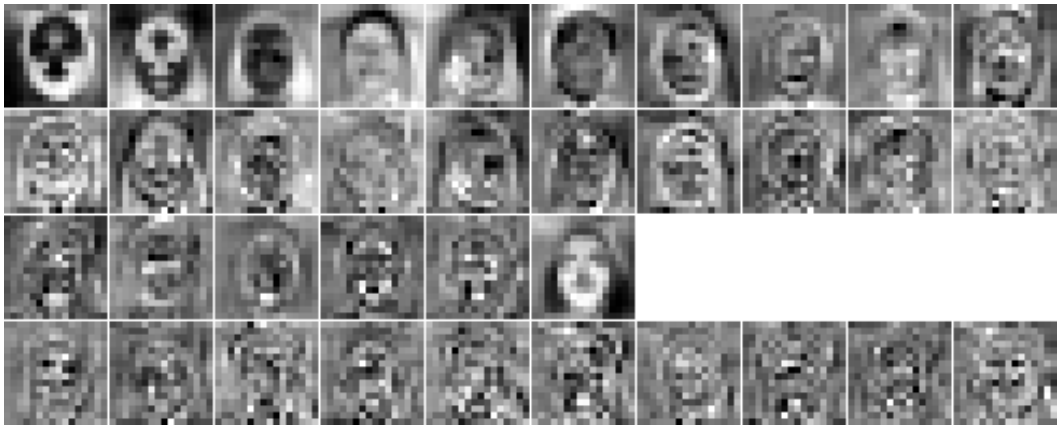


Fig. 15. Weights of various subspace methods for JAFFE dataset. (1st row: PCA (Eigenfaces), 2nd row: ICA, 3rd row: FLD (Fisherfaces), 4th row: ICA-FX)

are six Fisherfaces, because there are seven categories of facial expression.

The performances of the various methods are shown in Figure 16. In the figure, various numbers of principal components were used as the inputs to ICA, FLD and ICA-FX. For the ICA-FX, ten features were extracted. It can be seen that the performances of the FLD is even worse than those of the PCA especially when the number of principal components were small. The performance of the ICA-FX is better than those of the other methods in most cases. The error rates for ICA-FX decreases consistently as the number of principal components increases, while those of the others do not. This phenomenon is the same as in Yale and AT&T datasets and can be explained by '*Occam's razor*' and '*small sample size*' problem as before.

Figure 17 shows the error rates of the ICA-FX when different numbers of features were used with 50, 60, or 70 principal components. It is expected from Figure 16 and 17, that error rates can be reduced to below 5% with more principal components and 15 features extracted by the ICA-FX.

Table 4 shows the performances of the PCA, ICA, FLD, and ICA-FX when the first 60 principal components were used. In this table, the number of features by the ICA-FX was set to ten. The experimental results show that the classification rates of the ICA-FX are better than those of the other methods. Furthermore, the performance of the FLD is unsatisfactory in this case. The reason is that the number of extracted features by the FLD is too small to contain sufficient information on the class. When five features are extracted by the ICA-FX, the classification errors are approximately $19 \sim 23\%$ in Fig 17, and are close to that of the FLD in Table 4.

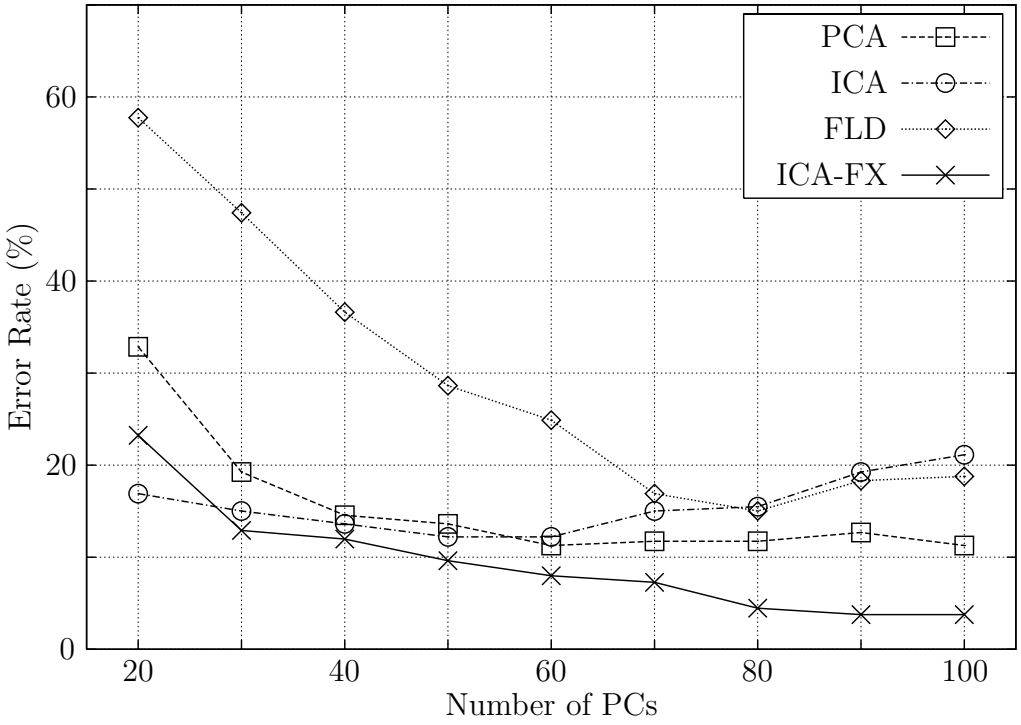


Fig. 16. Comparison of performances of PCA, ICA, FLD, and ICA-FX on JAFFE database with various number of PC's. (The numbers of features for FLD and ICA-FX are 6 and 10 respectively. The number of features for ICA is the same as that of PCA)

