Application of Independent Component Analysis to Hand-written Japanese Character Recognition

Seiichi Ozawa †, Toshihide Tsujimoto †, Manabu Kotani ††, and Norio Baba †
Email:{ozawa, baba}@is.osaka-kyoiku.ac.jp, kotani@cs.kobe-u.ac.jp

†Dept. of Information Science, Osaka Kyoiku University, Kashiwara, Osaka 582-8582, Japan
††Faculty of Engineering, Kobe University, Nada-Ku, Kobe 657-8510, Japan

Abstract

We explore an approach to recognizing Japanese Hiragana characters utilizing independent components of input images (we call this method ICA-matching). These components are extracted by Fast ICA algorithm proposed by Hyvärinen and Oja. We propose several formats of inputs, which are different in how a character image is transformed into time sequences. From recognition experiments, we show that ICA-matching outperforms conventional methods in some cases. However, in order to realize high performance, we have to pay attention to the following parameters: dimensions of feature vectors and rate of noise added to training data. In discussions, we try to study how these parameters are related to the performance of ICA-matching.

1. Introduction

Independent component analysis (ICA) has been developed as a decorrelation technique for high-order moment of input signals[1][2][3]. Using such characteristics, ICA has been so far applied to problems of blind signal separation such as sound/image separation and EEG signal separation. Recently, feature extraction and pattern recognition have been also focused as one of prominent applications of ICA. For example, Bartlett & Sejnowski showed that feature vectors extracted by ICA had greater viewpoint invariance for human faces as compared with PCA (principal component analysis) ones[4]. Since PCA feature vectors decorrelate only the second order statistics, this result means that higher-order features as well as second order ones are important for high-performance pattern recognition in specific problem domains.

In this paper, we will study the potential of ICA in pattern recognition tasks, especially for hand-written Japanese Hiragana characters. In Section 2, we will give a brief explanation of our adopted ICA algorithm. In Section 3, we will propose an approach to pattern recognition utilizing ICA feature vectors, then we will show the experimental results in the next section. In Section 5, we will state two key points to obtain high performance in this approach.

2. Independent Component Analysis

Several ICA algorithms have been proposed so far that are different in objective functions (or contrast functions) to obtain separation matrices. Therefore, estimated independent components are usually different depending on adopted ICA algorithms. However, at current state, it is difficult to discuss which algorithms are most appropriate for feature extraction. Considering convergence speed and usability, we adopt Fast ICA algorithm proposed by Hyvärinen and Oja[5].

Suppose that we observe a m-dimensional input signal at time t, $\boldsymbol{v}(t) = \{v_1, \cdots, v_m\}^T$, where T means the transposition of matrices and vectors. Then the n-dimensional whitened signal, $\boldsymbol{x}(t)$, is given by the following equation:

$$\boldsymbol{x}(t) = \boldsymbol{M}\boldsymbol{v}(t), \tag{1}$$

where $\boldsymbol{M}$ means a $n \times m$ whitening matrix. Here, we assume that $\boldsymbol{x}(t)$ is composed of n statistically independent signals, $\boldsymbol{s}(t) = \{s_1(t), \cdots, s_n(t)\}$, and they are obtained by the following linear transformation:

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

$\boldsymbol{W}$ is often called separating matrix and it can be trained by a two-layer feedforward neural network with n outputs. In this case, the ith column vector, $\boldsymbol{w}_i^T (i = 1, \cdots, n)$, of $\boldsymbol{W}$ corresponds to a weight vector from inputs to the ith output.

To obtain a separating matrix, Hyvärinen has proposed the following objective function, $J(\cdot)$, to be maximized or minimized in terms of output signals, $s_i = \boldsymbol{w}_i^T \boldsymbol{x}$:

$$J(\boldsymbol{w}_i) = E\{(\boldsymbol{w}_i^T \boldsymbol{x})^4\} - 3[E\{(\boldsymbol{w}_i^T \boldsymbol{x})^2\}]^2 + F(\| \boldsymbol{w}_i \|^2), \tag{3}$$

where $E\{\cdot\}$ means expectation. The first two terms in the right-hand side of Eq.(3) correspond to fourth-order statistics of $s_i(t)$, called *kurtosis*. The third term, $F(\cdot)$, is a penalty function so as to limit $\| \boldsymbol{w}_i \|$ to a constant value (see [5] for details). The learning algorithm of the ith weight vector is derived from the gradient of Eq.(3) with respect to $\boldsymbol{w}_i$ as follows:

