Facial Expression Recognition by Supervised Independent Component Analysis Using MAP Estimation

Fan CHEN(a), Nonmember and Kazunori KOTANI¹, Member

SUMMARY Permutation ambiguity of the classical Independent Component Analysis (ICA) may cause problems in feature extraction for pattern classification. Especially when only a small subset of components is derived from data, these components may not be most distinctive for classification, because ICA is an unsupervised method. We include a selective prior for de-mixing coefficients into the classical ICA to alleviate the problem. Since the prior is constructed upon the classification information from the training data, we refer to the proposed ICA model with a selective prior as a supervised ICA (sICA). We formulated the learning rule for sICA by taking a Maximum a Posteriori (MAP) scheme and further derived a fixed point algorithm for learning the de-mixing matrix. We investigate the performance of sICA in facial expression recognition from the aspects of both correct rate of recognition and robustness even with few independent components.

key words: facial expression recognition, supervised independent component analysis, fixed-point algorithm

1. Introduction

Various methods have been proposed for automatic recognition of facial expression in the past several decades, which could be roughly classified into three categories: 1) Appearance-based method, represented by eigenfaces, fisherfaces and other methods using machine-learning techniques, such as neural networks and Support Vector Machine (SVM); 2) Model-based methods, including graph matching, optical-flow-based method and others; and 3) Hybrids of appearance based and model-based methods, such as Active Appearance Model (AAM). Detailed review of these methods could be found in two surveys in Refs. [1], [2]. Appearance-based methods are superior to model-based methods in system complexity and performance reproducibility. Further, appearance-based methods allow efficient characterization of a low-dimensional subspace within the overall space of raw image measurement, which deepen our understanding of facial expressions from their manifolds in subspace, and provide a statistical framework for the theoretical analysis of system performance. ICA, a powerful technique for blind source separation, was applied to facial expression recognition by Bartlett et al. for feature extraction[3]. They argued that facial expression consists of those features standing for minor, non-rigid, local variations of faces [3]. Structural information for those local variations are related to higher-order statistics, which could be well extracted by ICA[5]. The efficiency of ICA in extracting features for facial expression recognition has been verified by many previous works [4], [6], [7].

The major purpose of the present work is to improve the performance of ICA in facial expression recognition. In the classical ICA, the derived independent components are in random order, i.e., permutation ambiguity, where the original order provides no information on the significance of components in discrimination[8]. As a result, the derived independent components may not be most distinctive for the classification task, especially when only a small subset of components is derived. Feature selection must be performed along with the feature extraction. The selection can be applied before, during or after ICA. In Ref. [4], Best Individual Feature (BIF) selection was adopted, where features were chosen according to some defined criteria individually. Methods by means of Sequential Forward Selection (SFS) and Sequential Floating Forward Selection (SFFS) were also proposed[9]. Since the selection is performed after ICA, the features are limited to those chosen from the set of independent components obtained. To create a candidate set with enough representative features for discrimination, a large number of independent components should be learned, which may be expensive in computational cost. It is meaningful to search for a way to affect the selection of features before or during ICA. GEMC[10] makes a selection before ICA by heuristically replacing PCA with a discriminant analysis as the pre-processing to ICA, which still lacks a mathematical explanation. ICA in a local facial residue space is also proposed for face recognition, which can be regarded as using the pre-specified residue space to limit the selection of independent components before applying ICA [11].

We propose an approach to implement the feature selection during the learning of independent components. A constraint ICA has been proposed for the analysis of EEG signals, where all components should be sparse and close to a supplied reference signal by including a correlation term [12]. In our case, we try to design a method to let those components with higher degrees of class separation emerge easier than others. The classical ICA in Ref. [13] was shown to be derivable under the scheme of Maximum Log-Likelihood (MLL) estimation [14]. Instead of using the uniform prior for de-mixing coefficients in MLL, we take the Maximum a Posteriori (MAP) estimation. A prior defined by the degree of class separation is introduced to the de-mixing coefficients, which in turn increases the proba-

¹The authors are with the school of information science, Japan Advanced Institute of Science and Technology, 923-1292 Japan.
a) E-mail: chen-fan@jaist.ac.jp
DOI: 10.1093/ietisy/e91-d.2.341

Copyright © 2008 The Institute of Electronics, Information and Communication Engineers
bility of the corresponding independent component being significant in classification. In Ref. [15], a neural-network based supervised ICA has been proposed, where a three-layered feed-forward network minimizes the dependency of hidden outputs and also minimizes errors between output signals and their targets. Compared to the neural-network based version, our method uses the MAP estimation, which provides a base for further derivations of various approximated learning algorithms. Especially, a fast fixed-point version has been derived in the present work, which needs fewer iterations than feed-forward neural network with fixed learning rate, and is more robust in convergence.

In Sect. 2, we formulate sICA and derive the learning rule in the fixed-point form. The algorithm for facial expression recognition is also given. In Sect. 3, numerical experiments are made and the performance of our proposed algorithm is investigated by making a comparison with ICA. We also discuss the influence of the introduced selective prior. Finally, we summarize the present paper and explain our future work.

