International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **8**, August **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 5003–5014

A STUDY ON JAPANESE HISTORICAL CHARACTER RECOGNITION USING MODULAR NEURAL NETWORKS

Tadashi Horiuchi[1] and Satoru Kato[2]

[1]Department of Control Engineering
[2]Department of Information Engineering
Matsue College of Technology
14-4 Nishi-ikuma, Matsue 690-8518, Japan
{ horiuchi; kato }@matsue-ct.ac.jp

Received May 2010; revised October 2010

Abstract. *In this research, we develop the Japanese historical character recognition system for the reading support system for Japanese historical documents. We use the directional element features as feature vectors and use the modular neural networks as pattern classification method. The modular neural networks consist of two kinds of classifiers: a rough-classifier and a set of fine-classifiers. In the rough-classifier, we use the multi-templates matching in order to find the several candidates of character categories for the input pattern. The multi-templates for each category are derived from the input samples using the Self-Organizing Maps (SOM). In the fine-classifiers, we use the multi-layered perceptrons (MLP), each of which solves the two-category classification problem. The final result of character recognition is derived by selecting the MLP which has the maximum output among the set of MLPs. We also use the rough-classifier for the selection the training samples in the learning process of multi-layered perceptrons in order to reduce the learning time. Through the experiments of historical character recognition for 57 character categories, we confirmed the effectiveness of our proposed method compared with the conventional research.*

Keywords: Japanese historical character recognition, Modular neural networks, Directional element features, Self-organizing maps

1. **Introduction.** It is one of the fundamental works to translate the historical characters called "kuzushi-ji" written in the Japanese historical documents into the contemporary characters in Japanese historical studies and Japanese literature. Not only in these academic fields, there are also many general people who enjoy reading the Japanese historical documents in the weekend as one of the hobbies. In order to read "kuzushi-ji" characters and translate it into the contemporary character, the expert knowledge on Japanese historical characters and documents is required. However, it takes a lot of period of learning and training to acquire such knowledge.

In this research, we develop the Japanese historical character recognition system to integrate into the reading support system for Japanese historical documents as shown in Figure 1. In this reading support system, the user selects the unknown character by mouse operation and the character recognition system outputs five candidate characters with each typical character image. In order to realize this historical character recognition system, it is necessary to recognize many kinds of "kuzushi-ji" characters and have the robustness for transformation of character shapes. The multi-layered perceptron is one of the useful methods for such pattern recognition. However, it is difficult to deal with many character categories in a single neural network. Therefore, we develop the Japanese historical character recognition system using the modular neural networks and reveal

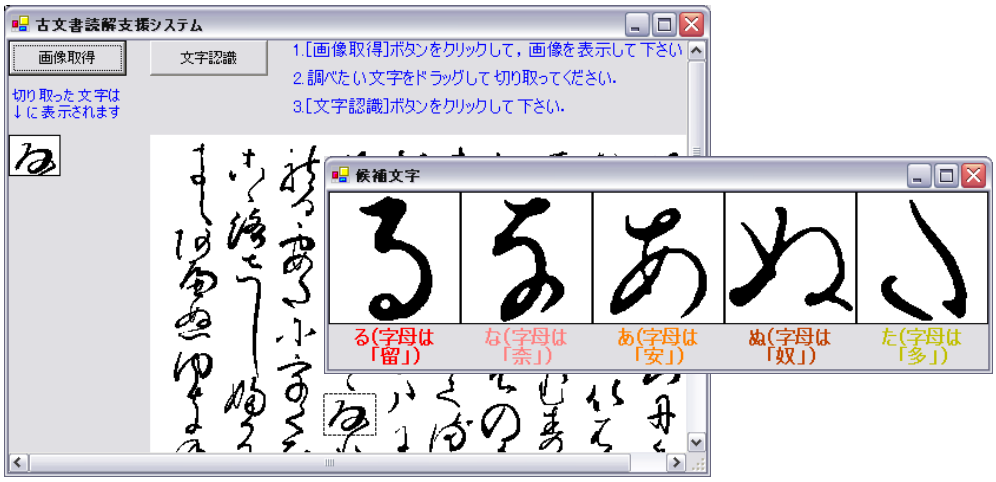


FIGURE 1. Prototype system of the reading support system for Japanese historical characters

how effective for more and more categories of historical characters than the conventional research.

In this research, we realize the modular neural networks consisting of two kinds of classifiers: a rough-classifier and a set of fine-classifiers. In the rough-classifier, we use the multi-templates matching in order to find the several candidates of character categories for the input pattern. The multi-templates for each category are derived from the input samples using the Self-Organizing Maps (SOM). In the fine-classifiers, we use the multi-layered perceptron (MLP) for each classifier, each of which solves the two-category classification problem. Only the several MLPs for the candidates derived in the rough-classifier are activated in the fine-classifier. The final result of character recognition is derived by selecting the MLP which has the maximum output among the set of MLPs.