Table 4
Experimental results on JAFFE database

Method	Dim. of Reduced Space	No. of Error	Error Rate (%)
Eigenface (PCA)	60	24	11.27
ICA	60	26	12.20
Fisherface (FLD)	6	53	24.88
ICA-FX	**10**	**17**	**7.98**

4.4 FERET Database

The Color FERET database contains a total of 11,338 facial images obtained from 994 subjects (individuals). The experimental settings were identical to that of [37]. Total 992 subjects that have both 'fa' and 'fb' frontal images were selected and these 1,984 images were used in the experiments. The size of each image is 768 × 512 (pixels). The first 200 subjects having true color images (the subject '00043' to the subject '00245') were used for training, while the remaining 792 subjects were used for testing. Total 400 images were

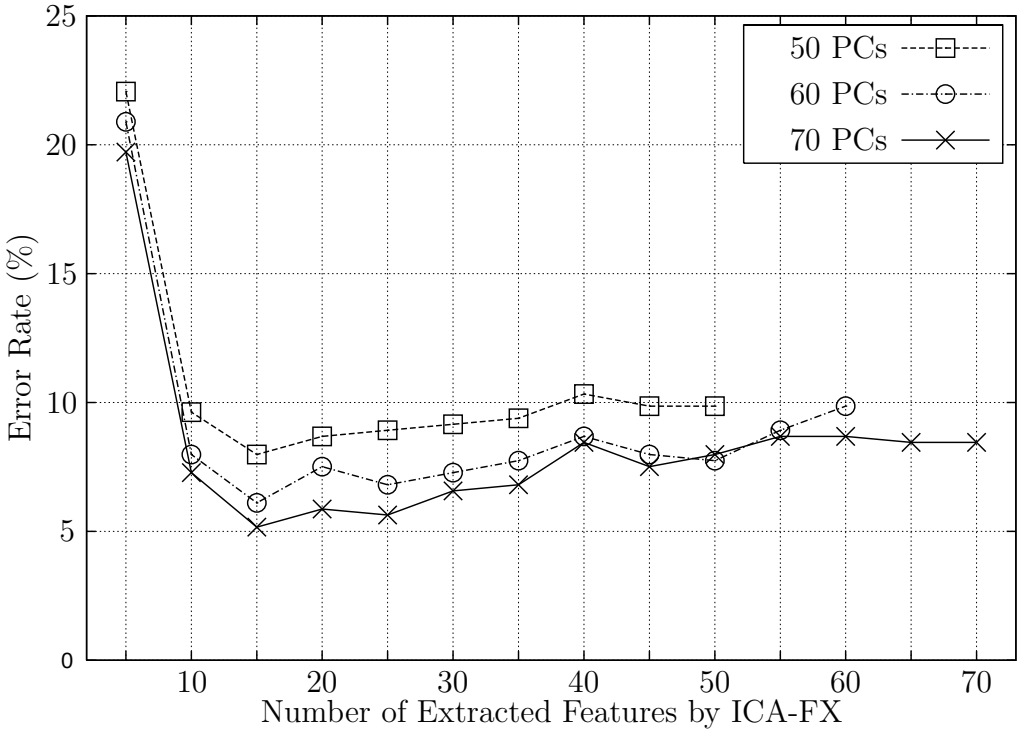


Fig. 17. Performances of ICA-FX on JAFFE database with various number of features used. (50, 60, and 70 principal components were used as inputs to ICA-FX.)

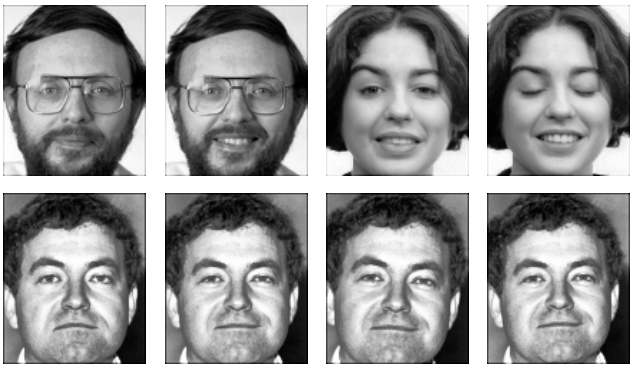


Fig. 18. Sample FERET images after histogram equalization. The top row shows the training images of two subjects, and the bottom row shows one gallery and three probe images of a subject.