2. Facial Expression Recognition by sICA

We propose sICA to improve the performance of ICA in the extraction of distinctive features. In this section, two architectures of ICA-based facial expression recognition proposed in Ref. [4] are explained. And we explain the generative model of sICA and give its MAP criterion. Then we derive a fast learning method for sICA by using a fixed-point algorithm for facial expression recognition.

2.1 Architectures of ICA-Based Facial Expression Recognition

Assume we have \( N \) training images samples \( X = \{ [x^{(k)}] | k \in \{1, \ldots, K\}, i \in \{1, \ldots, N_k\} \} \) from \( K \) classes, with \( N_k \) samples in the \( k \)-th class, and satisfy \( N = \sum_{k=1}^{K} N_k \). \( x^{(k)} = [x^{(ki)}_1, \ldots, x^{(ki)}_L]^T \) is a \( L \)-dimensional vector. In Ref. [4], two architectures for ICA-based face/facial expression recognition have been proposed. In both architectures, PCA is applied to the image data \( X \) before performing ICA. Assume that \( Y \) is the sample data derived from PCA projection of \( X \), i.e., \( Y = W_{\text{pca}}X \) and \( X = V_{\text{pca}}Y \). We have \( Y = \{ [y^{(k)}] | k \in \{1, \ldots, K\}, i \in \{1, \ldots, N_k\} \} \).

The \( i \)-th sample of class \( k \), \( y^{(ki)} = [y^{(ki)}_1, \ldots, y^{(ki)}_D]^T \), is a \( D \)-dimensional vector. \( W_{\text{pca}} \) is the matrix of PCA eigenvectors which satisfies \( V_{\text{pca}} = [W_{\text{pca}}]^T \). In Architecture I, ICA is performed on PCA bases \( V_{\text{pca}} \). In Architecture II, ICA is performed on PCA coefficients \( Y \), i.e.,

\[
\begin{align*}
\text{Arch. I} : \quad & [V_{\text{pca}}]^T V S \rightarrow X = [Y^T V] S, \\
\text{Arch. II} : \quad & Y = V S \rightarrow X = [V_{\text{pca}} V] S.
\end{align*}
\]

\( V \) and \( S \) are the mixing matrix and independent source matrix derived by ICA, respectively. If \( Q \) sources are extracted, \( S \) is a \( Q \)-row \( L \)-col matrix in Arch. I and a \( Q \)-row \( N \)-col matrix in Arch. II. We give the diagrams for both architectures in Fig. 1. In Arch. I, each pixel of an image is a mixed output from the corresponding pixel of independent image sources, while in Arch. II, each pixel of an image is produced from independent pixel sources for that image under different mixing styles. When applied to pattern recognition, Arch. I uses each row of mixing matrix \( [Y^T V] \) for classification, while Arch. II recognizes patterns from the column of pixel sources \( S \) corresponding to a given image.

In this paper, we adopt Arch. II and perform sICA on the PCA coefficients \( Y \) instead of directly on the image data \( X \). Accordingly, final bases for extracting features are computed as \( W = W W_{\text{pca}} \) with \( W = V^{-1} \). Let \( \bar{X} = \{ x_n | n \in \{1, \ldots, N\} \} \) be the matrix by putting all testing images into different columns, and let \( N \) be the number of samples in the testing set. We define the true classified labels as \( Z = \{ z_n | n \in \{1, \ldots, K\} \} \) for observed data, and define a recognition rate as

\[
r_c = \frac{1}{N} \sum_{n=1}^{N} \delta(z_n, z^*_n),
\]

where \( \delta(x, y) \) is the Kronecker delta. \( z^*_n \) is the estimated label value which belongs to the class whose center is the closest to the current feature vector, i.e.,

\[
z^*_n = \arg \min_k D(s_n, \bar{y}^{(k)})
\]

and \( s_n = W^T x_n \). \( \bar{y}^{(k)} = \sum_{i=1}^{N_k} y^{(ki)} \) is the mean of PCA-projected training samples belonging to class \( k \). \( D(s_n, s_k) \) is a distance function between two feature vectors, and can be defined in various ways. Especially, we adopt the following form in the present paper, i.e.,

\[
D(s_n, s_k) = 1 - \frac{s_n^T s_k}{|s_n||s_k|}
\]

which has been suggested to be the best metric for ICA Architecture II in many previous works [7], [16]. A block diagram for the whole process is given in Fig. 2.
2.2 Generative Model of sICA

Figure 3 illustrates the generative model of sICA. Provided \( Y \) as the training dataset, the classical ICA in Arch. II assumes that these samples are generated from \( Q \) statistically independent sources. \( S = [s^{(k)} | k \in \{1, \ldots, K\}, i \in \{1, \ldots, N_k\}] \) represents the signals generated by those sources, where \( s^{(k)} = [s^{(k)}_1, \ldots, s^{(k)}_Q]^T \) corresponds to \( y^{(k)} \). Those signals from different sources are linearly mixed, i.e., \( Y = VS \), where the \( D \)-row \( Q \)-column matrix \( V \) is for the mixing coefficients. The purpose of ICA is to search for the coefficients \( V \) that make the sources as statistically independent as possible. If we let \( W = V^{-1} \) be the inverse (or pseudo-inverse) of \( V \), \( W \) is the de-mixing matrix and satisfies \( S = WY \). For any sample \( y \), the extracted feature in ICA will be \( s = Wy \). Note that we consider the noiseless case of ICA in the present paper.