In the experiment of historical character recognition for 15 character categories and 57 character categories, we confirmed that our proposed method can achieve higher classification accuracy compared with the conventional research. We also use the rough-classifier for the selection the training samples in the learning process of multi-layered perceptrons in order to reduce the learning time. Through the experiments of historical character recognition for 57 character categories, we confirmed that proposed modular neural network can reduce learning time drastically, with keeping high recognition accuracy.

2. **Related Works.** There are several studies on Japanese historical character recognition system in order to help researchers as well as general people to read such historical characters [1]. The Historical Character Recognition Project [2] is one of the most famous projects in this area. As a part of this project, Waizumi et al. [3] achieved the 96.67% accuracy for 16 character categories in the historical character dataset "HCD1" by using the directional element features and multi-layered perceptrons. However, the number of character categories to be able to recognize is small. Therefore, it is necessary to realize the Japanese historical character recognition for many character categories.

In general, there are several kinds of modular neural networks such as Mixture of Experts [4], Modular Perceptron Networks [5] and so on. In our research, we realize the modular neural networks consisted of two kinds of classifiers: a rough-classifier using SOM and a set of fine-classifiers using MLPs.

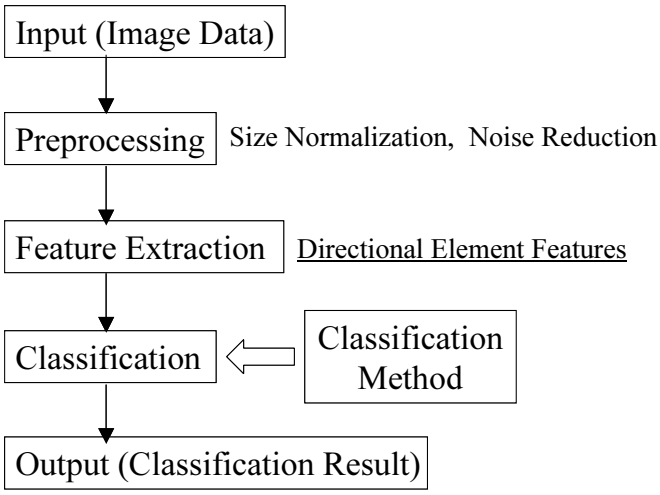


FIGURE 2. Flowchart of the character recognition system

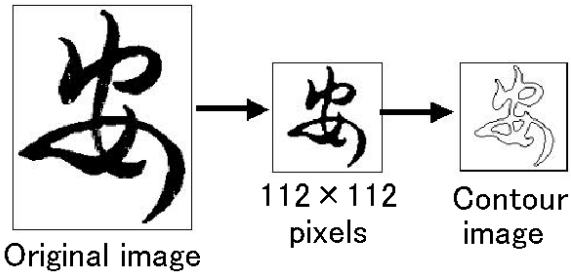


FIGURE 3. Preprocessing process

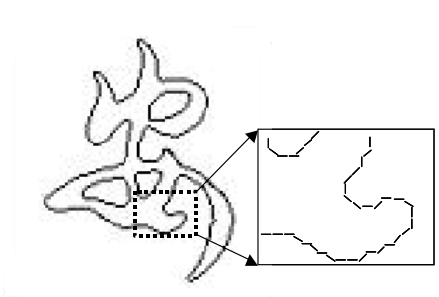
In addition, there are several related work on the applications of SOM such as a study on the visualization for fuzzy retrieval using self-organizing maps [6] and a proposal of semantic matching and annotation method of the images by self-organizing maps [7].

3. **Historical Character Recognition System.** The character recognition system for Japanese historical documents includes the following step as well as contemporary character recognition system, 1) input of the character image, 2) preprocessing, 3) feature extraction, 4) classification, 5) output of the classification result. The flowchart of the character recognition system for Japanese historical documents is illustrated in Figure 2.

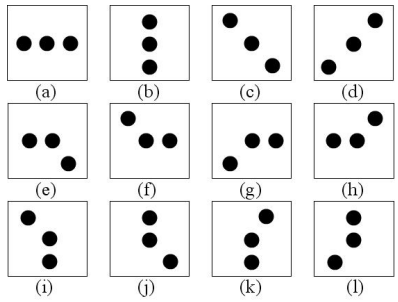
3.1. **Preprocessing.** In the preprocessing stage, the size of the character image is normalized and then the image is smoothed. In the size normalization, a linear normalization method is employed and an input image is adjusted to 128×128 dots. As a result of smoothing, bumps and holes of strokes are patched up by using 3×3 mask.