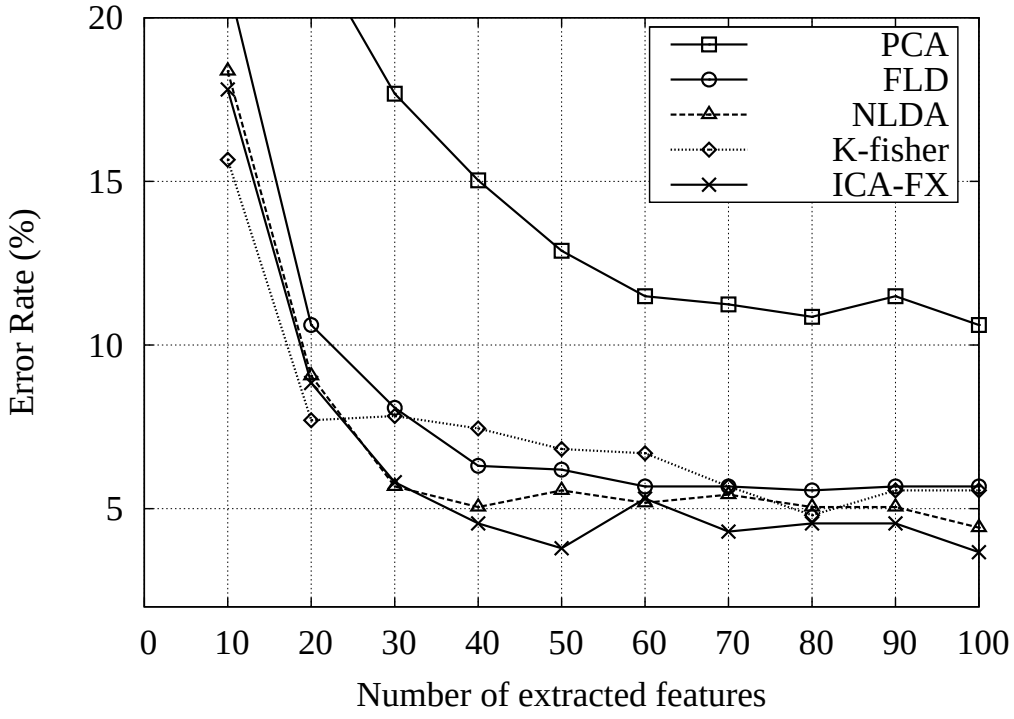
used as training examples, 792 'fa' images were used as the gallery, and 792 'fb' images were used for probing. All of these color images were converted into gray images and then cropped and rescaled to a size of 120 × 100 as in [37]. After this, histogram equalization was applied to the rescaled image and the resulting pixels were normalized to have zero means and unit variances. Figure 18 shows several sample images after histogram equalization. The top row shows the training images of two subjects, and the bottom row shows one gallery and three probe images of one subject.

For this database, we have compared the performance of ICA-FX with those of PCA, FLD, NLDA [38] and kernel Fisherface [4]. Note that as the number of training subject is 200, the dimension of class vector $\boldsymbol{c}$ is 200×1 in ICA-FX.

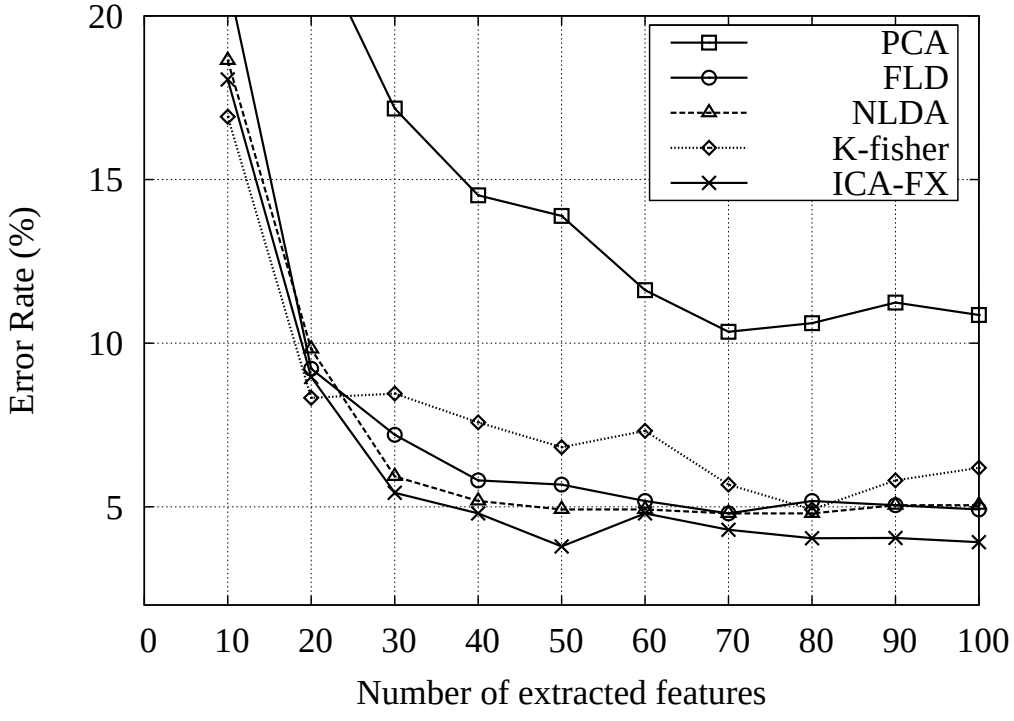
Firstly, PCA was performed on the original 12,000 (120×100) dimensional input space and 140 PCs were obtained. Then, FLD and ICA-FX was applied to this 140 PCs. For NLDA and kernel Fisherface, the original 12,000 pixels were used as input. As a classifier, one nearest neighborhood classifier was used. For kernel Fisherface, polynomial kernels of order one to five and Gaussian kernels $k(i,j) = \exp(-\frac{||x_i - x_j||^2}{N\sigma^2})$ with $\sigma^2 \in \{1, 2, \cdots 10\}$ were tested. Here, N denotes the number of inputs, i.e, it is 12,000 in this case.