The learning rule proposed by Bell and Sejnowski [13] could also be derived by maximizing the log-likelihood criterion, i.e., \( V = \arg \max_V \log P(Y|V) \) [14]. Motivated by the reasons described in the introduction, we search for a way to make a selection of features during ICA so that those independent components with higher degrees of class separation are easier to find than others, which is achieved by introducing a prior distribution for the coefficients. We derive the learning rule of \( V \) by maximizing the following criterion,

\[
V = \arg \max_V \log P(Y|V) = \arg \max_V [\log P(Y|V) + \log P(V)] .
\] (6)

By defining the prior function \( P(V) \) as a function of the classification information, we include the prior information into ICA to produce sICA base for pattern classification. Especially, we define the prior as follows:

\[
P(V) = P(W) = \prod_q P_w(w_q),
\] (7)

\[
P_w(w) = \frac{1}{Z_w} \exp(-\lambda w^T M_s(Y) w^T)
\] (8)

where \( w_q = [w_{q1}, \ldots, w_{qD}] \) and \( W = [w_1^T, \ldots, w_Q^T]^T \). \( Z_w \) is the partition function, which reads

\[
Z_w = \int \exp(-\lambda w^T M_s(Y) w^T) dw.
\] (9)

We define:

\[
M_s(Y) = M_{bc}(Y) - M_{wc}(Y),
\] (10)

where \( M_{bc}(Y) \) and \( M_{wc}(Y) \) are the between-class and within-class scatter matrices, defined by Eq. (11).

\[
M_{bc}(Y) = \frac{1}{N} \sum_k N_k (\bar{y}^{(k)} - \bar{y})(\bar{y}^{(k)} - \bar{y})^T,
\]

\[
M_{wc}(Y) = \frac{1}{N} \sum_k \sum_i (y^{(ki)} - \bar{y}^{(k)})(y^{(ki)} - \bar{y}^{(k)})^T .
\] (11)

\( \bar{y}^{(k)} \) represents the mean vector for samples in class \( k \), and \( \bar{y} \) is the mean value for all samples. \( \lambda \) is a hyper-parameter introduced to control the influence of the prior. For \( \lambda > 0 \), an independent component whose de-mixing coefficients are more distinctive in classification will have a higher prior probability.

2.3 Fast sICA by Fixed-Point Algorithm

We have derived sICA by using the natural gradient algorithm [17] and reported its performance in Ref. [18]. There are still some questions left in the natural gradient version of sICA. The iterative computation under the fixed learning rate \( \eta \) in the natural gradient algorithm may be very slow or even fail to converge. The gradient descedent scheme only provides a local optimum, therefore performance is significantly limited by the initial searching position. We thus further derive the learning rules of sICA in a fixed-point algorithm. The purposes are to improve the stability of convergence by adaptive learning rate \( \eta \) and to provide better optima by searching in a wider range. Based on the short introduction on fixed-point sICA that we made in Ref. [19], we will give more detailed discussions on the formulation of sICA and make further investigations on the characteristics of sICA in the present paper. Performances under other priors will also be studied.

Without loss of generality, we assume all sample data \( \{y^{(k)}\} \) are mean-centered. A pre-whitening preprocessing is first applied to sample data \( Y \) to reduce the complexity of ICA by de-correlating sample data. We solve the eigen-decomposition on the covariance matrix \( E[yy^T] \) as \( E[yy^T] = U^T \Lambda U \), where \( U \) is the matrix of eigen-vectors and \( \Lambda \) is the diagonal matrix of all eigen-values. Pre-whitened sample matrix \( Y^{pw} \) is computed as \( Y^{pw} = W^{pw} Y \), where \( W^{pw} = U^T \Lambda^{1/2} U \) is the whitening matrix. Note that pre-whitening step has a different purpose from that of PCA preprocessing. PCA performed before ICA provides the data for ICA decomposition in Architecture I, and helps to reduce the data dimensions in Architecture II. On the other hand, the pre-whitening preprocessing inside sICA is used in both Architectures for data de-correlation.