$$\begin{aligned} \boldsymbol{w}_i(t+1) &= \boldsymbol{w}_i(t) \pm \mu(t)[\boldsymbol{x}(t)(\boldsymbol{w}_i(t)^T \boldsymbol{x}(t))^3 \\ &\quad -3 \| \boldsymbol{w}_i(t) \|^2 \boldsymbol{w}_i(t) + f(\| \boldsymbol{w}_i(t) \|^2)\boldsymbol{w}_i(t)], \end{aligned} \quad (4)$$

where f is the derivative of $F/2$. Hyvärinen & Oja have proposed an on-line (or batch) algorithm to obtain fixed points of Eq.(4), called Fast ICA algorithm.

3. Pattern Recognition

Let us define the following $n \times m$ matrix, $\boldsymbol{B}$:

$$\boldsymbol{B} = \boldsymbol{W}\boldsymbol{M}. \quad (5)$$

As you can see from Eqs.(1)(2), $\boldsymbol{B}$ represents the relation between inputs (pixel data) and outputs (independent components, ICs). In other words, $\boldsymbol{B}$ corresponds to a transformation matrix from input space to another feature space. Hence, the ith column vector, $\boldsymbol{b}_i^T$ $(i = 1, \cdots, n)$, of $\boldsymbol{B}$ can be considered as a base vector spanning n-dimensional feature space (this base vector is called ICA-base for convenience). Note that $\boldsymbol{b}_i^T$s are not orthogonal each other in many cases. The ith output, $s_i(t)$, is a projection of inputs to the ith ICA-base; hence we shall use $\boldsymbol{s}(t)$ as a feature vector.

Here, we adopt ETL-4 database sets of hand-written Japanese Hiragana characters in recognition tasks. Each data set consists of 46 Hiragana characters, and each character image is composed of 76×72 binary values (PBM format). These images are preprocessed through centralization and size normalization before they are converted to 15×15 16-bit gray-scale images. When an input image is presented to ICA algorithm, several formats of inputs can be considered which are different in how a character image is transformed into time sequences. Let us consider the following input formats.

[Type-I] The whole pixels of a character image is simultaneously presented as an input, $\boldsymbol{x}(t)$. In this case, the dimensions of an input vector are 225 and the length of an input sequence is 46 for each data set.

[Type-II] A subimage is extracted by imposing $(S \times S)$-pixel window on an original 15×15 character image. This subimage is used as an input at time t, $\boldsymbol{x}(t)$. The window is shifted by S pixels to the right, then next input vector $\boldsymbol{x}(t+1)$ is extracted from the window. When the window reaches the end of the line, it goes down S pixels and scans from the leftmost. Such a scan procedure is continued until the window reaches the right bottom corner of a character image. This format of ICA inputs is represented by a sequence of $46 \times (15/S)^2$ $(S \times S)$-dimensional vectors.

[Type-III] The scan procedure to generate input sequences is the same as Type-II format except that the frame window moves either leftward or downward by half size of a frame window (i.e. $S/2$ pixels). This format of ICA inputs is represented by a sequence of $46 \times \{(15 - S/2)/(S/2)\}^2$ $(S \times S)$-dimensional vectors.

The data sets of twenty different testees are used for recognition: ten data sets are used for reference patterns and the other ten sets are used for test patterns. Training patterns are generated from reference patterns by adding random noise to them. Ten different random series are added to each of reference patterns, hence the number of training patterns is ten times of reference patterns (except for a noiseless case). Using these training patterns, ICs and ICA-bases are calculated based on Fast ICA algorithm. After training, ICs for reference patterns are calculated by using the obtained ICA-bases. These ICs are used as reference vectors in pattern matching. ICs for test patterns are also calculated, and the most matching reference vector for a test vector is searched based on the conventional similarity matching. We call this method ICA-matching.