Figure 19 shows the error rates of feature extraction methods with various number of extracted features. Fig. 19(a) shows the performances on the gallery which consists of 792 images while (b) corresponds to the performances on the 992 probe images. For both gallery and probe datasets, the performances improves as the number of extracted features increases and they saturate in the end. We can see that the differences in performances on gallery and probe datasets are small. Regardless of the number of extracted features, FLD, NLDA and ICA-FX performed better than PCA by more than 5%. Although the differences are small, ICA-FX performed better than FLD by around 1% in the saturation region. The figure also shows that ICA-FX is slightly better than NLDA in the saturation region but the difference is negligible. Regarding kernel Fisherface, the performance was best for Gaussian kernel with $\sigma^2 = 8$ and it is reported here. The performance of kernel Fisherface was best for small numbers of features, however it gets worse than other methods for large numbers of features. The best error rate of ICA-FX on the probe data was 3.79% when the number of extracted features was 50. On the other hand, the best error rates of FLD, NLDA and kernel Fishface was 4.80%, 4.80% and 4.92% respectively.

Note that the performance enhancement of ICA-FX over FLD on the FERET database is smaller than those on Yale and AT&T databases. Although this might be due to the characteristic of the dataset, the reason for this might also be attributed to the large number of classes in the training of FERET database. Because ICA-FX has a good chance of falling into a local minimum, the large number of classes which results in a high dimensional weight matrix V_a seems to have a bad effect on the performance of ICA-FX. Although this can be mitigated by performing the ICA-FX several times with different initial weight matrices, as a rule of thumb, we recommend the readers to use ICA-FX for the problems with less than 100 or 200 classes.



(a) Gallery



(b) Probe

Fig. 19. Error rates on the Color FERET database with various number of features

Fig. 20. Examples of CMU-PIE database: cropped images of three poses $(c22, c05, and c27)$

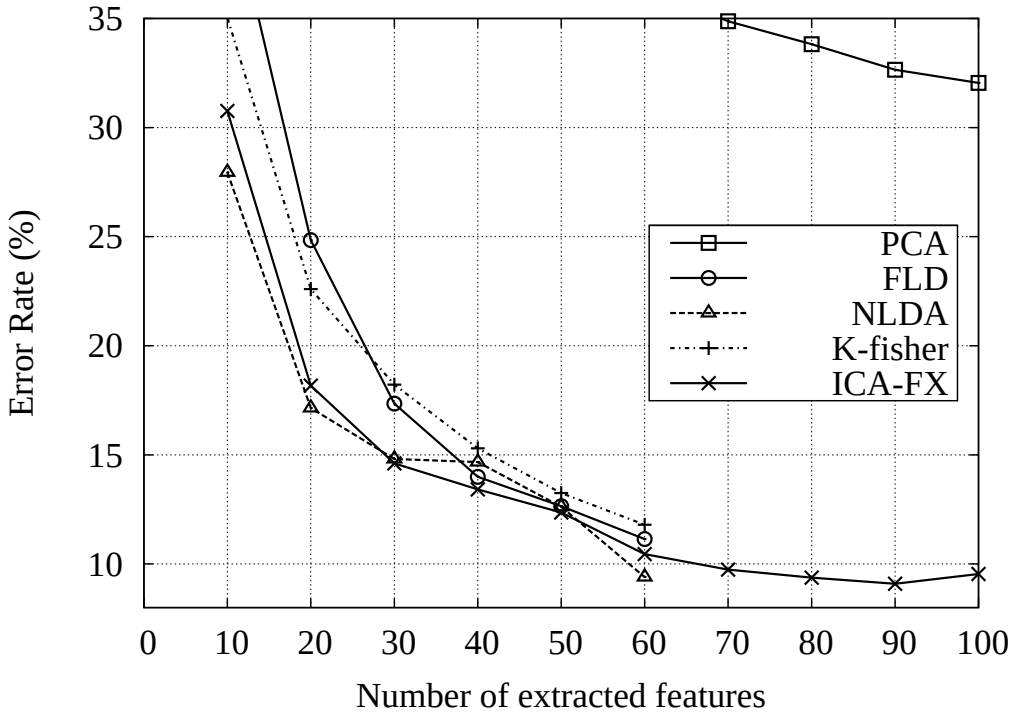


Fig. 21. Error rates on the CMU-PIE database with various number of features

4.5 CMU-PIE Database

The CMU-PIE database contains more than 40,000 facial images of 68 individuals, 21 illumination conditions, 13 poses and with four different expressions. Among them, we selected the images of 65 individuals with three pose indices $(c22, c05, c27)$ because the images of some individuals have defects or does not include all 21 illumination variations. Therefore, total 4,095 (65×3×21) images were used in this experiment.

For each of three poses, three images (illumination 19∼21) of each individual were used for constructing a feature space for training, while the other 18 images were used for testing. The original images were firstly cropped to include only facial part and then downscaled to a standard size of 120×100. Finally, histogram equalization was performed on them. Figure 20 shows the examples of cropped images of the three poses.

We have performed PCA, FLD, NLDA, kernel Fisherface, and ICA-FX on this dataset and show the classification error rates of the test data on Figure 21. For kernel Fisherface, polynomial kernels of order one to five and Gaussian kernels with $\sigma^2 \in \{1, 2, \cdots, 10\}$ were tested as in the FERET Color dataset. The performance of kernel Fisherface was best for Gaussian kernel with $\sigma^2 = 4$. Note that FLD, NLDA, and kernel Fisherface could extract only up to 64 features because the number of class was 65.