Independent components are then estimated by maximizing the criterion in Eq. (6) on pre-whitened samples \( Y^{pw} \).
We calculate the scatter matrix \( M_s(Y_{pw}) \) and then diagonalize it as
\[
M_s(Y_{pw}) = AA^T, \tag{12}
\]
where \( M_s(Y_{pw}) = M_{bc}(Y_{pw}) - M_{wc}(Y_{pw}) \) and \( A \) is a \( D \times D \) orthogonal matrix which satisfies \( A^TA = I \) and \( \hat{Y} = A^T Y_{pw} \). We use \( \hat{x} \) to denote the correspondence of variable \( x \) that is calculated from \( \hat{Y} \). It is easy to derive \( \hat{w}_x = wA \), which is a \( 1 \times D \) vector. We rewrite the priors in Eqs. (7)-(8) as
\[
P(W_{pw}) = \prod_q P_q(w_q^{pw}) = \prod_q P_q(\hat{w}_q), \tag{13}
\]
\[
\hat{p}_q(\hat{w}_q) = \frac{1}{Z_{\hat{w}}} \exp \{\lambda \hat{\Lambda}_s(\hat{Y}) \hat{w}_q^T\}, \tag{14}
\]
respectively.
\[
Z_{\hat{w}} = \int \exp \{\lambda \hat{\Lambda}_s(\hat{Y}) \hat{w}_q^T\} d\hat{w}_q. \tag{15}
\]
Accordingly, the a Posteriori reads
\[
\log P(\hat{Y}\|Y_{pw}) = N \log |\det W| + \sum \log P_q \left( \sum_d \hat{w}_{qd} \hat{y}_{d}^{(\hat{k})} \right) + \lambda \sum_q \hat{w}_q \hat{\Lambda}_s(\hat{Y}) \hat{w}_q^T + \text{Const}, \tag{16}
\]
where \(|x|\) denotes the absolute value of \( x \), and \( \det \) calculates the determinant of a matrix. Derivation of \( \log P(\hat{Y}\|\hat{Y}) \) can be found in Ref. [13]. \( P_q(x) \) defines the prior of sources, which takes super-Gaussian distribution, i.e., \( P_q(x) \propto 1/cosh(x) \), for the sparseness between sources. Let \( (y_{pw}^{(\hat{k})}) \) be the pre-whitened result of \( y^{(\hat{k})} \), i.e., \( y_{pw}^{(\hat{k})} = W_{pw} y^{(\hat{k})} \), \( \hat{y}^{(\hat{k})} \) is also whitened, which is derived from
\[
E[\hat{y}^{(\hat{k})}\hat{y}^{(\hat{k})^T}] = A^T E[y_{pw}^{(\hat{k})}][y_{pw}^{(\hat{k})^T}] A = I \tag{17}
\]
due to the orthogonality of \( A \). \( I \) is the indentity matrix. If we restrict the searching problem of independent components in the whitened space, we have uncorrelatedness and unit variance of \( \hat{s}^{(\hat{k})} \) and \( |\det \hat{W}| \) is constant because
\[
1 = |\det E[\hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T}]| = |\det E[\hat{W} \hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T}] \hat{W}^T| = |\det \hat{W}||\det E[\hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T}]||\det \hat{W}^T|. \tag{18}
\]
Further, \( \hat{W} \) is orthogonal due to
\[
E[\hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T}] = E[\hat{W} \hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T} \hat{W}^T] = \hat{W} E[\hat{s}^{(\hat{k})}\hat{s}^{(\hat{k})^T}] \hat{W}^T = \hat{W} \hat{W}^T = I, \tag{19}
\]
which derives \( ||\hat{w}_q|| = 1 \). In order to optimize the a Posteriori under the constraint \( ||\hat{w}_q|| = 1 \), we define the criteria by using the Lagrange multiplier,
\[
\mathcal{L} = \log P(\hat{\hat{Y}}|\hat{Y}) + \sum_q \alpha_q (\hat{w}_q \hat{w}_q^T - 1) \tag{20}
\]
with its first-order differential with respect to \( \hat{w}_{qd} \) being
\[
\frac{\partial \mathcal{L}}{\partial \hat{w}_{qd}} = \frac{\partial \log P(\hat{\hat{Y}}|\hat{Y})}{\partial \hat{w}_{qd}} + 2 \alpha_q \hat{w}_{qd} = 2 \alpha_q \hat{w}_{qd} - \sum_k p_q(d) \hat{w}_{qd} \hat{y}_{d}^{(\hat{k})} \hat{y}_{d}^{(\hat{k})^T} + 2 \lambda \hat{w}_q \hat{\Lambda}_s(\hat{Y}) \hat{w}_q^T \tag{21}
\]