Then, contour extraction is done. If a white pixel adjoins a black pixel to the upward, downward, left or right direction, the black pixel is regarded as on contour. The feature vector is extracted from the pixels of contour. An example of the preprocessing process is shown in Figure 3.

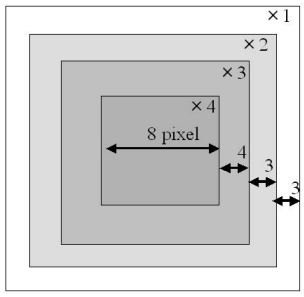
3.2. **Feature extraction.** In this section, the directional element feature (DEF), which was proposed for modern handwritten Japanese character recognition, is described. The operation for extracting the DEF includes the following steps.



(a) Extraction of the directional element features



(b) Line elements in the directional element features



(c) Weights in the directional element features

FIGURE 4. Directional element features

3.2.1. *Dot orientation.* In dot-orientation, four types of line elements, vertical, horizontal and two oblique lines slanted at ± 45 degree, are assigned to each black pixel. For a center black pixel in a 3×3 mask, two cases are considerable: One type of line element is assigned (see Figure 4(b)-(a) to Figures 4(b)-(d)); or if three black pixels are connected as in Figure 4(b)-(e) to Figure 4(b)-(l), two types of line elements are assigned. For example, in the case of Figure 4(b)-(k), 45 degree line element and vertical line element are assigned simultaneously. Here, eight-neighbors are used to determine the direction of a black pixel.

3.2.2. *Vector construction.* Consider an input pattern placed in a 128×128 mesh for which dot orientation has been completed. First, the 128×128 mesh is divided into 49, or 7×7 subareas of 28×28 pixels where each subarea overlaps eight pixels of the adjacent subareas (see Figure 4(c)). Furthermore, each subarea is divided into four areas A, B, C and D. A is a 8×8 area in the center. B is a 16×16 area exclusive of area A. C is a 22×22 area exclusive of areas A and B. D is a 28×28 area exclusive of areas A, B and C. In order to reduce the negative effect caused by position variation of image, weighting factors are defined greater at the center of each subarea and decrease towards the edge. The weight of each area is 4, 3, 2, 1 for the areas A, B, C, D, respectively. For each subarea, a four-dimensional vector (x_1, x_2, x_3, x_4) is defined where x_1, x_2, x_3 and x_4 represent the element quantities of the four orientations. Each element quantity is calculated by the following equation:

$$x_j = 4x_j^{(A)} + 3x_j^{(B)} + 2x_j^{(C)} + x_j^{(D)} \quad j = 1, \cdots, 4 \tag{1}$$

where $x_j^{(A)}$, $x_j^{(B)}$, $x_j^{(C)}$ and $x_j^{(D)}$ denote the quantity of each element in A, B, C and D, respectively. Since each subarea has four dimensions, the vector for one character is 196, or 49×4 dimensions. This vector is called "directional element feature".

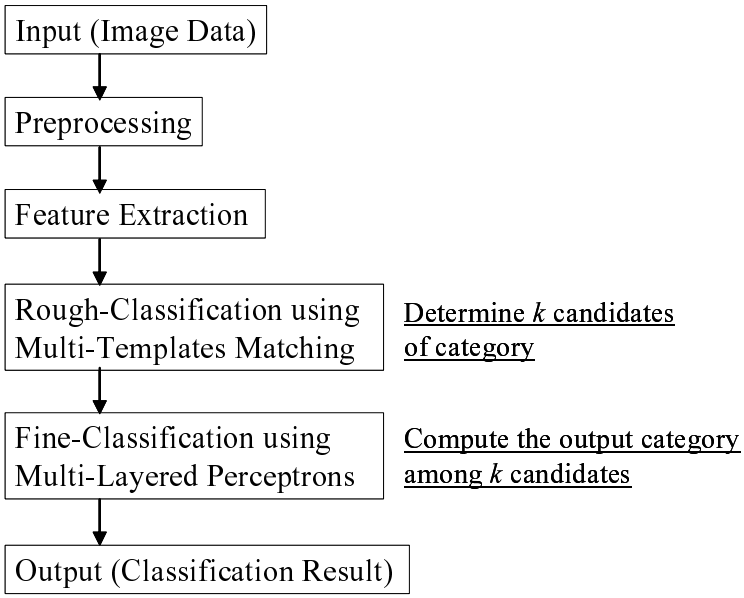


FIGURE 5. Flowchart of the character recognition system using modular neural networks

