

A Study of Feature Extraction Using Supervised Independent Component Analysis

Seiichi Ozawa[†], Yoshinori Sakaguchi[†], Manabu Kotani[‡]

[†] Graduate School of Science and Technology, Kobe University, Kobe 657-8501, JAPAN

[‡] Faculty of Engineering, Kobe University, Kobe 657-8501, JAPAN
{ozawa, y-sakag }@eedept.kobe-u.ac.jp kotani@cs.kobe-u.ac.jp

Abstract

Recently, Independent Component Analysis (ICA) has been applied to not only problems of blind signal separation, but also feature extraction of images and sounds. However, it is not easy to obtain high-performance features from real data by using conventional ICA algorithms. This might be originated in the fact that class information is not taken into consideration when feature extraction is carried out. It is considered that a remedy for this problem is to introduce a supervisor into ICA. Hence, in this paper, we shall study the effectiveness of Umeyama's Supervised ICA (SICA) for feature extraction of handwritten characters. Two types of control vectors (supervisor) are proposed for SICA: (1) average patterns (Type-I) and (2) eigen-patterns (Type-II). To demonstrate the usefulness of SICA, recognition performance is evaluated for handwritten digits that are included in the MNIST database. From the results of recognition experiments, we certify that SICAs with both types of control vectors work effective for feature extraction. Actually, the within-class variance between-class variance ratio of SICA features with Type-I control vectors becomes slightly larger as compared with a conventional ICA.

1 Introduction

Recently, independent component analysis (ICA) has been widely known as a decorrelation technique based on high-order moment of input signals [1]. ICA has been so far applied to problems of blind signal separation such as sound/image separation and EEG signal separation. On the other hand, feature extraction of images and sounds has been also focused as one of prominent applications of ICA [2, 3, 4, 5]. Bartlett & Sejnowski extracted feature vectors from images of human faces using ICA, and showed that these feature vectors had greater viewpoint invariance for human faces as compared with Principal Component Analysis (PCA) ones [6]. Since

PCA decorrelates only the second order statistics of input signals, this result indicates that higher-order features are useful for capturing invariant features of face patterns as well as the second-order features. Such invariant characteristics of features extracted with ICA might be attractive for other pattern recognition problems.

In our previous works [7], we have presented two types of feature selection based on the cumulative proportion of eigenvalues and kurtosis. The former selection is carried out for principal components (PCs) of inputs and the latter is done for independent components (ICs). In the recognition experiments, we have shown that a hybrid method, in which feature selection was carried out for ICs as well as for PCs, had interesting characteristics when low-dimensional feature vectors were used in recognition. However, the recognition performance was not high from the practical point of view. It might be originated in the fact that class information is not taken into consideration when feature extraction is carried out.

Recently, Umeyama has proposed supervised ICA (SICA) [8], in which class information can be considered in the learning of a separation matrix. To overcome the above problem, we shall study the effectiveness of Umeyama's SICA for feature extraction of handwritten characters.

2 Independent Component Analysis (ICA)

2.1 A Conventional ICA

Several ICA algorithms have been proposed so far, which are different in objective functions (or contrast functions) for statistical independence and how to derive ICA algorithms [1, 9, 10]. In general, estimated independent components obtained by these algorithms are different each other. However, it is difficult to discuss which algorithms are most appropriate for feature extraction. Therefore, we are not concerned here with the

adequacy for ICA algorithms. In the followings, we shall adopt the bigradient algorithm proposed by Karhunen and Oja [11] because supervised ICA adopted here is a extended version of this algorithm.