In the figure, we can clearly see that PCA is by far worse than the other supervised feature extraction methods. Also ICA-FX and NLDA are better than kernel Fisherface and FLD in most of the cases. The best classification error rates of FLD, kernel Fisherface, NLDA and ICA-FX are 11.14%, 11.80%, 9.40%, and 9.09% respectively.

5 Conclusions

In this paper, the feature extraction algorithm, ICA-FX, has been extended to multi-class problems and it has been applied to face recognition problems. The proposed algorithm is based on the standard ICA and can generate very useful features for classification problems.

Although ICA can be directly used for feature extraction, it does not generate useful information because of its unsupervised learning nature. In the proposed algorithm, class information was added in the learning stage of ICA. The added class information plays a crucial role in extracting useful features for classification. With the additional class information new features containing the maximum information about the class can be extracted.

The proposed algorithm is easy to implement and train, because it uses the standard feed-forward structure and the learning algorithm of ICA. Experimental results for several face databases show that the proposed algorithm performs well in face recognition problems. The number of extracted features of the ICA-FX was relatively small. By this property, it is concluded that the ICA-FX produces features that are in line with the 'law of parsimony', resulting in a better performance.

A Proof of the Theorem

The proof of the theorem undergoes almost the same steps as the one in [27] [12].

At first, the argument that the point $(W = \Lambda\Pi A^{-1}, V = -\Lambda\Pi A^{-1}B)$ is a stationary point of the learning rule (14) can be shown as follows. By substitution, one can easily show that $E\{[I_N - \boldsymbol{\varphi}(\boldsymbol{u})\boldsymbol{f}^T]\} = 0$ and $E\{\boldsymbol{\varphi}(\boldsymbol{u}_a)\boldsymbol{c}^T\} = 0$ at that point. After this, the proof of Theorem 1 in [12] can exactly applied to show the stationarity of the point.

For the second part, to show that the point in question is indeed a stable point, we use a standard tool for analyzing the local asymptotic stability of a stochastic algorithm. It makes use of the derivative of the mean field at a stationary point.

In doing so we introduce a new version of weight matrices Z and K such that

$$\begin{aligned} W^{(t)} &= Z^{(t)}W^* \\ v_{ij}^{(t)} &= k_{ij}^{(t)}v_{ij}^* \ (\neq 0), \qquad 1 \le i \le M, 1 \le j \le N_c \end{aligned} \tag{A.1}$$

where k_{ij} is the (i,j) component of K, W^* and v_{ij}^* are the optimal values of W and v_{ij} which are $\Lambda\Pi A^{-1}$ and $-(\Lambda\Pi A^{-1}B)_{ij}$, respectively. Note that the stability of W and v_{ij} in the vicinity of W^* and v_{ij}^* is equivalent to the stability of Z and k_{ij} in the vicinity of the identity matrix I_N and 1.

With the change of variables from (W, V_a) to (Z, K) the learning rule for W in (14) becomes

$$Z^{(t+1)} = \{I_N - \mu_1 G(Z^{(t)}, K^{(t)})\}Z^{(t)} \tag{A.2}$$

where the (i,j)th element of $G \in \Re^{N\times N}$ is

$$\begin{aligned} G(Z^{(t)}, K^{(t)})_{ij} &= \varphi_i(u_i)f_j - \delta_{ij} \\ &= \begin{cases} \varphi_i((Z^{(t)}W^*\boldsymbol{x})_i + \sum_{n=1}^{N_c} k_{in}^{(t)}v_{in}^*c_n)(Z^{(t)}W^*\boldsymbol{x})_j - \delta_{ij} & \text{if } 1 \le i \le M \\ \varphi_i((Z^{(t)}W^*\boldsymbol{x})_i)(Z^{(t)}W^*\boldsymbol{x})_j - \delta_{ij} & \text{if } M < i \le N \end{cases}. \end{aligned} \tag{A.3}$$

and

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} - \mu_{ij}^{(t)}\varphi_i(u_i)c_j v_{ij}^* v_{ij}^{(t)}, \qquad 1 \le i \le M, 1 \le j \le N_c. \tag{A.4}$$

Here we assume that the learning rate $\mu_{ij}^{(t)}(> 0)$ changes over time t and varies with different index (i,j) such that it satisfies $\mu_{ij}^{(t)}v_{ij}^{(t)}v_{ij}^* = \mu_2$. The assumption is justified because $v_{ij}^{(t)}v_{ij}^* \cong v_{ij}^{*2}$ is positive when $v_{ij}^{(t)}$ is near a stationary point

v_{ij}^*. This assumption holds only after V_a has reached sufficiently near a stable point V_a^*.

Using the fact that $v_{ij}^{(t)} = k_{ij}^{(t)} v_{ij}^*$ we can rewrite (A.4) as

$$k_{ij}^{(t+1)} = [1 - \mu_i^{(t)} g_{ij}(Z^{(t)}, K^{(t)})] k_{ij}^{(t)}, \qquad 1 \le i \le M \tag{A.5}$$

where

$$\begin{aligned} g_{ij}(Z^{(t)}, K^{(t)}) &= \varphi_i(u_i) v_{ij}^* c_j \\ &= \varphi_i((Z^{(t)} W^* \boldsymbol{x})_i + \sum_{n=1}^{N_c} k_{in}^{(t)} v_{in}^* c_n) v_{ij}^* c_j \end{aligned} \tag{A.6}$$