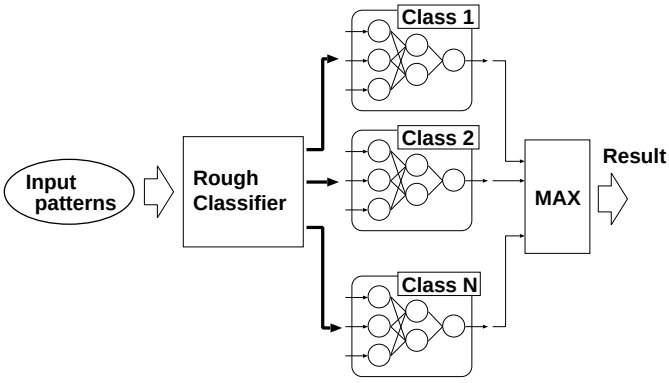
3.3. **Classification.** Details of the classification method will be described in the next section.

4. Modular Neural Networks for Historical Character Recognition.

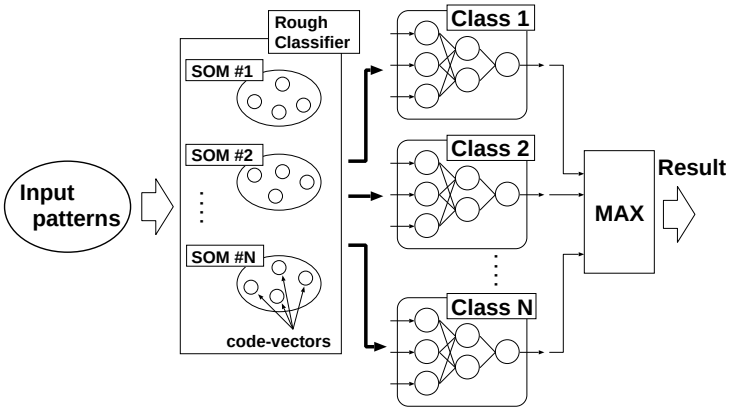
4.1. **Pattern recognition by modular neural networks.** The flowchart of the character recognition system for Japanese historical documents using modular neural networks is shown in Figure 5. As shown in this figure, the classification part consists of the rough-classification and the fine-classification. The basic structure of our pattern classifier using the modular neural network is also shown in Figure 6. Figure 6(a) illustrates the basic structure of the conventional modular neural networks. Figure 6(b) shows the proposed structure of the modular neural networks using SOM in the rough-classifier. That is, our pattern classifier consists of two kinds of classifiers: a rough-classifier and a set of fine-classifiers using MLPs. The rough-classifier determines k candidates of character category (k is small positive integer number) for the input pattern. Then, a set of fine-classifiers computes the output category the input pattern among k candidates in the rough-classifier.

The fine-classifiers are realized using a set of MLPs, each of which solves the two-category classification problem. In other words, each MLP is learned to output the value '1' only when the patterns belonging to the class for which the MLP is responsible, and otherwise to output the value '0'. The number of MLPs N is equal to the number of character categories to recognize. The rough-classifier derives the several candidates of character category for the input pattern by using the multi-templates matching. We use Self-Organizing Maps (SOM) in order to obtain the multi-templates for each category from input data.

4.2. **Rough-classification.** Overview of the template matching and the multi-templates matching is illustrated in Figure 7. In the template matching, each template is derived by computing the mean vector for the input data of each class (category). This method is so simple that we cannot expect the high accuracy for the real-world complex data. In the multi-templates matching, the templates are derived as the code-vectors of SOM for



(a) Basic structure of the modular neural networks



(b) Proposed structure of the modular neural networks using SOM in the rough-classifier

FIGURE 6. Basic structure of pattern classifier using the modular neural networks

each class after the learning of SOM. We use the multi-templates matching and SOM in the rough-classification.

