PAPER

A Character-Based Postprocessing System for Handwritten Japanese Address Recognition

Keiji YAMANAKA†, *Nonmember*, **Susumu KUROYANAGI**†, *and* **Akira IWATA**†, *Members*

SUMMARY Based on a previous work on handwritten Japanese kanji character recognition [1], a postprocessing system for handwritten Japanese address recognition is proposed. Basically, the recognition system is composed of CombNET-II, a general-purpose large-scale character recognizer and MMVA, a modified majority voting system. Beginning with a set of character candidates, produced by a character recognizer for each character that composes the input word and a lexicon, an interpretation to the input word is generated. MMVA is used in the postprocessing stage to select the interpretation that accumulates the highest score. In the case of more than one possible interpretation, the Conflict Analyzing System calls the character recognizer again to generate scores for each character that composes each interpretation to determine the final output word. The proposed word recognition system was tested with 2 sets of handwritten Japanese city names, and recognition rates higher than 99% were achieved, demonstrating the effectiveness of the method.

key words: *neural network, Lexicon, handwritten kanji, character recognition, word recognition, postprocessing*

1. Introduction

Since its appearance in 1955 as a commercial product, the OCR has became more and more powerful because of advances in computer technology. Its application has successfully expanded from the reading of printed documents to hand-printed documents [2]–[5]. One of the principal reasons for its development is the necessity of automatically handling an uncountable quantity of documents daily, such as commercial forms and mail. However, OCRs have not yet achieved good results on handwritten character recognition mainly because of the great variability in handwriting styles.

A typical architecture used in OCRs for word recognition of Japanese and Chinese characters is composed of a feature extractor, a character recognizer that classifies each input character and generates n solutions that are input to a postprocessing system [3]. The postprocessing system compares these solutions with words in a lexicon and an interpretation that is most appropriate for the input character sequence is produced.

In this paper, we propose a solution for word interpretation from a list of possible candidates. The proposed system overcomes the common problem of postprocessing algorithms, that is, the necessity for many correct recognitions by the character recognizer before a correct word interpretation can be generated. Correct character recognition is difficult because the input image can be degraded and, in the case of *kanji* characters, the variety of writing styles, the structural complexity of the patterns, and the large number of character categories results in similar stroke structures.

In Sect. 2 the character recognizer utilized in this system is described. In Sect. 3, MMVA, the Modified Majority Voting Algorithm, for word recognition and its performance are presented. Section 4 presents a method of selecting the best solution between a set of word candidates generated by MMVA, and finally, the performance of the whole recognition system is discussed in Sect. 5.

2. The Character Recognizer CombNET-II

The name CombNET-II [1] comes from the fact that this neural net has a comb structure where the first layer is composed of a stem network which has the function of dividing the input feature vector space into several subspaces, and the following layers form the branch network modules which classify the input data into a specific category. CombNET-II was developed taking into consideration the fact that when we work with a large number of patterns, it is very difficult to achieve a good recognition capability using a simple feed-forward neural network, and when using the backpropagation learning algorithm, there is no guarantee that the network will converge, or that the network can be trained to the best configuration possible because the problem of local minima.

CombNET-II is a self-growing neural network that uses a set of small and manageable neural networks, called branch networks, to recognize a small quantity of patterns. It does not cause the local minimum state once the complexity of the problem to be solved by each branch module is restricted.

2.1 CombNET-II Neural Network Structure

Figure 1 shows the CombNET-II Neural Network Structure. It basically consists of two networks, the Stem Network and the Branch Network.

A vector-quantizing network forms the first layer of a stem and three-layered network modules form the

Manuscript received April 3, 1998.

†The authors are with the Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.

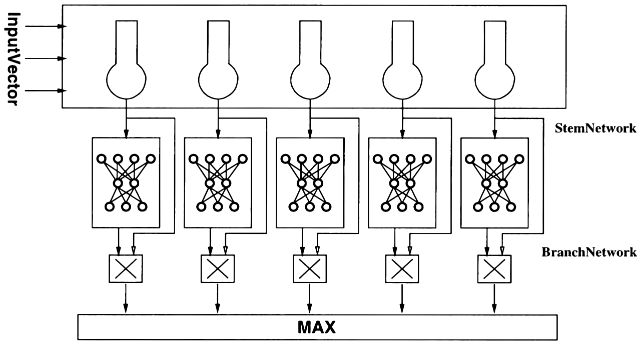

Fig. 1 CombNET-II neural network structure.

second layer. The number of branch modules is equal to the number of stem neurons. Given an input data, one of the subspaces is selected according to the similarity measure between the input vector and the synaptic weights of the stem neurons. Once the sub-space is selected, the corresponding branch module classifies the input data into a specific category.