Suppose that we observe a m -dimensional zero-mean input signal at time t , $\mathbf{v}(t) = \{v_1, \dots, v_m\}'$, where $'$ means the transposition of matrices and vectors. Then the n -dimensional whitening signal, $\mathbf{x}(t)$, is given by the following equation:

$$\mathbf{x}(t) = \mathbf{M}\mathbf{v}(t) = \mathbf{D}^{-1/2}\mathbf{E}'\mathbf{v}(t), \quad (1)$$

where \mathbf{M} means a $n \times m$ ($n \leq m$) whitening matrix that is given by a matrix of eigenvalues, \mathbf{D} , and a matrix of eigenvectors, \mathbf{E} . Here, assume that $\mathbf{v}(t)$ is composed of n statistically independent components (ICs), $\mathbf{s}(t) = \{s_1(t), \dots, s_n(t)\}'$. Then, the following linear transformation from $\mathbf{x}(t)$ to $\mathbf{s}(t)$ exists:

$$\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t). \quad (2)$$

$\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}'$ is often called a separation matrix, and it can be obtained through the training of a two-layer feedforward neural network. This neural network has n outputs denoted as $\tilde{\mathbf{s}}(t) = \{\tilde{s}_1(t), \dots, \tilde{s}_n(t)\}'$ and the i th row vector, \mathbf{w}_i' ($i = 1, \dots, n$), of \mathbf{W} corresponds to a weight vector from inputs to the i th output, \tilde{s}_i .

The term ‘independent’ is used here according to the following definition in statistics:

$$p[s_1(t), \dots, s_n(t)] = \prod_{i=1}^n p_i[s_i(t)], \quad (3)$$

where $p[\cdot]$ is a probability density function. Since the above probability density function is not preliminary unknown, suitable objective functions should be devised such that neural outputs, \tilde{s}_i , are satisfied with Eq. (3) as much as possible, i.e. $\tilde{\mathbf{s}}(t) \simeq \mathbf{s}(t)$. Karhunen and Oja have proposed the following contrast function [11], $J(\cdot)$, to be maximized in terms of output signals, $\tilde{\mathbf{s}}$:

$$J(\tilde{\mathbf{s}}) = \sum_{i=1}^n |E[\tilde{s}_i^4] - 3\{E[\tilde{s}_i^2]\}^2|, \quad (4)$$

where $E[\cdot]$ means expectation. As well known, Eq. (4) corresponds to the fourth-order cumulants of $\tilde{s}_i(t)$, called *kurtosis*. The following learning algorithms for a separation matrix, \mathbf{W} , are derived from the gradient of Eq. (4) and the orthonormality constraints of \mathbf{W} [12]:

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k + \mu(\tanh \tilde{\mathbf{s}}_k)\mathbf{x}'_k \\ &+ \gamma(\mathbf{I} - \mathbf{W}_k\mathbf{W}'_k)\mathbf{W}_k, \end{aligned} \quad (5)$$

where k means time step.

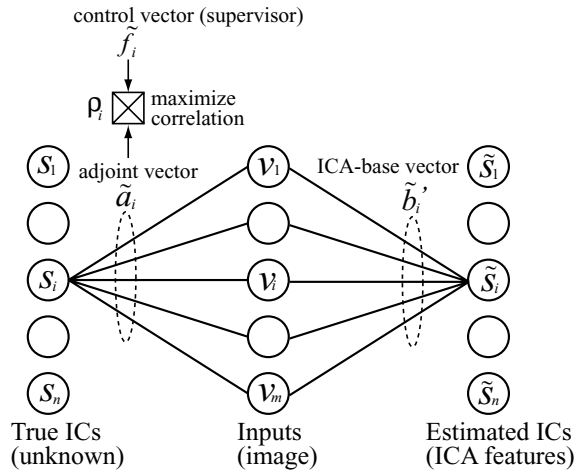


Figure 1: Schematic diagram of information processing in SICA.

2.2 Supervised ICA

Umeyama has proposed a supervised version of ICA (SICA), in which a separation matrix is trained such that the contributions of ICs to input patterns could be controlled by *supervisor*. In other words, the training of SICA is carried out by maximizing correlations between each IC and specific sets of inputs as well as by maximizing independency of ICs.