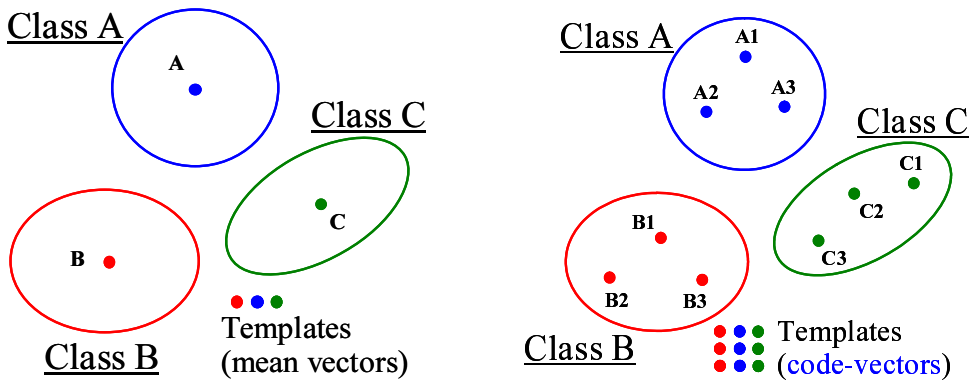
4.2.1. *Structure of SOM and its learning algorithm.* Basic structure of SOM is shown in Figure 8. In the learning algorithm of SOM [9], the code-vector $\boldsymbol{w}$ for the winner cell which is nearest to the input vector $\boldsymbol{x}$ and its neighborhood cells are updated by the following equation:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t)\Phi(p_i)(\boldsymbol{x} - \boldsymbol{w}_i(t)) \tag{2}$$

$$\Phi(p_i) = \exp\left(-\frac{{p_i}^2}{\sigma^2(t)}\right) \tag{3}$$

where $\alpha(t)$ is learning coefficient after t learning steps. The coefficient starts from its initial value α_{ini} and then decreases monotonically as t increases, thus reaching its minimum value at the pre-set maximum number of learning steps $T_{\max}$. $\Phi(p_i)$ is a neighborhood function with the center at winner cell c and p_i is the distance from cell i to the winner cell c. In Equation (3), $\sigma(t)$ is a time-varying parameter that defines the neighborhood size in the competitive layer. As well as parameter $\alpha(t)$ in Equation (2), this parameter decreases monotonically from the initial value σ_{ini} as t increases.

4.2.2. *Multi-templates learning using SOM in rough classifier.* In general, SOM has significant characteristics that the distribution of code-vectors after the learning of SOM reflects the distribution of the input data. That is, code-vectors tend to converge on the area where the density of the input data is high. Based on this characteristics of



(a) Template matching (b) Multi-templates matching

FIGURE 7. Overview of the template matching and the multi-templates matching

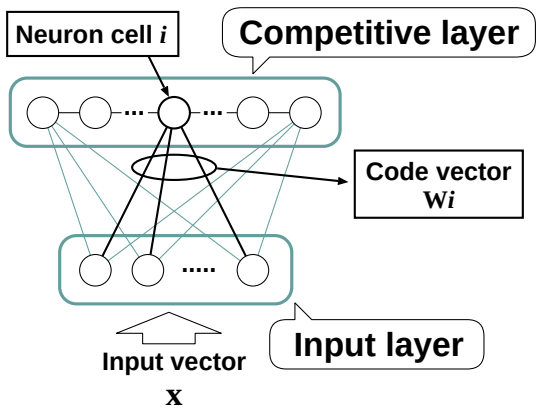


FIGURE 8. Basic structure of SOM (one-dimensional SOM)

SOM, we construct the rough-classifier of our modular neural networks by assigning each SOM to each class category. Here, we use one-dimensional SOM for each SOM in the rough-classifier.

In the rough-classification, we select k templates which are nearest to the test sample among the all templates for all class categories. The class categories of the selected k templates result in the candidate classes in the fine-classifiers.

4.3. **Fine-classification.** In the fine-classifiers, we use the multi-layered perceptron (ML P) for each classifier, each of which solves the two-category classification problem. In other words, each MLP is learned to output the value '1' only when the patterns belonging to the class for which the MLP is responsible, and otherwise to output the value '0'.

Only k MLPs for the candidate classes derived in the rough-classifier are activated in the fine-classifiers. The final result of character recognition is derived by selecting MLP which has the maximum output among the set of MLPs.

Each fine-classifier is realized by three layered perceptron with 10 units in the hidden layer. The number of nodes in the input layer in each MLP is 196, which means the dimension of the directional element feature described in the Section 3.2. Therefore, each MLP has 196 units in the input layer, 10 units in the hidden layer and 1 unit in the output layer.

The backpropagation algorithm, which adjusts the weight vectors in the steepest descent direction, is a widely-used algorithm for multi-layered perceptron. However, it turns out

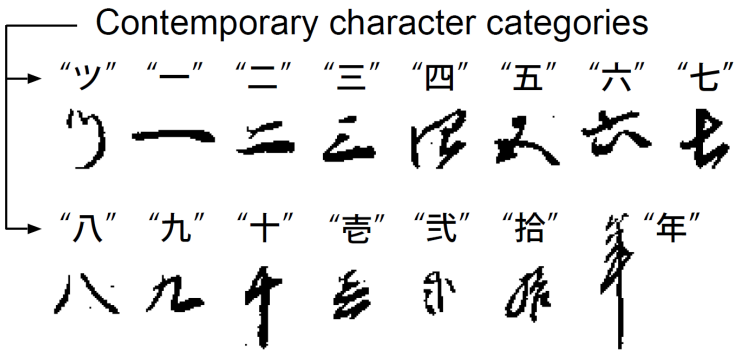


FIGURE 9. Examples of character images used in the experiment

that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms, search is performed along the conjugate directions, which produces generally faster convergence than steepest descent directions. In this research, we adjusts the weight vectors by *Fletcher-Reeves update* method, which is one of the conjugate gradient algorithms [11]. The outline of Fletcher-Reeves update method is shown in the following.