During the recognition phase, it is possible for an incorrect stem neuron to be selected because the synaptic weights of the stem neurons can be similar to each other. If this happens, it is not possible to obtain the correct answer from the selected branch module. To avoid this problem, the outputs of the branch module and the output of the stem neuron are used to decide the final output of the recognition system, as shown in Fig. 1.

2.2 Training of CombNET-II

The training of CombNET-II proceeds in two stages. First, the stem neural network is trained using the Self-Growing Neural Network Learning Procedure [1] and the input feature vector space is divided into a number of subspaces. Each subspace has only a small number of categories so that each branch module always has a restricted number of outputs and interconnections. As the number of categories to be classified increases and input data become more varied, more stem neurons and branch modules are created in order to mantain proper sizes of the branch networks. After the production of these subspaces, the branch networks are trained using the Backpropagation Training Algorithm [6] and the training data that belong to the corresponding subspace, in order to generate discriminating boundaries of the categories.

2.3 Pattern Recognition with CombNET-II

Let I be the input word to be recognized, which is composed of characters $i_1, i_2, \ldots, i_k, \ldots, i_N$, where N is the length of the input word.

During recognition, the input data vector $\mathbf{D}_k$ corresponding to the input character i_k is input to the stem network and, the similarity measure between the vector $\mathbf{D}_k$ and the template pattern vector $\mathbf{M}_i$ corresponding to the stem neuron s_i, given by Eq. (1), is calculated.

$$sim\left(\mathbf{D}_k, \mathbf{M}_c\right) = \max_{i=1,\ldots,N} \left\{sim\left(\mathbf{D}_k, \mathbf{M}_i\right)\right\} \tag{1}$$

where

$$sim\left(\mathbf{D}_k, \mathbf{M}_i\right) = \frac{\mathbf{D}_k \cdot \mathbf{M}_i}{|\mathbf{D}_k| \cdot |\mathbf{M}_i|} \tag{2}$$

Some of the stem neurons with the largest similarities are selected and the corresponding branch modules are activated for further processing. Then, to define the category of the input character, the corresponding data vector is input to these branch modules. The final scores Z are calculated using the equation

$$Z = \theta_1^{1-\gamma} \cdot \theta_2^{\gamma}, \tag{3}$$

where θ_1 is the matching score of the stem neuron; θ_2 is the maximum output score of the branch network module.

The parameter $0 \leq \gamma \leq 1$ defines the relative significance of each output to calculate the outputs of the recognizer. An input is classified into the category that has the maximum score Z. Using the criteria given by Eq. (3), mistakes produced in the stem network are recovered by the branch network modules since high scores produced by branch network modules can be considered even if the stem neuron output is small.

3. MMVA - Modified Majority Voting Algorithm

The main objective of a character-based word recognition system is to obtain an interpretation of the input word based on a set of character candidates generated by the character recognition system. An algorithm for achieving this is the Majority Voting Algorithm which counts the score of each character candidate. In this section the Modified Majority Voting Algorithm used in this work is presented.

3.1 MVA - Majority Voting Algorithm

To recognize an input word using the Majority Voting Algorithm, a set of candidates and their scores are produced by the character recognizer for each character of the input word. Each character candidate is compared with each character belonging to the lexicon word and when it matches, the corresponding score is added to that word. The scores are accumulated and the word candidate that accumulates the highest score is selected as the final interpretation of the input word. The application of Eq. (4) gives the accumulated score to each word candidate X.

$$V\left(X\right) = \sum_{k=1}^{N} \sum_{j=1}^{H} \delta\left(x_k, y_{kj}\right) z_{kj} \tag{4}$$