Let us describe the details of SICA. From Eqs. (1) and (2), the relation between inputs and estimated ICs is rewritten as follows:

$$\mathbf{v} = (\mathbf{E}\mathbf{D}^{1/2}\tilde{\mathbf{W}}^{-1})\tilde{\mathbf{s}} = (\mathbf{E}\mathbf{D}^{1/2}\tilde{\mathbf{W}}')\tilde{\mathbf{s}} = \tilde{\mathbf{A}}\tilde{\mathbf{s}}, \quad (6)$$

where we should note that \mathbf{W} is an orthogonal matrix. Here, $\tilde{\mathbf{A}}$ corresponds to an estimated mixture matrix. The i th column vector, $\tilde{\mathbf{a}}_i$ ($i = 1, \dots, n$), of $\tilde{\mathbf{A}}$ is called an *adjoint vector* whose element values mean the contribution of the i th IC, \tilde{s}_i , to an input pattern, \mathbf{v} (see Fig. 1). Therefore, if we want to control the contributions of the i th IC, we should give desired signals to these adjoint vectors. In SICA, as shown in Fig. 1, a normalized control vector, $\mathbf{f}_i = \{f_{i1}, \dots, f_{im}\}'$, is given to $\tilde{\mathbf{a}}_i$ as its desired signal, and the following correlation, ρ_i , between \mathbf{f}_i and $\tilde{\mathbf{a}}_i$ is maximized:

$$\rho_i = \frac{\mathbf{f}_i' \tilde{\mathbf{a}}_i}{\|\tilde{\mathbf{a}}_i\|}. \quad (7)$$

The update rule of a separation matrix, \mathbf{W} , at time k is shown as follows:

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k + \mu(\tanh \tilde{\mathbf{s}}_k)\mathbf{x}'_k \\ &+ \gamma(\mathbf{I} - \mathbf{W}_k\mathbf{W}'_k)\mathbf{W}_k + \mathbf{A}\mathbf{G}, \end{aligned} \quad (8)$$

where

$$\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_p, 0, \dots, 0]'. \quad (9)$$

Here, p is the number of ICs to be controlled. In Eq. (9), \mathbf{g}_i is obtained from the derivative of ρ_i with \mathbf{W} in Eq. (7), and \mathbf{A} is a matrix of learning coefficients shown below:

$$\mathbf{A} = \text{diag}(\lambda_1, \dots, \lambda_p, 0, \dots, 0), \quad (10)$$

where λ_i is given by

$$\lambda_i = \lambda'_\mu \frac{\|\tanh \tilde{s}_i \mathbf{x}\|}{\|\mathbf{g}_i\|}. \quad (11)$$

Here, λ'_μ is a negative constant that determines the balance between the independence term (the second term) and the correlation term (the fourth term) in the right hand side of Eq. (8) (see [8] for details).

3 Feature Extraction of Handwritten Characters

3.1 Feature Extraction Using ICA

As described in Section 2, ICA algorithms allow us to decompose input signals into their independent components such that they are satisfied with Eq. (3) as much as possible. Such characteristics of ICA can be applied to feature extraction of hand-written characters.

Based on Eqs. (1) and (2), the relation between inputs and outputs of ICA is given by

$$\tilde{\mathbf{s}}(t) = \tilde{\mathbf{W}}\mathbf{D}^{-1/2}\mathbf{E}'\mathbf{v}(t) = \tilde{\mathbf{B}}\mathbf{v}(t), \quad (12)$$

where $\tilde{\mathbf{W}}$ is a separation matrix trained by an ICA algorithm and $\tilde{\mathbf{B}} = \tilde{\mathbf{W}}\mathbf{D}^{-1/2}\mathbf{E}'$ is a $n \times m$ matrix. When an input, $\mathbf{v}(t)$, corresponds to the t th presentation of character images, the ICA output, $\tilde{\mathbf{s}}(t)$, can be considered as its feature vector (see Fig. 1). Here, the i th row vector, $\tilde{\mathbf{b}}'_i$ ($i = 1, \dots, n$), of $\tilde{\mathbf{B}}$ corresponds to a base vector spanning n -dimensional feature space (such base vectors are called ICA-bases for convenience). Since $\mathbf{E}'\mathbf{v}(t)$ corresponds to principal components (PCs) of $\mathbf{v}(t)$, one can say that an ICA feature vector is given by the transformation $\tilde{\mathbf{W}}\mathbf{D}^{-1/2}$ of a PCA feature vector*. Therefore, we can consider that the process of feature extraction using ICA consists of two types of transformations (see Fig. 2). One is the transformation from an input image to PCA features, and the other is the transformation from PCA features to ICA features.