4. Experimental Results

Figures 1(a)-(c) respectively show the recognition results for Type-I, II, III data sets. Horizontal axis means the rate of noise added to training patterns, σ, and vertical axis means the rate of correct recognition. The recognition accuracy is calculated by averaging performances for five test pattern sets. Three lines in Figs.1(b)(c) correspond to the results for different sizes of frame window, S. To show the usefulness of our approach, we also examine the recognition performances of two conventional methods: one is the similarity matching between pixel data of reference and test patterns, and the other is the similarity matching between feature vectors that are given by principal component analysis (PCA). For convenience, the former method is called pixel-matching and the latter is called PCA-matching. The recognition rate for pixel-matching is 82% and that for PCA-matching is shown by dotted lines in Figs.1(a)-(c).

In Fig.1(a), one can say that the performance for Type-I training data greatly depends on the rate of adding noise and they are lower than both pixel-matching and PCA-matching except for $\sigma = 0.4$. Especially, in case of $\sigma = 0.0$, the performance is crucially poor. On the other hand, the performances for Type-II and Type-III

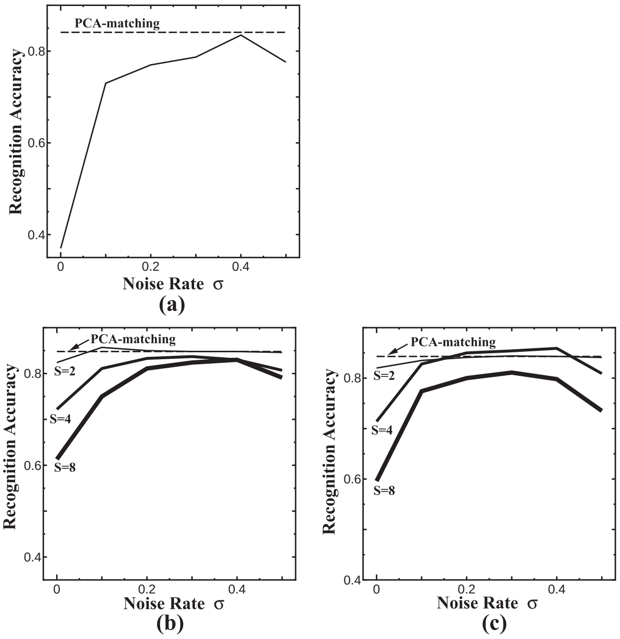

Figure 1: Recognition results for (a)Type-I (b)Type-II (c)Type-III training data.

training data are fairly good except for $\sigma = 0.0$. Although performance dependency on σ still remains, one can say that ICA-matching outperforms PCA-matching as well as pixel-matching in some cases.

From the results in Figs.1(a)-(c), characteristics of ICA-matching are summarized as follows:

1. When training data are not contaminated by random noise (i.e. training patterns are equal to reference patterns), the performance of ICA-matching is degraded as compared with other noise-added cases. Furthermore, it is inferior to the performances of both pixel-matching and PCA-matching.
2. In general, when window size is small (i.e. the dimensions of ICs are small), the performance of ICA-matching tends to be high. This suggests that feature vectors should be represented in small dimensional space. However, small size of windows causes the increase in the length of input sequences. Considering the computational costs, $S = 4$ is preferred in our experiments.
3. When window size is large, the performance is largely affected by the rate of adding noise, σ. Roughly speaking, σ=0.3~0.4 is preferred for all S.
4. Obtaining ICA-bases from Type-I data sets is unsuitable for high-performance ICA-matching. The difference in recognition performance between Type-II and Type-III training data is not so distinctive. Although the best performance is achieved when Type-III training data are adopted, the length of training data is much longer than that of Type-II ones because every window frame is overlapped each other. Therefore, considering the computational costs, Type-II training data are preferred.

5. Discussions

From the experimental results, the following questions arise:

1. Why is the performance of ICA-matching high when small dimensional ICs are adopted as feature vectors?
2. Why is the performance of ICA-matching high when ICA-bases are obtained from noise-added training data?

Although we have not had exact answers to these questions yet, we can show what factors are related to the performance improvement. Let us study these factors in the followings.