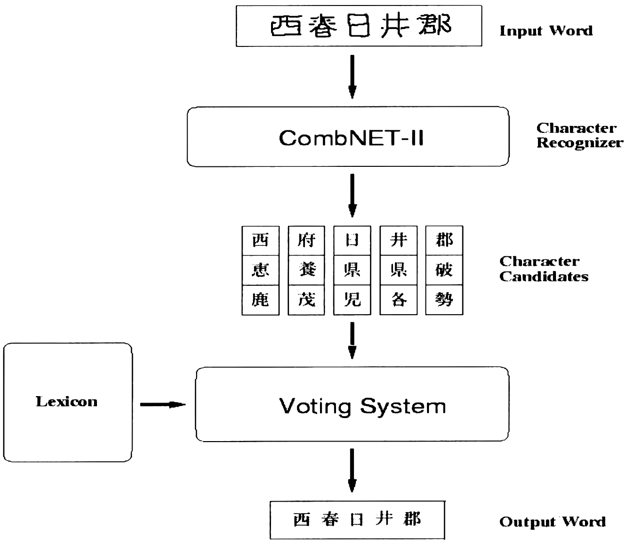

Fig. 2 Word recognition system using voting algorithm.

X is the word candidate composed of characters $x_1, x_2, \ldots, x_N$.
$V(X)$ is the accumulated score of the word candidate X.
y_{kj} is the jth character candidate of the kth character of the input word.
z_{kj} is the score for character y_{kj}.
H is the number of character candidates of each input character.
The function δ is defined as

$$\delta(x_k, y_{kj}) = \begin{cases} 1, & \textit{if} \quad x_k = y_{kj} \\ 0, & \textit{otherwise.} \end{cases}$$

The interpretation of the input word will be the word X that presents the highest value $V(X)$.

3.2 Experiment with MVA

Figure 2 shows the implemented recognition system. In this experiment, CombNET-II was used as a character recognizer that generates three character candidates for each input character, and the Majority Voting Algorithm was used to generate an interpretation of the input word.

The set of word patterns used is composed of 45 city names in Aichi Prefecture, as shown in Table 1.

The character database used in this experiment was collected from the ETL9 database produced by Electro Technical Laboratory, Tsukuba, Japan. From the ETL9 database, 134 *kanji* patterns written by 200 people were selected and half of them were used to train the character recognizer. The other half were used to make the words to be recognized. A total of 4500 words was submitted for recognition. The recognition results are shown in Table 2.

The recognition rate of 90.6% achieved by CombNET-II was calculated considering only the first character candidate corresponding to each input character.

Table 1 45 city names in Aichi prefecture.

安城市	一宮市	稲沢市	犬山市
岩倉市	大府市	岡崎市	尾張旭市
春日井市	蒲郡市	刈谷市	江南市
小牧市	新城市	瀬戸市	高浜市
知多市	知立市	津島市	東海市
常滑市	豊明市	豊川市	豊田市
豊橋市	名古屋市	西尾市	半田市
尾西市	碧南市	愛知郡	渥美郡
海部郡	北設楽郡	知多郡	中島郡
西春日井郡	西加茂郡	丹羽郡	額田郡
葉栗郡	幡豆郡	東加茂郡	宝飯郡
南設楽郡			

Table 2 Recognition results.

	Errors	Recog. Rate
CombNET-II (only characters)	–	96.8%
CombNET-II (names)	438	90.6%
MVA	55	98.6%

Table 3 Scores produced by CombNET-II.

	1st *kanji*	2nd *kanji*	3rd *kanji*
1st cand.	小 0.725	牧 0.881	市 0.834
2nd cand.	加 0.631	沢 0.855	市 0.823
3rd cand.	城 0.606	沢 0.806	古 0.787

Although MVA is fast and can produce good results, misrecognition occurs when a situation like the one shown in Table 3 arises.

There is a repetition of the *kanji* character 沢 in the set of candidates for the 2nd *kanji*. Since MVA performs word recognition by only accumulating the scores attributed to each character candidate, the two scores of the repeated *kanji* are accumulated, resulting inf the following scores for each name shown below.

word candidate	score
小牧市	3.265
稲沢市	3.320

MVA misrecognized the input word 小牧市.

3.3 MMVA - Modified Majority Voting Algorithm

The first layer of CombNET-II constitutes a stem network which quantizes the input feature vector space into several subspaces. However, because of the variability in handwriting styles and the large number of character categories that make *kanji* characters have similar stroke structures, the discriminating boundary process is not perfect and it is possible that some of the character patterns belong to more than one subspace. Therefore, during the character recognition phase, repetition of character candidates can occur for the same position in the word, as was illustrated in Table 3. When this repetition appears, the application of the

Table 4 Recognition results of MVA and MMVA.

	Errors	Recognition Rate
MVA	55	98.6%
MMVA	50	98.8%

Table 5 Scores produced by CombNET-II.