Different feature selection (dimension reduction) can be applied to outputs of the above transformations; that is,

*For notational convenience, we denote feature vectors extracted with ICA and PCA as ICA features and PCA features, respectively.

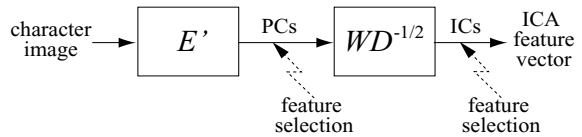


Figure 2: A block diagram of feature extraction of characters using ICA.

we can reduce dimensions of PCA features and/or ICA features. In our previous work [7], however, it is not easy to extract useful features by reducing dimensions of ICA features. Therefore, feature selection is carried out only for PCA features here.

Cumulative proportion has been often used in feature selection for PCA features as a criterion of determining useful features. For convenience, eigenvalues of a covariance matrix of training samples are denoted in order of their magnitude: $\lambda_1 \geq \dots \geq \lambda_m$. Then, the cumulative proportion, c_n , is defined as follows:

$$c_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^m \lambda_i}, \quad (13)$$

where n is the number of large eigenvalues to be selected. Let us introduce an upper bound of cumulative proportion, c_0 , that gives a threshold value of determining what feature vectors should be adopted, then the largest value of n can be determined such that $c_n \leq c_0$ holds. We select n eigenvectors with the largest n eigenvalues as PCA-bases; that is, we consider a n -dimensional subspace spanned by eigenvectors with $\lambda_1, \dots, \lambda_n$. After this feature selection is carried out, a n -dimensional vector of PCs, $\mathbf{E}'\mathbf{v}(t)$, is obtained, then a n -dimensional ICA feature vector, $\tilde{\mathbf{s}}(t)$, is calculated from Eq. (12).

3.2 Feature Extraction Using Supervised ICA

In pattern recognition problems, it is more desirable that extracted features belonging to different classes are mutually separated as much as possible in the feature space. Conventional ICAs are, however, categorized in unsupervised learning; therefore, good separability for extracted features is not always ensured. To overcome this problem, we should utilize class information (teacher signals) for extracting useful features. Hence, supervised ICA (SICA) shown in the previous section is adopted here for feature extraction of handwritten characters.

As stated in Section 2, an adjoint vector, $\tilde{\mathbf{a}}_i$ ($i = 1, \dots, n$), in SICA indicates the contribution of the i th IC, \tilde{s}_i , to an input pattern (e.g. character image), \mathbf{v} . Through the learning of SICA, this contribution can be

controlled by varying a control vector, f_i . However, it is not clear how these control vectors should be designed in order to extract good independent features. As seen from Eq. (6), one can say that an input pattern is given by weighted sum of some adjoint vectors. Therefore, control vectors should be designed such that input patterns are approximated by weighted sum of the control vectors (note that adjoint vectors are trained so as to maximize the correlation with control vectors).

Hence, we shall present two types of control vectors whose two-dimensional representation corresponds to (1) average patterns (Type-I) and (2) eigen-patterns (Type-II). As shown in Fig. 3, Type-I control vectors are obtained by simply averaging training samples in each category. On the other hand, Type-II control vectors are obtained from eigenvectors for a covariance matrix of training data in each class. Fig. 4 shows examples of Type-II control vectors that are given by eigenvectors with the largest two eigenvalues. The above two types of control vectors are normalized such that their average and variance are equal to 0 and 1, respectively.