with \( p_q(x) = P_q'(x)/P_q(x) \) and \( P_q'(x) \) is the derivative of \( P_q(x) \) with respect to \( x \). Note that we use \( [\Lambda_s(\hat{Y})]_{dd} \) to denote the vector from the \( d \)-th row of \( \Lambda_s(\hat{Y}) \) and will use \( [\Lambda_s(\hat{Y})]_{dd} \) to represent the \( d \)-th row of \( \Lambda_s(\hat{Y}) \) later. In a vector form of differential, we derive
\[
\frac{\partial \mathcal{L}}{\partial \hat{w}_q} = -\lambda NE[g(\hat{w}_q \hat{y}) \hat{y}^T] + 2 \lambda \hat{w}_q \Lambda_s(\hat{Y}) \hat{w}_q^T + 2 \alpha_q \hat{w}_{qd}. \tag{22}
\]
To maximize the criterion, we set the first-order differential to zero, i.e.,
\[
\frac{\partial \mathcal{L}}{\partial \hat{w}_q} = 0. \tag{23}
\]
Under the constraint \( ||\hat{w}_{qd}|| = 1 \), we multiply both sides of Eq. (23) by \( \hat{w}_q \) and obtain \( \alpha_q \) as
\[
\alpha_q = \frac{N}{2} E[g(\hat{w}_q \hat{y})] - \lambda E[\hat{w}_q \hat{\Lambda}_s(\hat{Y}) \hat{w}_q^T]. \tag{24}
\]
The second-order gradient is
\[
\frac{\partial^2 \mathcal{L}}{\partial \hat{w}_{qd}^2} = -NE(g(\hat{w}_q \hat{y}) \hat{y}^T) + 2 \lambda [\Lambda_s(\hat{Y})]_{dd} + 2 \alpha_q \tag{25}
\]
\[
\approx -NE(g(\hat{w}_q \hat{y})E[\hat{y}^T] + 2 \lambda [\Lambda_s(\hat{Y})]_{dd} + 2 \alpha_q \approx -NE(g(\hat{w}_q \hat{y}) + 2 \lambda [\Lambda_s(\hat{Y})]_{dd} + 2 \alpha_q. \tag{25}
\]
where \( E[g(\hat{w}_q \hat{y}) \hat{y}^T] \approx E[g(\hat{w}_q \hat{y})] \hat{y}^T \) is used. Finally, the fixed-point update rule for \( \hat{w}_{qd} \) reads
\[
\hat{w}_{qd} \leftarrow \hat{w}_{qd} - \frac{1}{-NE[g(\hat{w}_q \hat{y})] + 2 \alpha_q + 2 \lambda [\Lambda_s(\hat{Y})]_{dd}} \times [-NE(g(\hat{w}_q \hat{y}) \hat{y}^T) + 2 \lambda \hat{w}_q [\Lambda_s(\hat{Y})]_{dd} + 2 \alpha_q \hat{w}_{qd}] = \frac{NE(g(\hat{w}_q \hat{y}) \hat{y}^T) - \hat{w}_{qd} E[g(\hat{w}_q \hat{y})]}{-NE[g(\hat{w}_q \hat{y})] + 2 \alpha_q + 2 \lambda [\Lambda_s(\hat{Y})]_{dd}}. \tag{26}
\]
If we let \( g(S) \) and \( g'(S) \) denote the element-wise calculation of functions \( g(x) \) and \( g'(x) \) on sources \( S = \hat{W} \hat{Y} \), we have \( NE[g(\hat{w}_q \hat{y})] = g(S) \hat{y}_{qd} \) and \( 1 = [1, \cdots, 1]^T \) is an \( N \) dimensional vector. The update rule is simplified:
Table 1: Fast sICA by fixed-point algorithm.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>Set the number of independent components, and let ( Y ) be the matrix of sample data;</td>
</tr>
<tr>
<td>b)</td>
<td>Produce pre-whitened data ( Y_{pw} ) from ( Y ) by computing ( Y_{pw} = W^{T}Y );</td>
</tr>
<tr>
<td>c)</td>
<td>Calculate ( M_{y}(Y_{pw}) ) and diagonalize ( M_{y}(Y) = \Lambda_{y}(\tilde{Y})A^{T} ), where ( \tilde{Y} = A^{T}Y_{pw} ) represents the sample data in the subspace rotated by ( A );</td>
</tr>
<tr>
<td>d)</td>
<td>Initialize randomly ( \hat{W}^{(0)} );</td>
</tr>
<tr>
<td>e)</td>
<td>Set iteration time ( t = 0 ) and convergence threshold ( \epsilon ).</td>
</tr>
<tr>
<td>f)</td>
<td>Calculate the de-mixed source ( \hat{S}^{(t)} ) as ( \hat{S}^{(t)} = \hat{W}^{(t)}\tilde{Y} );</td>
</tr>
<tr>
<td>g)</td>
<td>Calculate ( g(\hat{S}^{(t)}) ), ( g(\hat{S}^{(0)}) ), ( K^{(0)} ), and ( \Phi^{(0)} );</td>
</tr>
<tr>
<td>h)</td>
<td>Update the de-mixing matrix ( \hat{W} ) according to the following rule: ( \hat{W}^{(t+1)} = \hat{W}^{(t)} \cdot [g(\hat{S}^{(t)})]^{T} - G^{T} Y^{(t)} );</td>
</tr>
<tr>
<td>i)</td>
<td>Do a symmetric orthogonalization on ( \hat{W} ) as ( \hat{W}^{(t+1)} = (\hat{W}^{(t+1)})^{T} T_{-1/2}^{(t+1)} \cdot \hat{W}^{(t+1)} );</td>
</tr>
<tr>
<td>j)</td>
<td>Check the convergence. If (</td>
</tr>
<tr>
<td>k)</td>
<td>Compute the overall sICA bases as ( W = \hat{W}^{T} \hat{W}^{pw} ).</td>
</tr>
</tbody>
</table>