	1st *kanji*	2nd *kanji*	3rd *kanji*
1st cand.	知 0.822	芸 0.545	市 0.880
2nd cand.	志 0.807	愛 0.449	弁 0.840
3rd cand.	儀 0.798	巣 0.402	府 0.835

voting algorithm can produce an incorrect interpretation, as shown in the previous subsection. To avoid this kind of interpretation error, the original algorithm was modified. The original equation (4) is applied with the following change. Let $Y = (y_{k1}, y_{k2}, \ldots, y_{kH})$, be the set of M character candidates generated by CombNET-II for the k_{th} character of the input word and $Z = (z_{k1}, z_{k2}, \ldots, z_{kH})$ be the corresponding set of scores.

If during the application of equation (4) *repetition of a character is found in the set* Y, *the corresponding score in* Z *is not accumulated.*

The results produced by MMVA for the same test set as described in the previous subsection are shown in Table 4. About 9% of errors generated by MVA were recovered by MMVA.

3.4 Generating Multiple Interpretations

The Modified Majority Voting Algorithm produces two kinds of results. It can select the word candidate with the highest score that can be correct or wrong or produce a list of word candidates with the same highest score. An example of a list with two word candidates produced by MMVA is shown in the following.

Based on the scores shown in Table 5, MMVA generated two word candidates from the lexicon of Aichi Prefecture (Table 1), as shown in the following.

word candidate	score
知多市	1.702
知立市	1.702

When this situation arises nothing can be done by MMVA and another strategy is necessary. In the following, a strategy for selecting one word from a list of word candidates in conflict is proposed.

4. CA - The Conflict Analyzer

The proposed character-based word recognition system has, in its first stage, the character recognizer CombNET-II that produces candidates for each character that makes up the input word. Using this set of candidate characters and a lexicon, MMVA is applied to get an interpretation of the input word. If it produces more than one interpretation, the Conflict Analyzer calls the character recognizer CombNET-II to produce scores, now only for each character that forms these interpretations. MMVA is used to generate scores for each word candidate and select the one that has the highest score.

If, there is more than one word candidate with the same highest score, the recognition system will consider the input word as unrecognizable and the input word will be rejected.

The process of the proposed conflict analyzing algorithm is described in Table 6.

Table 6 Conflict-analyzing process.

1. Activate the character recognizer for each input character i_k
   a. For each interpretation W_m
      a.1. If not already done for category w_{mk}
         a.1.1. Get all outputs of the character recognizer corrresponding to category w_{mk}
         a.1.2. Select the output with the highest score
   b. Calculate $V(W_m)$
2. Select the interpretation W_m with the highest $V(W_m)$.

Let, I be the input word to be recognized which is composed of characters $i_1, i_2, \ldots, i_k, \ldots, i_N$.
$SI = \{W_1, W_2, \ldots, W_m, \ldots\}$ is the set of interpretations with the same highest score HS.
W_m is the mth interpretation composed of characters $w_{m1}, w_{m2}, \ldots, w_{mk}, \ldots, w_{mN}$.
$V(W_m)$ is the accumulated score of the interpretation W_m.

5. Simulation Results and Discussion

The proposed character-based handwritten word recognition system was tested using city names in Aichi and Gifu Prefectures in the Tokai region of Japan. Aichi has 45 cities and Gifu 31 cities. A database with 134 patterns of *kanji* characters, shown in Table 7, written by 150 people, extracted from the ETL9 database, was used in the experiments.
The proposed character-based handwritten word recognition system was tested using city names in Aichi, Gifu and Mie Prefectures that form the region of Tokai in Japan. Aichi has 45 cities, Gifu 31 cities and Mie 27 cities. A database with 134 patterns of Kanji characters, shown in Table 7, written by 150 people, extracted from ETL9 database was used in the experiments.

Fifty exemplars of each character were used for training the character recognizer CombNET-II and the other 100 exemplars as the city names to be recog-

Table 7 Table of 134 *kanji* patterns.