4 Simulations

To investigate the usefulness of SICA, the recognition performance is evaluated for handwritten digits. Six thousands of digit patterns in the MNIST database are used for training, and ten thousands of digit patterns are used for evaluation. Although the MNIST database originally includes 60,000 training samples, we use only 6,000 samples (600 samples for each digit class) in order to reduce learning time of ICA and SICA. Each image of handwritten digits is composed of 28x28 pixels and no preprocessing is carried out before feature extraction. Training samples are used for generating reference vectors (prototypes) as well as learning ICA-bases. The numbers of prototypes, P , for each class are set to the following values: 1, 250, 500, and 600. In the case of $P = 600$, all training samples are used as prototypes without any modifications. In the other cases, prototypes are produced based on an agglomerative hierarchical clustering [13] from training samples, in which direction cosine is used for a similarity measure. In classification, we also adopt direction cosine as a similarity measure between input images and prototypes. After calculating similarities with all prototypes, recognition is conducted based on the nearest neighbor rule.

In SICA, the number control vectors, n , are set to the following values: 10, 20, 30, 40, 50. Since the number of classes is 10, the number of control vectors for each class is $n/10$: 1, 2, 3, 4, 5. For each class, $n/10$ of

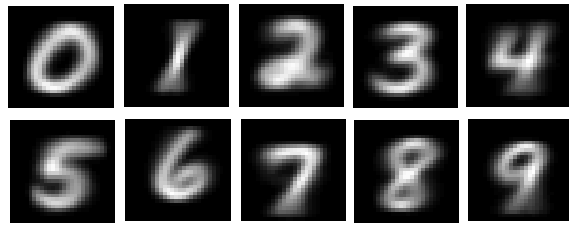


Figure 3: Type-I control vectors.

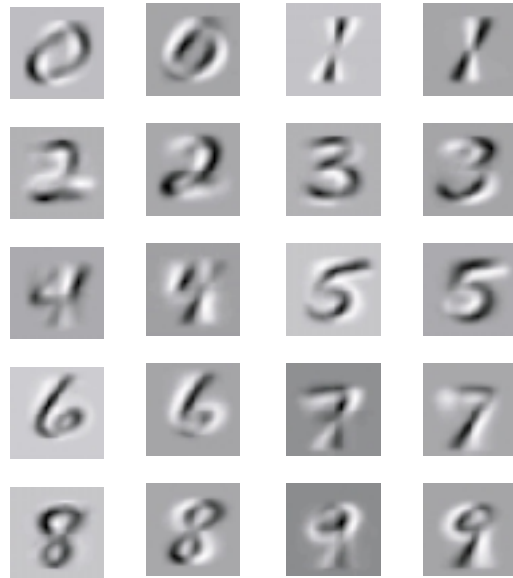


Figure 4: Examples of Type-II control vectors. ($n=20$)

Type-I control vectors are given by the same average pattern shown in Fig. 3. On the other hand, Type-II control vectors for each class correspond to eigenvectors with the largest $n/10$ eigenvalues for a covariance matrix of training samples. Note that the number of control vectors is equal to the dimensions of feature vectors, n . The dimension reduction of feature vectors is carried out by the sphering of input images.

For comparative purposes, we evaluate the usefulness of ICA features obtained by a conventional ICA. Tables 1(a)-(c) show the results of recognition accuracy for ICA features and those for features extracted by SICA with Type-I / Type-II control vectors[†], respectively. As you can see from Tables 1(a)-(c), the differences in the performances of ICA and SICA are not so much distinct.

[†]Feature vectors extracted by SICA with Type-I / Type-II control vectors are respectively noted as 'SICA (Type-I) features' and 'SICA (Type-II) features' for short.

Table 1: Recognition accuracy [%] for ICA features, SICA (Type-I) features, and SICA (Type-II) features. n means the dimensions of feature vectors and P corresponds to the number of prototypes.