1. Choose an initial weight vector $\boldsymbol{w}_0$.
2. Initialize $k = 0$, $\beta_0 = 0$.
3. Derive the gradient vector $\boldsymbol{g}_k$ at the weight vector $\boldsymbol{w}_k$ based on the error function E as follows.
$$\boldsymbol{g}_k = \left.\frac{\partial E}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}_k}$$
4. If the stopping criterion $\|\boldsymbol{g}_k\| = 0$ is satisfied, then exit the algorithm.
5. Derive the new search direction $\boldsymbol{p}_k$ by a linear combination of the current gradient and the previous search direction, as shown in the following equation.
$$\boldsymbol{p}_k = -\boldsymbol{g}_k + \beta_k \boldsymbol{p}_{k-1}$$
The various versions of the conjugate gradient algorithm are distinguished by the manner in which the constant β_k is computed. For the Fletcher-Reeves update, the procedure is as follows.
$$\beta_k = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1}}$$
6. Update the weight vector $\boldsymbol{w}$ by the following equation.
$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \alpha_k \boldsymbol{p}_k$$
7. Set $k = k + 1$ and go to Step 3.

5. Experiments.

5.1. Experiment 1 (evaluation of rough-classifier).

In order to verify the performance of the rough-classifier, we carried out the recognition experiments using historical character dataset HCD1 and HCD1a-e which are open to be public by Historical Character Recognition Project [2].

In our recognition experiments, we use 15 character categories from HCD1 and 42 character categories from HCD1a-e, which satisfies the condition that 200 character samples exist for each character category. Examples of character images in HCD1 are shown in Figure 9. In addition, the list of 42 character categories in HCD1a-e is shown in Table 1.

TABLE 1. 42 character categories used in the experiment (HCD1a-e)

No.	category	*No.*	category	*No.*	category	*No.*	category
16	田	17	畑	18	高	19	石
20	斗	21	升	22	合	23	金
24	両	25	分	26	朱	27	家
28	軒	29	間	30	馬	31	疋
32	内	33	人	34	男	35	女
36	〆	37	長	38	横	39	夕
40	父	41	母	42	子	43	忰
44	祖	45	弟	46	娘	47	房
48	村	49	組	50	頭	51	百
52	姓	53	代	54	借	55	賃
56	質	57	地				

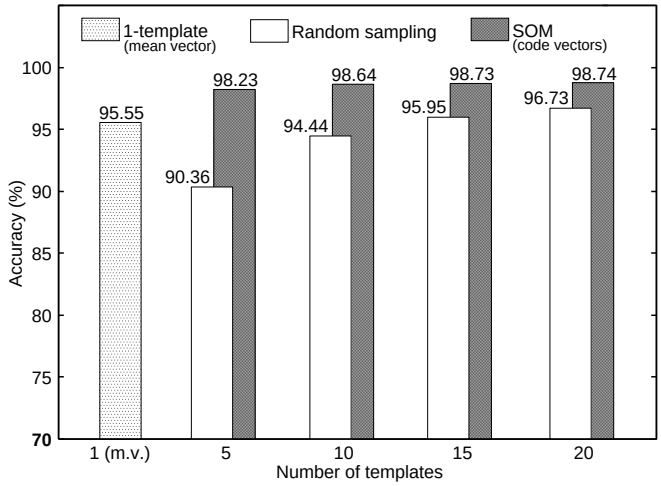
In the experiment of historical character recognition, we obtained the recognition accuracy by calculating the average of 10 sets of 2-fold cross-validation. We changed the number of cells N_c in the competitive layer in SOM from 5 to 20. Here, the value of N_c means the number of templates for each class category in the multi-templates matching. The values of other learning parameters in SOM are decided as follows through the preliminary experiments: $\alpha_{ini} = 0.3$, $\sigma_{ini} = 1.0$, $T_{\max} = 20,000$.