\[
\hat{W} \leftarrow \Phi \cdot [g(\hat{S}^{(t)})^{T} - G^{T} Y],
\]

by defining \( G^{T} = \text{diag}[g(\hat{S}^{(0)})], q = 1, \ldots, Q \) and a \( Q \times D \) matrix \( \Phi \) with \( \Phi_{q} = [-G_{q} + 2\alpha_{q} + 2\lambda(\Lambda_{y}Y)_{q}]. \)
The symbol \( \circ \) denotes the component-wise multiplication of two matrices. The algorithm is given in Table 1. Finally, we have de-mixed source

\[
S = \hat{S} = \hat{W}^{T} \hat{Y} = \hat{W} A^{T} W^{pw} Y = WY
\]
de-mixing bases \( W = \hat{W} A^{T} W^{pw} \).

2.4 sICA vs. ICA on Generated Data

We made a simple experiment on some generated data to further explain the difference between sICA and ICA in feature extraction for pattern classification. In Fig. 4(a), we generate two data clusters from three independent super-gaussian components. To separate a component from the other two, the de-mixing axis should be perpendicular to the corresponding component, as shown in Fig. 4(a). If only 2 components are extracted, estimated de-mixing axes by ICA and sICA are different, as shown in Fig. 4(b). From the prewhitened data shown in Fig. 4(c), we found that the component, which demix axis 2 is perpendicular to, contributes more to the overall variance, and decides one axis in uncorrelation. Since de-mixing matrix becomes orthogonal after prewhitening [8], demix axis 2 is easy to emerge during the estimation of orthogonal ICA axes from the prewhitened data. In sICA, the inclusion of classification information eases the emergence of demix axis 1, which extracts an independent component with higher ability of class separation.

Obviously, the axes of sICA work better than those of ICA in classifying these two clusters.

Although facial data are much more complex than the above generated data and the number of independent component is unknown, we expect sICA to obtain more distinctive components in facial expression recognition than ICA.

3. Numerical Experiments and Discussions

We will focus on the comparison between sICA and ICA under the same conditions to investigate their effects by introducing the prior and by changing hyper-parameter \( \lambda \) for facial expression analysis.

3.1 Experimental Conditions

We have tested two databases: the Japanese Female Facial Expression (JAFFE) Database [20], which includes 213 images in total; and a subset of Cohn-Kanade AU-Coded Facial Expression Database with 396 images [21]. These images are aligned in face position and histogram-equalized. All images are resized to 64 \times 80 pixels. The goal of recognition is to classify them as a neutral face, or as one of the six elemental facial expressions suggested by Ekman et al. [22], i.e., happiness, anger, fear, disgust, sadness and surprise. Numerical experiments have been performed on 4 randomly selected training sets of various sizes \( N \) (70 and 100 for JAFFE Database, 100 and 220 for Cohn-Kanade Database). For each \( N \), a pair of two mutually exclusive sets was created, one with \( N \) images selected for training, and another
with all unselected images in that database for testing. In total, 4 pairs of training and testing datasets are used in our experiments. To alleviate the influence of initial values in the performance evaluation, we perform five-time computations starting from different random initialization for each combination of experimental conditions. Their average recognition rates are shown and compared in the present paper.

3.2 Recognition Rates in sICA

The first experiment is made to investigate the performance of ICA and sICA in facial expression recognition. For comparison, we have also investigated the performances of PCA and MDA (a multi-class extension of linear discriminant analysis) in our experiments. We plot recognition rate \( r_c \) on all four testing datasets as a function of \( Q \), the number of components, in Fig. 5.

Results from sICA are marked in up-triangle, while results from PCA, MDA, and classical ICA are shown by circles, solid lines and down-triangles, respectively. Results from the fourth pairs of datasets are shown in four subgraphs, respectively. Hyper-parameter \( \lambda \) is set to 0.35 for JAFFE Database and 0.25 for Cohn-Kanade Database. In Fig. 5, we show that:

1.) In our experiments of facial expression recognition, ICA outperforms PCA in all cases. As for the reason, we consider that facial expression is mainly caused by local variations, which are less significant in PCA bases than global variations, such as, pose variation, lighting, and personal differences, etc.

2.) We find that in a wide range of \( Q \), our sICA outperforms the classical ICA. Especially, our sICA significantly improves the robustness of performance with fewer components.

3.) Compared with MDA, sICA increases the recognition rate by 5 points on average for all testing datasets. We conclude that the improvement is not caused by deriving the same bases as those in MDA.

4.) Under a large number of independent components, we find comparable performance in both ICA and sICA, which proves that the inclusion of the selective prior does not deteriorate the separation ability of classical ICA. Therefore, we consider the improvement of performance under a small number of components by sICA as a result of the selective prior, which chooses the most distinctive components during the learning of de-mixing matrix. Later in this section, we will investigate the derived bases for further discussion.