In our problem, $Z \in \Re^{N \times N}$ and $K \in \Re^{M \times N_c}$ constitute an $N^2 + MN_c$ dimensional space, and we can denote this space as a direct sum of Z and K; i.e., $Z \oplus K$. Then the derivative considered here is that of a mapping $H : Z \oplus K \to E\{G(Z,K)Z\} \oplus E\{g_{11}(Z,K)k_{11}\} \oplus \cdots \oplus E\{g_{MN_c}(Z,K)k_{MN_c}\}$ at the stationary point (Z^*, K^*) where $Z^* = I_N$ and $K^* = \mathbf{1}_{MN_c}$, where $\mathbf{1}_{MN_c}$ is the $M \times N_c$ dimensional matrix whose every components are all 1's. The derivative is of $(N^2 + MN_c)^2$ dimension, and if it is positive definite, the stationary point is a local asymptotic stable point. As in [27] [12], because the derivative of the mapping H is very sparse, we can use the first-order expansion of H at the point (Z^*, K^*) rather than trying to use the exact derivatives.

For convenience, let us split H into two functions H^1 and H^2 such that

$$\begin{aligned} &H^1 : Z \oplus K \to E\{G(Z,K)Z\} \in \Re^{N \times N} \\ &H_{ij}^2 : Z \oplus K \to E\{g_{ij}(Z,K)k_{ij}\}, \qquad 1 \le i \le M, 1 \le j \le N_c. \end{aligned} \tag{A.7}$$

Note that $H = H^1 \oplus H^2$. To get the first order linear approximation of the function at a stationary point (Z^*, K^*), we evaluate H^1 and H^2 near a small variation of the stationary point $(Z, K) = (Z^* + \mathcal{E}, K^* + \boldsymbol{\varepsilon})$, where $\mathcal{E} \in \Re^{N \times N}$ and $\boldsymbol{\varepsilon} \in \Re^{M \times N_c}$.

With the independence and zero mean assumptions on e_i's and class label c, which results in the independence of e_i's and c_j's, it becomes

$$\begin{aligned} &H_{ij}^1(I_N + \mathcal{E}, \mathbf{1}_{MN_c} + \boldsymbol{\varepsilon}) \\ &= \begin{cases} \mathcal{E}_{ij} E\{\dot{\varphi}_i(e_i) e_j^2\} + E\{\varphi_i(e_i)e_i\}\mathcal{E}_{ji} + \sum_{m=1}^{M} \mathcal{E}_{im} E\{\dot{\varphi}_i(e_i)\alpha_j\alpha_m\} \\ \qquad -E\{\dot{\varphi}_i(e_i)\alpha_j\beta_i\} + o(\mathcal{E}) + o(\boldsymbol{\varepsilon}) & \text{if} \quad 1 \le i, j \le M \\ \mathcal{E}_{ij} E\{\dot{\varphi}_i(e_i) e_j^2\} + E\{\varphi_i(e_i)e_i\}\mathcal{E}_{ji} + \sum_{m=1}^{M} \mathcal{E}_{im} E\{\dot{\varphi}_i(e_i)\alpha_j\alpha_m\} \\ \qquad +o(\mathcal{E}) + o(\boldsymbol{\varepsilon}) & \text{if} \quad M < i \le N,\ 1 \le j \le M \\ \mathcal{E}_{ij} E\{\dot{\varphi}_i(e_i) e_j^2\} + E\{\varphi_i(e_i)e_i\}\mathcal{E}_{ji} + o(\mathcal{E}) + o(\boldsymbol{\varepsilon}) & \text{if} \quad M < j \le N \end{cases} \end{aligned} \tag{A.8}$$

and

$$
\begin{aligned}
&H^2_{ij}(I_N + \mathcal{E}, \mathbf{1}_{MN_c} + \boldsymbol{\varepsilon}) \\
&= -v^*_{ij} \sum_{m=1}^{M} \mathcal{E}_{im} E\{\dot{\varphi}_i(e_i)\alpha_m c_j\} + v^*_{ij} E\{\dot{\varphi}_i(e_i)\beta_i c_j\} + o(\mathcal{E}) + o(\boldsymbol{\varepsilon}) \qquad \text{(A.9)} \\
&\qquad\qquad 1 \le i \le M, 1 \le j \le N_c.
\end{aligned}
$$

Here, $\alpha_i \triangleq \sum_{n=1}^{N_c} v^*_{in} c_n$ and $\beta_i \triangleq \sum_{n=1}^{N_c} v^*_{in} c_n \varepsilon_{in}$.

Now, we develop the local stability conditions case by case. As mentioned before, the detailed derivation is almost the same as that in [12].

(Case 1) $i, j > M$

In this case, H^1_{ij} and H^1_{ji} only depend on $\mathcal{E}_{ij}$ and $\mathcal{E}_{ji}$ and are represented as

$$
\begin{aligned}
&\begin{bmatrix} H^1_{ij} \\ H^1_{ji} \end{bmatrix} = \begin{bmatrix} E\{\dot{\varphi}_i(e_i)\}E\{e_j^2\} & E\{\varphi_i(e_i)e_i\} \\ E\{\varphi_i(e_j)e_j\} & E\{\dot{\varphi}_j(e_j)\}E\{e_i^2\} \end{bmatrix} \begin{bmatrix} \mathcal{E}_{ij} \\ \mathcal{E}_{ji} \end{bmatrix} \triangleq D_{ij} \begin{bmatrix} \mathcal{E}_{ij} \\ \mathcal{E}_{ji} \end{bmatrix} \qquad \text{if} \quad i \neq j \\
&H^1_{ii} = [E\{\dot{\varphi}_i(e_i)e_i^2\} + E\{\varphi_i(e_i)e_i\}]\mathcal{E}_{ii} \triangleq d_i \mathcal{E}_{ii}.
\end{aligned} \qquad \text{(A.10)}
$$

Thus for $i \neq j$, Z_{ij} and Z_{ji} are stabilized when D_{ij} is positive definite. And if $i = j$, Z_{ii} is stabilized when d_i is positive. Using the fact that $E\{\varphi_i(e_i)e_i\} = 1$ $\forall i = 1, \cdots, N$, we can show that the local stability condition for the pair (i, j) when $i, j > M$ is (16).