Figure 2 shows the first sixteen ICA-bases that are obtained from noiseless Type-I training data. Each square box with 16×16 light and shade pixels represents a ICA-base. A light and shade pixel shows a value of a

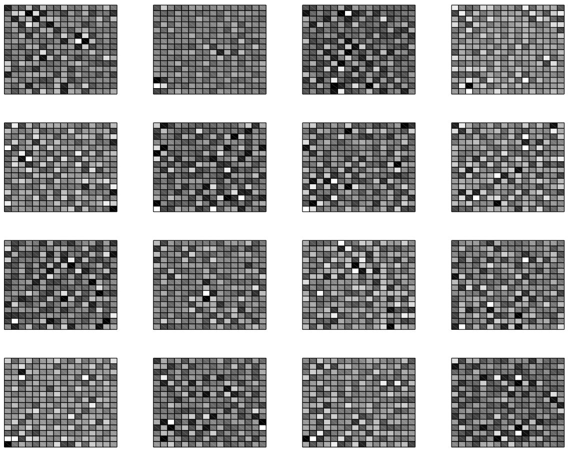

Figure 2: The first 16 ICA-bases obtained from noiseless Type-I training data.

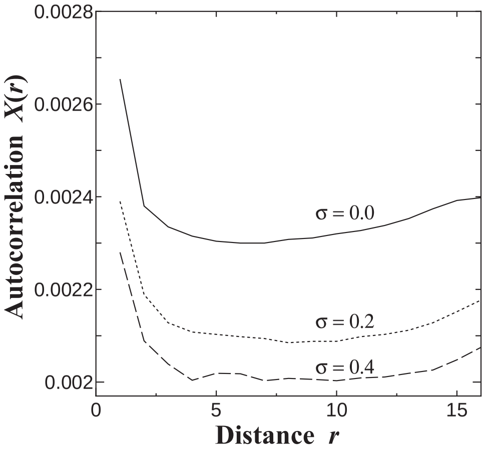

Figure 3: Autocorrelation of absolute values of ICA-base elements (Type-I).

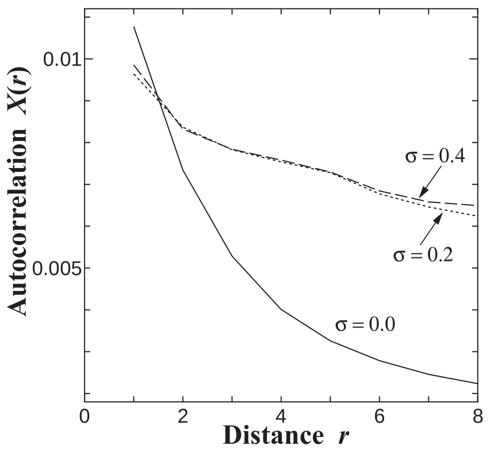

Figure 4: Autocorrelation of absolute values of ICA-base elements (Type-II, $S = 8$).

ICA-base element, black means minus value and white means plus value (i.e. gray means zero). The topology of pixels in Fig.2 is equivalent to that of the corresponding input image, therefore we can see from their absolute values what input information is mainly extracted by each ICA-base. As seen in Fig.2, ICA-base elements with large absolute values are widely distributed over input space. This suggests that global input features tend to be reflected on ICs. In this situation, it is expected that even small structural noises in test patterns (e.g. shift, scale variation, distortion, etc.) cause large variation in ICs. Since ICA-matching is done through the comparison between such ICs and reference ICs, one can say that such characteristics of ICA-bases can cause performance deterioration in recognition.

If we believe that broad distribution of large absolute values of ICA-base elements affects the deterioration of performance, it is possible to give an answer to the second question stated before. Let us first define the following autocorrelation of absolute values of ICA-base elements:

$$X(r) = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{N_i(j,r)} \sum_{k \in D_i(j,r)} \mid b_{ij} \mid \times \mid b_{ik} \mid, \quad (6)$$

where r is the distance between two pixels in input space. $D_i(j,r)$ means an index set of ICA-base elements with distance r from the jth element, and $N_i(j,k)$ is the number of indices belonging to $D_i(j,r)$. From the definition of Eq.(6), $X(r)$ is large when absolute values of two ICA-base elements with distance r are simultaneously large. As seen in Fig.3, $X(r)$ becomes smaller when the rate of adding noise is increasing. These results suggest that increasing noise prevents ICA-bases from extracting widely-distributed input features; as a result, the recognition performance improves.