Figure 10(a) and Figure 10(b) indicate the recognition accuracy for 15 character categories (HCD1) and 57 character categories (HCD1+HCD1a-e). In these figures, we show the results of the following methods: Method 1: select the mean vector as a template (conventional template matching method: "1-template" in the figures), Method 2: randomly select 5 - 20 templates from the learning samples ("Random sampling" in the figures), Method 3: use the code vectors of SOM as templates ("SOM" in the figures) are also shown in the figures.

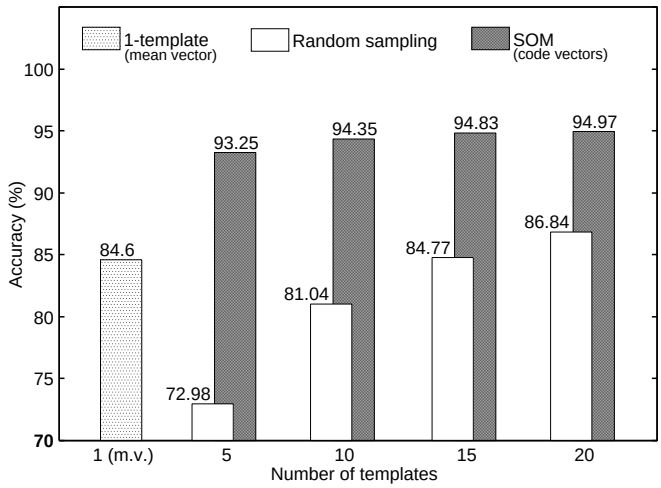
From the figures, we can see the proposed rough-classifier using SOM (Method 3) shows higher accuracy than conventional template matching (Method 1) and multi-template matching with randomly selected templates (Method 2). Moreover, we can confirm that the proposed rough-classifier using SOM shows higher accuracy with small number of templates. We can see this tendency more clearly in case of 57 character categories compared with in case of 15 character categories. For example, we can achieve the accuracy of 94.97% using 20 templates for each class and the accuracy of 93.25% using 5 templates for each class. The difference of accuracy between them is small. We think this is because the multi-templates derived by SOM are suitable to reflect the distribution of the input patterns of each character category.

5.2. **Experiment 2 (evaluation of modular neural networks).** In this section, we carried out the character recognition, in order to confirm the effectiveness of the proposed modular neural networks with rough-classifier and a set of fine-classifiers.

In this experiment, we use the following parameter values: the number of cells N_c in the competitive layer in SOM is 10, other learning parameters in SOM are same as previous section. We changed the number of candidates in the rough-classifier for the selection of the training samples from 5 to 15. Each fine-classifier is realized by three layered perceptron with 10 units in the hidden layer. Therefore, each MLP has 196 units in the input layer, 10 units in the hidden layer and 1 unit in the output layer. We used batch



(a) In case of 15 character categories (HCD1)



(b) In case of 57 character categories (HCD1+HCD1a-e)

FIGURE 10. Comparison of the experimental results of the rough-classifier

learning method and set the number of learning iteration $Epoch = 100$. We applied the conjugate gradient backpropagation with *Fletcher-Reeves update*.

The parameter in the recognition after the learning, we set the value $k = 2$. The value of k means the number of activated MLPs among the all MLPs, as shown in Section 4.1. This is decided through the preliminary experiments.

As same as the previous section, we obtained the recognition accuracy by calculating the average of 10 sets of 2-fold cross-validation. The programs for learning and recognition are implemented by using MATLAB. The computer used in the experiments has CPU of Core2Duo T7200 (2GHz) and Main Memory of 1Gbyte.

Figure 11 shows the relationship between classification accuracy and learning time in case of changing the number of candidates in the rough-classifier for the selection of the training samples.

When we select the training samples $(5, 10, 15)$, we can obtain almost same recognition accuracy compared with the case without selection of the training samples ("Full (no selection)" in the Figure 11). All the recognition accuracy of these four case are about 95% (from 95.11% to 95.35%). Moreover, we can see that great reduction of learning time is realized by the selection of the training samples. For example, we can obtain 95.3%