阿	愛	渥	旭	安	伊	井	一	稲	員	羽
益	岡	屋	加	可	賀	会	海	垣	各	楽
額	滑	蒲	刈	関	岩	岐	気	亀	儀	吉
久	宮	居	橋	熊	栗	桑	郡	恵	芸	犬
県	見	原	古	戸	江	高	坂	崎	三	山
四	市	志	児	治	鹿	重	春	小	松	上
城	常	新	瑞	瀬	勢	西	設	川	倉	巣
多	大	沢	谷	丹	知	中	張	鳥	津	田
度	土	島	東	豆	那	南	日	濃	破	幡
八	半	飯	斐	尾	美	浜	不	府	阜	武
部	碧	弁	宝	豊	北	牧	本	摩	務	牟
名	明	茂	野	揖	葉	養	立	鈴	婁	浪
老	鷲									

Table 8 31 city names in Gifu prefecture.

岐阜市	恵那市	大垣市	各務原市
可児市	関市	高山市	多治見市
土岐市	中津川市	羽島市	瑞浪市
美濃市	美濃加茂市	安八郡	揖斐郡
恵那郡	大野郡	海津郡	可児郡
加茂郡	郡上郡	土岐郡	羽島郡
不破郡	益田郡	武儀郡	本巣郡
山県郡	養老郡	吉城郡	

Table 9 Recognition results for Aichi prefecture city names.

	Errors	Recognition Rate
CombNET-II (names)	448	90.0%
MMVA	48	98.9%
Conflict Analyzer	20	99.5%

Table 10 Recognition results for Gifu prefecture city names.

	Errors	Recognition Rate
CombNET-II (names)	312	89.9%
MMVA	27	99.1%
Conflict Analyzer	12	99.6%

nized. Seven hundred an sixty eight peripheral direction contributivity (PDC) features [7], extracted from each handwritten *kanji* character, were used as input data for CombNET-II. PDC features extract the complexities of characters, such as local features of line directions and relationships among lines.

The city names used in this experiment are shown in Tables 1 and 8.

To test the proposed system, 4500 handwritten city names in Aichi Prefecture and 3100 in Gifu Prefecture divided in two lexicons were used.

The performance of the proposed recognition system is shown in Tables 9 and 10.

5.1 Conflicts Recovered by Conflict Analyzer

Table 11 shows the number of conflicts recovered by the conflict analyzer.

During the recognition, MMVA produced 48 misrecognitions of city names in Aichi Prefecture. From them, 7 were direct misrecognitions where no conflicts were generated and the word that received the highest score is not the correct interpretation of the input word. The other 41 were conflicts and 28 of them were recovered producing a recovery rate of 68.2% by the conflict analyzer.

As can be seen in Table 11, more than 60% of conflicts were correctly solved for these two sets of city names by using the proposed method, and the performance of the recognition system has been improved. In the following, two examples of correctly solved conflicts are shown. It can be seen in Table 12 that a list of two city names in Aichi Prefecture, 安城市 and 新城市, was generated by MMVA when the city name 安城市 was input. The application of CA solved the conflicts and the final output word was the correct interpretation of the input word.

In Table 13, the scores produced by CombNET-

Table 11 Recovery rate by Conflict Analyzer.

Prefecture	Number of Conflicts	Number of Recoveries	Recovering Rate
Aichi	41	28	68.2%
Gifu	24	15	62.5%

Table 12 Example of conflicts correctly solved.

	candidate	*kanji*	score	list of words generated by MMVA	recog. result by CA
1st *kanji*	1st	稲	0.463	安城市	安城市
	2nd	市	0.336	新城市	
	3rd	愛	0.330		
2nd *kanji*	1st	治	0.828		
	2nd	城	0.855		
	3rd	飯	0.820		
3rd *kanji*	1st	市	0.886		
	2nd	井	0.822		
	3rd	市	0.818		

Table 13 Outputs of CombNET-II for each character.

Branch Number	安	城	市	新
4		0.359		
7		0.855		
11		0.855		
21		0.855		
23		0.855		0.295
25		0.855	0.818	
26	0.312		0.886	0.324
27			0.886	0.324
28				0.324
29				0.324
30	0.312			
31		0.855		
32	0.312			
35			0.886	
39				
42	0.327			
43				0.324
51	0.327	0.855		0.324
62				
63		0.855		
66			0.886	

Table 14 Example of conflict correctly solved.

	candidate	*kanji*	score	list of words generated by MMVA	recog. result by CA
1st *kanji*	1st	江	0.530	安八郡	揖斐郡
	2nd	小	0.378	揖斐郡	
	3rd	市	0.322	恵那郡	
2nd *kanji*	1st	豊	0.826	大野郡	
	2nd	益	0.824	海津郡	
	3rd	婁	0.820	可児郡	
3rd *kanji*	1st	郡	0.911	加茂郡	
	2nd	稲	0.867	郡上郡	
	3rd	部	0.847	土岐郡	
				羽島郡	
				不破郡	
				益田郡	
				武儀郡	
				本巣郡	
				山県郡	
				養老郡	
				吉城郡	

II when called on by the Conflict Analyzer are shown. The first column of the table shows the branch module of CombNET-II to which each *kanji* category belongs. It can be seen that a *kanji* category may be in more than one branch module. All scores of each *kanji* used in the word candidates are shown, and MMVA takes the highest of them to calculate the total score for each word candidate.