(a) Conventional ICA

| | $P=1$ | $P=250$ | $P=500$ | $P=600$ |
|----------|-------|---------|---------|---------|
| $n = 10$ | 75.14 | 89.60 | 94.91 | 95.20 |
| $n = 20$ | 81.01 | 93.71 | 97.17 | 97.24 |
| $n = 30$ | 82.96 | 93.36 | 97.24 | 97.35 |
| $n = 40$ | 84.11 | 92.91 | 97.08 | 97.13 |
| $n = 50$ | 84.60 | 92.35 | 96.87 | 97.05 |

(b) SICA with Type-I control vectors

| | $P=1$ | $P=250$ | $P=500$ | $P=600$ |
|----------|-------|---------|---------|---------|
| $n = 10$ | 75.14 | 89.59 | 94.90 | 95.19 |
| $n = 20$ | 81.00 | 93.73 | 97.18 | 97.25 |
| $n = 30$ | 83.00 | 93.36 | 97.25 | 97.34 |
| $n = 40$ | 84.09 | 92.91 | 97.09 | 97.14 |
| $n = 50$ | 84.61 | 92.33 | 96.90 | 97.07 |

(c) SICA with Type-II control vectors

| | $P=1$ | $P=250$ | $P=500$ | $P=600$ |
|----------|-------|---------|---------|---------|
| $n = 10$ | 75.06 | 89.49 | 94.90 | 95.22 |
| $n = 20$ | 81.10 | 93.79 | 97.15 | 97.24 |
| $n = 30$ | 83.01 | 93.46 | 97.30 | 97.37 |
| $n = 40$ | 84.19 | 92.92 | 97.07 | 97.15 |
| $n = 50$ | 84.63 | 92.36 | 96.90 | 97.06 |

tive even though the dimensions of feature vectors and the number of prototypes are varied. However, when $P=500$, the performances of SICA with Type-I control vectors are slightly better than those of ICA. The maximum performance is realized by SICA with Type-II control vectors when $P=600$ and $n=30$.

Table 2 shows the within-class variance between-class variance ratio of ICA features extracted from test samples. This ratio is calculated as follows:

$$\sigma_W^2 = \frac{1}{n} \sum_{i=1}^c \sum_{\mathbf{v} \in \mathcal{V}_i} (\mathbf{v} - \mathbf{m}_i)' (\mathbf{v} - \mathbf{m}_i) \quad (14)$$

$$\sigma_B^2 = \frac{1}{n} \sum_{i=1}^c n_i (\mathbf{m}_i - \mathbf{m})' (\mathbf{m}_i - \mathbf{m}) \quad (15)$$

$$J_\sigma = \sigma_B^2 / \sigma_W^2, \quad (16)$$

where \mathcal{V}_i means a set of class i test samples and c is the number of classes. \mathbf{m} is a mean vector of all test samples, \mathbf{v} , and \mathbf{m}_i is a mean vector of class i test samples. As seen in Table 2, the within-class variance between-class variance ratio for SICA (Type-I) features

Table 2: The within-class variance between-class variance ratio, J_σ , of features extracted from 10,000 test samples.

| | ICA | Type-I | Type-II |
|----------|--------|--------|---------|
| $n = 10$ | 0.4959 | 0.4961 | 0.4898 |
| $n = 20$ | 0.2611 | 0.2611 | 0.2606 |
| $n = 30$ | 0.1754 | 0.1755 | 0.1745 |
| $n = 40$ | 0.1315 | 0.1317 | 0.1314 |
| $n = 50$ | 0.1052 | 0.1053 | 0.1050 |

Table 3: Kurtosis of extracted features for 6,000 training samples.

| | ICA | Type-I | Type-II |
|----------|-------|--------|---------|
| $n = 10$ | 1.476 | 0.810 | 0.3129 |
| $n = 20$ | 2.382 | 1.1821 | 0.7978 |
| $n = 30$ | 3.084 | 1.8257 | 1.9823 |
| $n = 40$ | 3.706 | 2.5547 | 2.3387 |
| $n = 50$ | 4.051 | 3.2734 | 2.863 |

is higher than that for ICA features; this means that good features are extracted from handwritten characters by SICA with Type-I control vectors. However, by contrast, the ratio for SICA (Type-II) features becomes lower.