Although in our experiments ICA outperforms PCA, we recognize that the comparison is not sufficient for evaluating their relative performance. PCA with other metrics may provide better results, and training conditions also affect the results. We do not intend to prove their relative performance in the present paper. Our experiments focus on the comparison between sICA and ICA. In sICA, we found that components with a large degree of separation can be obtained, even when only a small subset of components is derived. With this characteristic of sICA, it is possible to derive a small set of independent components that are distinctive in classification, which will significantly reduce the training time when dealing with a large training dataset. A smaller set of independent components also speeds up the running phase of recognition.

Since ICA is an nonlinear method for feature extraction, we also compared our sICA with two well-known nonlinear methods, nonlinear SVM and nonlinear MDA. Kernel techniques are used for the nonlinear representation of data in both methods. However, the thorough comparison is difficult, because the performance of these two methods varies according to many factors, such as, the type of kernel, the implementation of optima searching, the architecture of multi-class extension and so on. In our case, we adopted the Gaussian Radius Basis Function (RBF) kernel, which reads

\[
\phi(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2) \tag{29}
\]

for any two variables \( x_i, x_j \), and tested two multi-class SVMs, i.e., Max-Win (one vs. the rest) and Pairwise (one vs. one), to explain the advantage of using sICA. In Fig. 6, we plotted recognition rate \( r_c \) under different \( \gamma \) for all the four tested pairs of datasets. Since the hyper-parameter \( \gamma \) of RBF kernel does not exist in sICA, we use the best result achieved by sICA as a reference for performance evaluation. We conclude that:

1.) As hyper-parameter \( \lambda \) affects the performance of sICA, kernel hyper-parameter influences the recognition-rate of kernel-based methods much, whose automatic decision is still left as an unsolved problem.

2.) Due to the small size of database, the difference less than 2% means that only 3 images are different, which might be caused by other reasons, e.g., the normalization error. Therefore, we conclude that for Cohn-kanade database, sICA achieves comparable performances as nonlinear MDA and nonlinear SVM.

3.) Since sICA is an improvement of ICA in extract-
3.3 Characteristics of sICA

a) Behavior under Different Hyper-parameter $\lambda$

We plot recognition rate $r_c$ as a function of $\lambda$ for different numbers of components $Q$ for the four testing datasets with training sample sizes $N$ of 70, 100 for JAFFE Database and 100, 220 for Cohn-Kanade Database in four subgraphs of Fig. 7, respectively. By increasing $\lambda$, the robustness of recognition rate $r_c$ against a small number of components $Q$ improved. The experimental results for the transition of recognition rate $r_c$ with regard to hyper-parameter $\lambda$ appear in Fig. 8. These plots show that the recognition rates first increase with the increase of $\lambda$, and then decrease when $\lambda$ gets too large and causes a heavy bias on the sparseness of the obtained independent components. The tradeoff between sparseness and degree of discrimination should be taken to achieve the best results.

b) Degree of Separation in sICA

We compare the de-mixing components in PCA, MDA, ICA and sICA in Fig. 9 (a), (b), (c) and (d), respectively. For PCA and MDA, the topmost five components are shown. For ICA and sICA, we derive the bases by estimating the de-mixing matrix under $Q = 10$. In Fig. 9, we confirm that:

1.) Independent components in images are regarded as edge filters [23]. We also confirm that de-mixing components in ICA and sICA focus more on the local features of images.  

We believe that nonlinear SVM or MDA that focus on direct searching of optimal discrimination should achieve better results than sICA for unbiased training datasets. We argue that the proposed method is meaningful for the following reasons: due to the various factors that affect the facial appearance, most databases are not sufficient in scale and thus are biased for the representation of expressions; sICA is mainly proposed as a method for feature extraction rather than a classifier, which might achieve good results by combining with other classifiers; kernel tricks and other improvements have also been proposed to ICA, which are also applicable to sICA for better results. These will be studied in our future work.
such as edges or contours of faces, compared with those de-mixing components in PCA and MDA that represent holistic features of faces.

2.) We define degree of class separation \( d_{s}(w) = wM_{s}(Y)w^{T} \) for de-mixing component \( w \). In Fig. 10(a), all components derived by sICA are compared to those derived by ICA regarding the degree of class separation with regard to the classification of facial expression \( d_{s}(w) \). sICA achieves a higher degree of class separation than ICA for almost all components. A common feature of independent components of high \( d_{s}(w) \) and MDA bases is that they both emphasize some facial parts, e.g., eyebrows, cheeks, mouth corners, etc., that are thought to be essential in understanding facial expressions.

3.) In Fig. 10(b), we further compare their degrees of class separation with regard to the classification of individuals, i.e., \( d_{s}(w) \), between sICA and ICA. We found most components from sICA have lower \( d_{s}(w) \) than those from ICA.