(Case 2) $i \le M, j > M$

In this case, H^1_{ij} and H^1_{ji} are dependent not only on $\mathcal{E}_{ij}$ and $\mathcal{E}_{ji}$ but also on all $\mathcal{E}_{jm}$, $m = 1, \cdots, M$. Thus for a fixed j, we augment all the H^1_{ij} and H^1_{ji}, $i = 1, \cdots, M$, and construct a $2M$-dimensional vector $\boldsymbol{H}_j \triangleq [H^1_{1j}, \cdots, H^1_{Mj}, H^1_{j1}, \cdots, H^1_{jM}]^T$. Now this augmented vector $\boldsymbol{H}_j$ depends only on $\boldsymbol{\mathcal{E}}_j \triangleq [\mathcal{E}^1_{1j}, \cdots, \mathcal{E}_{Mj}, \mathcal{E}_{j1}, \cdots, \mathcal{E}_{jM}]^T$ and can be represented as a linear equation $\boldsymbol{H}_j = \boldsymbol{D}_j \boldsymbol{\mathcal{E}}_j$, using an appropriate matrix $\boldsymbol{D}_j \in \Re^{2M \times 2M}$. The stability of $\boldsymbol{Z}_j = [Z_{1j}, \cdots, Z_{Mj}, Z_{j1}, \cdots, Z_{jM}]^T$ for $j > M$ is equivalent to the positive definiteness of $\boldsymbol{D}_j$ and it can be checked by investigating the sign of the $\boldsymbol{H}_j^T \boldsymbol{\mathcal{E}}_j$.

Substituting (A.8) and using $E\{\varphi_i(e_i)e_i\} = 1\ \forall i = 1, \cdots, N$, we get

$$\begin{aligned}\boldsymbol{H}_j^T\boldsymbol{\mathcal{E}}_j &= \sum_{i=1}^{M}(H_{ij}^1\mathcal{E}_{ij} + H_{ji}^1\mathcal{E}_{ji}) \\ &= \sum_{i=1}^{M}[E\{\dot{\varphi}_i(e_i)e_j^2\}\mathcal{E}_{ij}^2 + 2\mathcal{E}_{ij}\mathcal{E}_{ji} + E\{\dot{\varphi}_j(e_j)e_i^2\}\mathcal{E}_{ji}^2] + E\{\dot{\varphi}_j(e_j)\}E\{(\sum_{i=1}^{M}\mathcal{E}_{ji}\alpha_i^*)^2\}.\end{aligned} \tag{A.11}$$

If we assume that $\dot{\varphi}_j(\cdot)$ is nonnegative, as we did in the proof of the uniqueness of the scalar λ_j, the last term is nonnegative. Thus, a sufficient condition for this equation to be positive is to make the first term positive, and this condition is satisfied if and only if equation (16) holds. Therefore, (16) becomes a sufficient condition for the local stability of $\boldsymbol{Z}_j$.

(Case 3) $i, j \leq M$

In this case, because H_{ij}^1 and H_{ij}^2 are dependent both on $\mathcal{E}$ and $\boldsymbol{\varepsilon}$, we construct a new vector and investigate the stability condition of the vector as in the previous case.

Consider the $M\times M+M\times N_c$ dimensional vectors $\boldsymbol{H} \triangleq [H_{11}^1, H_{12}^1, \cdots, H_{MM}^1, H_{11}^2, \cdots, H_{MN_c}^2]^T$ and $\boldsymbol{\mathcal{E}} \triangleq [\mathcal{E}_{11}, \mathcal{E}_{12}, \cdots, \mathcal{E}_{MM}, \varepsilon_{11}, \cdots, \varepsilon_{MN_c}]^T$. Using (A.8) and (A.9), $\boldsymbol{H}$ can be represented as the linear equation $\boldsymbol{H} = \boldsymbol{D}\boldsymbol{\mathcal{E}}$, where $\boldsymbol{D}$ is an appropriate matrix. Thus, the stability of the $\mathcal{Z} = [Z_{11}, Z_{12}\cdots, Z_{MM}]^T$ and K can be checked using the same procedure as the previous case.

$$\begin{aligned}\boldsymbol{H}^T\boldsymbol{\mathcal{E}} &= \sum_{i=1}^{M}\sum_{j=1}^{M}H_{ij}^1\mathcal{E}_{ij} + \sum_{i=1}^{M}\sum_{j=1}^{N_c}H_{ij}^2\varepsilon_{ij} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M}(\mathcal{E}_{ij}^2E\{\dot{\varphi}_i(e_i)e_j^2\} + \mathcal{E}_{ij}\mathcal{E}_{ji}) + \sum_{i=1}^{M}[E\{\dot{\varphi}_i(e_i)\}E\{(\beta_i - \sum_{j=1}^{M}\alpha_j\mathcal{E}_{ij})^2\}]\end{aligned} \tag{A.12}$$

The last term is nonnegative with the assumption of $\dot{\varphi}_i(\cdot) \geq 0$, and a sufficient condition for the double summation to be positive is (16). Thus, $\mathcal{Z} \oplus K$ is locally stable if condition (16) holds.