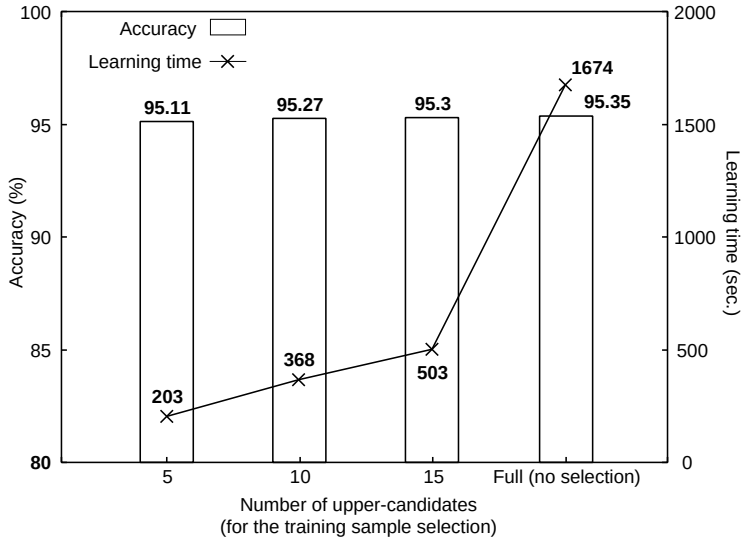


FIGURE 11. Relationship between classification accuracy and learning time

TABLE 2. Summary of the experiments

	Accuracy (%)	
	15 character categories	57 character categories
Rough-classifier (template matching)	95.55	84.60
Rough-classifier (multi-templates matching)	98.74	94.97
Rough-classifier + Fine classifiers (Modular NNs)	98.95	95.35

accuracy with 503 seconds of learning time which is less than one third in case without selection of the training samples which takes 1674 seconds). Therefore, we confirmed that we are able to reduce the learning time of modular neural networks without decrease of the recognition accuracy by the selection of the training samples using SOM in the rough-classifier.

We also show the summary of the experiments 1 and 2 in the Table 2. We confirmed to obtain quite high recognition accuracy of 95.35% for 57 character categories, which contain more character categories than the conventional research [3].

6. **Conclusions.** In this paper, we develop the Japanese historical character recognition system for the reading support system for Japanese historical documents. In order to realize the historical character recognition system, we use the directional element features and use the modular neural networks. The modular neural networks consist of two kinds of classifiers: a rough-classifier using multi-templates matching and a set of fine-classifiers using MLPs.

Through the experiments of Japanese historical character recognition for 57 character categories, we confirmed the effectiveness of our proposed method compared with the conventional research. The main results of this paper are summarized as follows:

1. We realized the Japanese historical character recognition system for the reading support system for Japanese historical documents.
2. We confirmed this historical character recognition system has quite high recognition accuracy of 95.35% for 57 character categories, which contain more categories than the conventional research.
3. By multi-templates learning using SOM in the rough-classifier, we revealed that our proposed modular neural network can reduce learning time drastically, with keeping high recognition accuracy.

There are several future works including the comparison of recognition performance of support vector machines (SVM) and nearest-neighbors classifier with Mahalanobis distance and other classification methods. Moreover, we are going to apply the recognition methods to the reading support system for Japanese historical documents.

Acknowledgment. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 20560389, 2008-2010 in Japan.

REFERENCES

[1] S. Yamada and M. Shibayama, Outline of historical character recognition project, *Journal of IPSJ*, vol.43, no.9, pp.950-955, 2002.
[2] *Historical Character Recognition Project*, http://www.nichibun.ac.jp/˜shoji/hcr/index.html.
[3] Y. Waizumi et al., A study for character recognition of ancient documents using neural network, *IPSJ Technical Report (Computer and Humanities)*, vol.2000, no.8, pp.9-15, 2000.
[4] R. A. Jacobs and M. I. Jordan, Learning piecewise control strategies in a modular neural network architecture, *IEEE Trans. on Systems, Man and Cybernetics*, vol.23, pp.337-345, 1993.
[5] H. C. Fu, Y. P. Le, C. C Chiang and H. T. Pao, Divide-and-conquer learning and modular perceptron networks, *IEEE Trans. on Neural Networks*, vol.12, no.2, pp.250-263, 2001.
[6] T. Samatsu, K. Tachikawa and Y. Shi, Visualization for fuzzy retrieval using self-organizing maps, *ICIC Express Letters*, vol.3, no.4(B), pp.1345-1350, 2009.
[7] H. C. Yang and C. H. Lee, Semantic matching and annotation of images by self-organizing maps, *International Journal of Innovative Computing, Information and Control*, vol.5, no.3, pp.677-688, 2009.
[8] K. Saruta, N. Kato, M. Abe and Y. Nemoto, High accuracy recognition of ETL9B using exclusive learning neural network-II (ELNET-II), *IEICE Trans. on Inf. and Syst.*, vol.E79-D, no.5, pp.516-522, 1996.
[9] T. Kohonen, *Self-Organizing Maps*, 3rd Edition, Springer-Verlag, 2000.
[10] N. Sun, M. Abe and Y. Nemoto, A handwritten character recognition system by using improved directional element feature and subspace method, *IEICE Trans.*, vol.J78-D-II, no.6, pp.922-930, 1995.
[11] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1996.