Because ICA is an unsupervised method, the order of bases includes no information on the contribution of components to classification. In both \( d_{s}(w) \) and \( d_{i}(w) \), the components derived from classical ICA have rather flat significance. This might explain why the recognition rate of ICA is almost linearly proportional to the number of components in Fig. 5. The similar abilities of class separation of various bases confirm the necessity of including classification information.

A question left is the determination of optimal \( \alpha \). Regarding the selective prior as a constraint, components are affected by two factors, i.e., sparseness maximized by ICA and discrimination maximized by selective prior. Hyper-parameter \( \lambda \) controls their relative strength. Large \( \lambda \) increases the degree of separation, and reduces the sparseness. As a result, we note that bases in sICA include more global features than those in ICA. The trade-off between sparseness and degree of discrimination should be considered to achieve the best results by selecting proper \( \lambda \). We found that optimal values of \( \lambda \) are all approx. 0.35 and 0.25 for tested datasets from JAFFE and Cohn-Kanade Database, respectively. It is still not clear whether a constant \( \lambda \) exists according to the power relation between the sparseness term and the prior. The estimation of \( \lambda \) is possible if we add a bias to the selective prior to make it integrable. However, estimation of optimal bias introduces new hyper-parameters. Therefore, we leave the auto-selection of \( \lambda \) as an open question. For the current work, we could choose the best value from different trials of \( \lambda \) according to performances with a small number of components, which in fact does not require much computational cost.

c) Performance by Using Other Priors in sICA

Besides the prior defined in Eqs. (8)–(10) (Referred to as Prior 1), it is possible to define the selective prior as

\[
P_{w}(w) = \frac{1}{Z_{w}} \exp[-\lambda M_{w}(Y)w^{T}] \quad (30)
\]

\[
M_{w}(Y) = [M_{bc}(Y)]^{-\frac{1}{2}} M_{wc}(Y)[M_{bc}(Y)]^{-\frac{1}{2}} \quad (31)
\]

which is referred to as Prior 2. In Fig. 11, we compared these two priors in recognition accuracy. Prior 2 also shows the ability in improving the robustness of ICA against few numbers of components. A demerit of using Prior 2 is that...
large variations exist in the eigen-values of $M(Y)$, which causes problem in manual selection of $\lambda$ and becomes less robust than the case of using Prior 1. We use $M(Y) + \epsilon I$ instead of $M(Y)$ to solve the inverse problem, and set $\epsilon$ be half of non-zero minimal eigenvalue of $M(Y)$ to reduce the variations of $M(Y)$'s eigen values. It still failed to reach the same accuracy as that of Prior 1 due to the lack of an optimal $\lambda$. The advantage of using Prior 2 is that $Z$ could be computed because $\exp(-\lambda w M(Y) w^T)$ is integrable by using Gaussian integral, which provides possibilities for the auto-decision of $\lambda$. We have tried to estimate $\lambda$ by maximizing $\log p(W|Y)$ but found no high accuracy obtained by the optimal $\lambda$. Hence, we omit those results and leave the search for better criteria as a future work.

4. Conclusion and Future Work

In the present paper, we have proposed a supervised version of ICA for facial expression recognition by performing feature selection along with the learning of ICA. In order to decrease the computational burden of ICA, we try to obtain distinctive bases by only estimating a small subset of independent components. A selective prior defined on the scatter matrix has been introduced to the classical ICA, and MAP estimation is applied to derive the learning rule in a fixed-point algorithm. We performed numerical experiments to investigate the influence of new prior term. We also compared the results of sICA with the results of classical ICA. Our method shows better performance than ICA in that: a) recognition rate under a small number of independent components is largely improved, which is meaningful in dealing with a large training dataset, and b) sICA provides more robust performance than ICA in cases with few components. Experimental results on JAFFE and Cohn-Kanade Databases show that a high recognition rate of about 85% has been obtained by our method, when nearly half of all images are randomly chosen as training data. Further, sICA outperforms MDA by over 5 points on average, and also outperforms PCA in our experiments. There are still some problems left for us to study, such as the decision of optimal $\lambda$. Investigation on various priors is also a part of our future work.

References


Fan Chen  received the BS degree in computer science from Nanjing University in 2001. He received the MS degree in information science from Tohoku University in 2005. He is currently a PhD candidate in the School of Information Science at the Japan Advanced Institute of Science and Technology. His research interests include computer vision, image processing and pattern recognition.

Kazunori Kotani  received the BS and MS degrees in electrical and electronic systems engineering from Nagaoka University of Technology in 1981 and 1983. He was with the Consumer Products Research Center at Hitachi, Ltd. during 1983–1989. He received the PhD degree in information science and control engineering from the Nagaoka University of Technology in 1990. He was a research associate in the Department of Electrical Engineering, Nagaoka University of Technology, during 1990–1991. He is currently an associate professor in the School of Information Science at the Japan Advanced Institute of Science and Technology from 1991. His research interests include computer vision, facial image analysis and CG. He is a member of ITE, and J-FACE.