Combining the stability conditions for the case 1, 2, and 3, we conclude that the learning rule (14) for ICA-FX is locally asymptotically stable at the stationary point if the condition (16) holds.

References

[1] M. Turk, A. Pentland, Face recognition using eigenfaces, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1991, pp. 586–591.

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

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

[4] M.-H. Yang, Face recognition using kernel methods, Advances of Neural Information Processing Systems 14.

[5] J. Lu, N. Plataniotis, A. N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, IEEE Trans. on Neural Networks 14 (1) (2003) 117 – 126.

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

[7] A. Bell, T. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation 7 (6).

[8] M. Bartlett, T. Sejnowski, Viewpoint invariant face recognition using independent component analysis and attractor networks, Neural Information Processing Systems-Natural and Synthetic 9 (1997) 817–823.

[9] G. Donato, M. Bartlett, J. Hager, P. Ekman, T. Sejnowski, Classifying facial actions, IEEE Trans. on Pattern Analysis and Machine Intelligence 21 (10) (1999) 974–989.

[10] N. Kwak, C.-H. Choi, C.-Y. Choi, Feature extraction using ica, in: Proc. Int'l Conf. on Artificial Neural Networks 2001, Vienna Austria, 2001, pp. 568 – 573.

[11] N. Kwak, C.-H. Choi, A new method of feature extraction and its stability, in: Proc. Int'l Conf. on Artificial Neural Networks 2002, Madrid Spain, 2002, pp. 480–485.

[12] N. Kwak, C.-H. Choi, Feature extraction based on ica for binary classification problems, IEEE Trans. on Knowledge and Data Engineering 15 (6).

[13] N. Yukinawa, S. Oba, K. Kato, S. Ishii, Multi-class pattern classification based on a probabilistic model of combining binary classifiers, in: Proc. Int'l Conf. on Artificial Neural Networks 2005, 2005, pp. 337–342.

[14] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. on Neural Networks 13 (2) (2002) 415–425.

[15] D. Tax, R. Duin, Using two-class classifiers for multiclass classification, in: Proc. Int'l Conf. on Image Processing 2002, 2002.

[16] T.-F. Wu, R. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 979–1005.

[17] J. Herault, C. Jutten, Space or time adaptive signal provessing by neural network models, in: Proc. AIP Conf. Neural Networks Computing, Vol. 151, Snowbird, UT, USA, 1986, pp. 206–211.

[18] J. Cardoso, Source separation using higher order moments, in: Proc. ICASSP, 1989, pp. 2109–2112.

[19] P. Comon, Independent component analysis, a new concept?, Signal Processing 36 (1994) 287–314.

[20] D. Obradovic, G. Deco, Blind source seperation: are information maximization and redundancy minimization different?, in: Proc. IEEE Workshop on Neural Networks for Signal Processing 1997, Florida, 1997.

[21] J. Cardoso, Infomax and maximum likelifood for blind source separation, IEEE Signal Processing Letters 4 (4).

[22] T.-W. Lee, M. Girolami, A. Bell, T. Sejnowski, A unifying information -theretic framework for independent component analysis, Computers and Mathematics with Applications 31 (11).

[23] T.-W. Lee, M. Girolami, T. Sejnowski, Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources, Neural Computation 11 (2).

[24] L. Xu, C. Cheung, S.-I. Amari, Learned parametric mixture based ica algorithm, Neurocomputing 22 (1-3) (1998) 69 – 80.

[25] A. Hyvarinen, E. Oja, P. Hoyer, J. Hurri, Image feature extraction by sparse coding and independent component analysis, in: Proc. Fourteenth International Conference on Pattern Recognition, Brisbane, Australia, 1998.

[26] T. Cover, J. Thomas, Elements of Information Theory, John Wiley & Sons, 1991.

[27] J. Cardoso, On the stability of source separation algorithms, Journal of VLSI Signal Processing Systems 26 (1) (2000) 7 – 14.

[28] N. Kwak, C.-H. Choi, N. Ahuja, Face recognition using feature extraction based on independent component analysis, in: Proc. Int'l Conf. on Image Processing 2002, Rochester, NY, 2002.

[29] E. Micheli-Tzanakou, Supervised and unsupervised pattern recognition, CRC Press, 2000.

[30] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Proc. 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL, 1994.

[31] P. Phillips, H. Moon, S. Rizvi, P. Rauss, The feret evaluation methodology for face recognition algorithms, IEEE Trans. on Pattern Recognition and Machine Intelligence 22 (10) (2000) 1090–1104.

[32] National institute of standars and technology, the color feret database, http://www.nist.gov/humanid/colorferet.

[33] T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression database,.

[34] M. J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, IEEE Trans. on Pattern Analysis and Machine Intelligence 21 (12) (1999) 1357 – 1362.

[35] T. Mitchell, Machine learning, McGraw-Hill, 1997.

[36] J. Lu, K. Plataniotis, A. Venetsanopoulos, Regularization studies of linear discriminant analysis in small sample size scenarios with applications to face recognition, Pattern Recognition Letters 26 (2005) 181–191.

[37] C. Kim, C.-H. Choi, Image covariance-based subspace method for face recognition, Pattern Recognition 40 (2007) 1592–1604.

[38] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, IEEE Trans. on Pattern Recognition and Machine Intelligence 27 (1) (2005) 4–13.