On the other hand, when Type-II training data are adopted, the relation between performances and autocorrelations is different from that for Type-I training data. Figure 4 depicts calculated autocorrelations for Type-II training data when $S = 8$. As seen in Fig.4, the result for noiseless training data ($\sigma = 0.0$) demonstrates a tendency that large-valued ICA-base elements are localized. This is also certified from the ICA-bases shown in Fig.5. These ICA-bases look like wavelet type filters (the same results have been reported for natural images[6][7]). However, such ICA-bases are not the best at least in our recognition experiments. From the previous results shown in Fig.1(b), the best performance is realized when $\sigma = 0.4$. Hence, it seems that widely-distributed large-valued ICA-base elements contribute to the performance improvement in opposition to the case of Type-I training data.

To conclude, one can say that the effect of noise addition to Type-I training data is equivalent to the effect of adopting small windows (i.e. reducing ICs' dimensions) for Type-II, III training data: extracting local features of input images. However, the effect of noise addition to Type-II, III training data is obviously different from the effect of noise addition to Type-I training data. From the recognition results, it seems that such noise addition contributes to further improvement of recognition performance rather than extracting local features. It has not been cleared yet how noise addi-

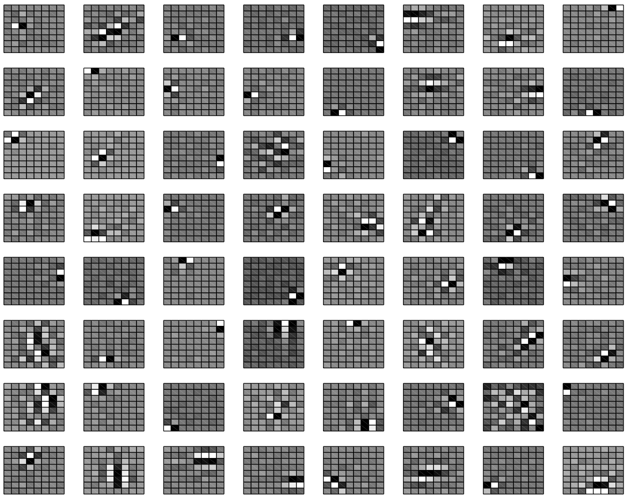

Figure 5: The ICA-bases obtained from noiseless Type-II training data ($S = 8$).

tion affects ICA-bases in this case. This is our open question.

6. Conclusions

We presented an approach to recognizing Japanese Hiragana characters utilizing independent components of input images (we call this method ICA-matching). From recognition experiments, we demonstrated that ICA-matching outperformed two conventional approaches (pixel-matching and PCA-matching) when the following parameters were adjusted properly: dimensions of feature vectors and rate of noise added to training data. We found that reducing ICs' dimensions and adding noise to training data were effective in performance improvement. When dimensions of ICs are large, ICA-bases generated from noise-added training data are apt to extract local features of input images. One can say that such characteristics of ICA-bases realize robustness for structural noises (e.g. invariance for shift, rotation, distortion, etc.). When dimensions of ICs are small, it seems that ICA-bases generated from noise-added training data extract somewhat global features. However, the recognition accuracy increases in opposition to the previous case. This reason has not been clarified yet, hence it is left as our future works.

Acknowledgment

The authors would like to express sincere thanks to Dr. Maekawa (Communications Research Laboratory, Ministry of Posts and Telecommunications) for his instructive discussions and useful comments.

References

[1] A. J. Bell and T. J. Sejnowski: "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, **7**, 1129-1159, 1995.

[2] S. Amari, A. Chichocki, and H. Yang: "A new learning algorithm for blind signal separation", *Advances in Neural Information Processing Systems 8*, MIT press, 757-763, 1996.

[3] J. Karhunen and E. Oja: "A class of neural networks for independent component analysis", *IEEE Trans. on Neural Networks*, **8**, 3, 486-503, 1997.

[4] M. S. Bartlett and T. J. Sejnowski: "Viewpoint invariant face recognition using independent component analysis and attractor networks", *Neural Information Processing System-Natural and Synthetic*, **9**, MIT Press, Cambridge, MA, 817-823, 1997.

[5] A. Hyvärinen and E. Oja: "A fast fixed-point algorithm for independent component analysis", *Neural Computation*, **9**, 1483-1492, 1997.

[6] A. J. Bell and T. J. Sejnowski: "Edges are the 'independent components' of natural scenes", *Advances in Neural Information Processing Systems 9*, MIT press, 1997.

[7] 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.