Finally, we evaluate kurtosis of features extracted from training samples, $\tilde{\mathbf{s}} = \{\tilde{s}_1, \dots, \tilde{s}_n\}$, in order to examine their independency. The following absolute value of kurtosis is evaluated here:

$$\text{kurt}(\tilde{\mathbf{s}}) = \frac{1}{n} \sum_{i=1}^n \left| \frac{E[\tilde{s}_i^4]}{E[\tilde{s}_i^2]^2} - 3 \right|. \quad (17)$$

If $\tilde{\mathbf{s}}$ has larger kurtosis, one can say that this feature vector is more statistically independent. Table 3 shows the results for ICA features, SICA (Type-I) features, and SICA (Type-II) features. As we might expect, the independency of ICA features is the highest. The reason why the kurtosis of SICA features becomes small is that the maximization of correlation between independent features and control vectors (supervisor) as well as the maximization of independency is carried out.

From the above results, the characteristics of ICA and SICA features are summarized as follows:

1. From recognition experiments and the study of within-class variance between-class variance ratio, the separability of SICA (Type-I) features is slightly better than that of ICA features.

2. Kurtosis of SICA features is smaller than that of ICA features. Considering the previous fact, one can say that features whose independency is simply increased are not always effective. In other words, this fact suggests that increasing independency and introducing supervisor into ICA is a key to enhancing the performance of feature extraction.

5 Conclusions

We applied Supervised Independent Component Analysis (SICA) to feature extraction of handwritten digits. Two types of control vectors were introduced into SICA in order to examine the effectiveness of supervisor. From the results of recognition experiments, we certified that not only increasing independency but also introducing supervisor into ICA could realize high-performance feature extraction. Although both types of control vectors presented here worked effectively to extract useful features of handwritten digits, there are slight differences in within-class variance between-class variance ratio. This suggests that the appropriateness in design of control vectors can reflect the separability of extracted features. The appropriate design of control vectors is one of the most important issues in this work. This is left as our future work.

References

- [1] A. Hyvärinen: "Survey on independent component analysis", *Neural Computing Surveys*, **2**, 94-128, 1999.
- [2] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja: "Applications of neural blind separation to signal and image processing", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 131-134, 1997.
- [3] M. Kotani, Y. Shirata, S. Maekawa, S. Ozawa, and K. Akazawa: "Application of independent component analysis to feature extraction of speech", *Proc. of Int. Joint Conf. on Neural Networks (IJCNN99-Washington DC)*, CD-ROM #704, 1999.
- [4] S. Ozawa, T. Tsujimoto, M. Kotani, and N. Baba: "Application of independent component analysis to hand-written Japanese character recognition", *Proc. of International Joint Conf. on Neural Networks (IJCNN99-Washington DC)*, CD-ROM #462, 1999.
- [5] Y. Watanabe, M. Hirahara, and T. Nagano: "Feature extraction of palm prints using supervised independent component analysis", *CD-ROM Proc. of 7th Int. Conf. on Neural Info. Processing*, 2000.

- [6] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski: "Independent component representations for face recognition", *Proc. of the SPIE*, **3299**, 528-539, 1997.
- [7] S. Ozawa, M. Kotani: "A study of feature extraction and selection using independent component analysis", *Proc. of 7th Int. Conf. on Neural Info. Processing*, **I**, 369-374, 2000.
- [8] S. Umeyama, S. Akaho, Y. Sugase: "Supervised independent component analysis and its applications to face image analysis" (in Japanese), *Tech. Report of IE-ICE*, **NC99-2**, 9-16, 1999.
- [9] A. J. Bell and T. J. Sejnowski: "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, **7**, 1129-1159, 1995.
- [10] S. Amari, A. Chichocki, and H. Yang: "A new learning algorithm for blind signal separation", *Advances in Neural Information Processing Systems*, **8**, MIT Press, Cambridge, MA, 757-763, 1996.
- [11] J. Karhunen and E. Oja: "A class of neural networks for independent component analysis", *IEEE Trans. on Neural Networks*, **8**, 3, 486-503, 1997.
- [12] L. Wang, J. Karhunen, and E. Oja: "A bigradient optimization approach for robust PCA, MCA, and source separation", *Proc. IEEE Int. Conf. on Neural Networks (ICANN95-Perth)*, 1684-1689, 1995.
- [13] R. O. Duda and P. E. Hart: *Pattern classification and scene analysis*, John Wiley & Sons, 1973.