For each word, CA produced the following scores.

word candidate	score
安城市	2.070
新城市	2.067

In Table 14, the character recognizer recognized only one kanji (郡) correctly when the city name of 揖斐郡 was input.

From Table 8, 17 city names in Gifu Prefecture that have this kanji form the list of word candidates. From this list, the Conflict Analyzer was able to find the correct interpretation.

6. Conclusions

A system for character-based handwritten word recognition was presented and a postprocessing system based on the Conflict Analyzing Model was proposed. From a sequence of handwritten *kanji* characters used in names of Japanese cities and a lexicon containing these names, an interpretation of the input name could be produced. The proposed system used the Modified Majority Voting Algorithm which is a fast method of generating a word interpretation, and in the case that a list of word candidates is produced in a situation where MMVA is not successful, the Conflict-Analyzing System is applied to produce the final word interpretation. The Conflict Analyzer is a robust system once it can find the correct interpretation, even if the character recognizer does not perform well. Handwritten address recognition can be performed effectively using by the structure of addressing, subdividing the set of all names in a set of small lexicons, and applying the proposed system, without the necessity of training the network when the lexicon is changed. Future work will be on the improvement of the vector quantizing process of the input vector space of the character recognizer.

References

[1] K. Hotta, N. Suzumura, A. Iwata, and Y. Ino, "Handwritten Japanese kanji character recognition by a structured neural network combnet-ii," Proc. International Conference of Artificial Neural Networks, IV, pp.228–234, 1992.
[2] L.D. Harmon, "Automatic recognition of print and script," Proc. IEEE, vol.60, pp.1165–1176, 1972.
[3] C.Y. Suen, S. Mori, and K. Yamamoto, "Historical review of ocr research and development," Proc. IEEE, vol.80, no.7, pp.1029–1058, 1992.
[4] M. Sabourin and A. Mitiche, "Optical character recognition by a neural network," Neural Networks, vol.5, pp.843–852, 1992.
[5] G. Nagy, "At frontier of ocr," Proc. IEEE, vol.80, no.7, pp.1093–1100, 1992.
[6] R. Hecht-Nielsen, "Neurocomputing," Addison-Wesley, 1989.
[7] S. Naito, N. Hagita, and I. Masuda, "Handprinted chinese character recognition by peripheral direction contributivity feature," IEICE Trans., vol.J66-D, no.10, pp.1185–1192, 1983.

Keiji Yamanaka received his B.E. degree in 1980 and M.E. degree in 1992 in electrical engineering from the Federal University of Uberlandia in Uberlandia City, Brazil. He is presently a doctoral student in the Department of Electrical and Computer Engineering at Nagoya Institute of Technology, Japan. His research interests include neuro-computing and pattern recognition.

Susumu Kuroyanagi received his B.E. degree in 1991, M.E. degree in 1993, and Ph.D. in 1996, all in electrical and computer engineering from Nagoya Institute of Technology. He is presently a research associate in the Department of Electrical and Computer Engineering, Nagoya Institute of Technology. His research interests include neurocomputing, application of neural networks, and modeling of the auditory nerve system. He is a member of the Acoustical Society of Japan and the Japanese Neural Network Society.

Akira Iwata received his B.S. degree in 1973, M.S. degree in 1975, and Ph.D. in 1981, all in electrical and electronics engineering from the Faculty of Engineering, Nagoya University, Nagoya, Japan. He is presently a professor in the Department of Electrical and Computer Engineering, Nagoya Institute of Technology. He was an Alexander Humboldt Foundation Research Fellow at Giessen University, Federal Republic of Germany, in 1982–83. His current interests are in the fields of neural network architecture, neurocomputing, application of neural networks, and collaboration system development. He is a member of Japanese Neural Network Society, Japan Society of Medical Electronics and Biological Engineering, the Information Processing Society of Japan, and the IEEE.