Correspondence

Integrated Segmentation and Recognition of Handwritten Numerals with Cascade Neural Network

Seong-Whan Lee and Sang-Yup Kim

***Abstract*—In this paper, we propose an integrated segmentation and recognition method using cascade neural network. In the proposed method, a new type of cascade neural network is developed to train the spatial dependences in connected handwritten numerals. This cascade neural network was originally extended from the multilayer feedforward neural network to improve the discrimination and generalization power.**

In order to verify the performance of the proposed method, recognition experiments with the National Institute of Standards and Technology (NIST) numeral databases have been performed. The experimental results reveal that the proposed method has higher discrimination and generalization power than the previous integrated segmentation and recognition (ISR) methods have. Moreover, the network-size of the proposed method is smaller than that of previous integrated segmentation and recognition methods.

***Index Terms*—Cascade neural network, handwritten character recognition, segmentation and recognition of numerals.**

I. Introduction

Over recent years, the area of pattern recognition has had a great deal of work done on it, and neural network models have been successfully applied to various applications of pattern recognition. Especially, multilayer feedforward neural network has shown good performance in recognizing handwritten characters with various writing styles and sizes [1]–[3].

However, as conventional methods usually involve a segmentation step prior to a recognition step, their recognition accuracy depends on the accuracy of the underlying segmentation algorithm. That is, the separation between segmentation and recognition becomes unreliable if characters are touching each other, touching bounding boxes, broken, or noisy. The problem in these cases is that a correct recognition requires making a correct segmentation, and also a correct segmentation often requires making a correct recognition. Therefore, researches for integrating segmentation and recognition have been in progress vigorously in recent years and some of them have shown promising results [4]–[8]. However, because most ISR methods can not train spatial dependencies in connected handwritten characters, they are inefficient for real data with various connection styles. A cascade neural network offer a framework suitable to train spatial dependencies of connected characters in handwritten numerals because these networks are able to encode, store, and process the context information about the input history of the network.

Manuscript received March 18, 1996; revised September 1, 1998. This work was supported by the Hallym Academy of Science, Hallym University and the Creative Research Initiatives of the Korea Ministry of Science and Technology. A preliminary version of this paper was presented at the Third International Conference on Document Analysis and Recognition, Montreal, P.Q., Canada, August 1995.

S.-W. Lee is with the National Creative Research Initiative Center for Artificial Vision Research, Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-701, Korea.

S.-Y. Kim is with Human Computer, Inc., Seoul, Korea.

Publisher Item Identifier S 1094-6977(99)02772-8.

In this paper, we propose an ISR method to recognize connected handwritten numerals with a cascade neural network. The main characteristics of the proposed method are the following:

1) segmentation and recognition is integrated within a single neural network;
2) spatial dependencies in connected handwritten characters can be trained.

The proposed method may look similar to that of Martin *et al.* [7] in some respects. However, it differs from the method of Martin *et al.* in that the network architecture of the proposed method is based on the cascade neural network which is able to retain the spatial relationships of connected characters in handwritten numerals.

In order to verify the performance of the proposed method, experiments with the National Institute of Standards and Technology (NIST) database have been carried out and the performance of the proposed method has been compared with that of previous ISR methods. The experimental results reveal that the proposed method has higher discrimination and generalization power than the previous ISR methods. Moreover, the network-size of the proposed method is smaller than that of the previous ISR methods.

II. Related Works

In recent years, a number of methods have been developed to segment and recognize handwritten numerals [4]–[9]. They all work by generating multiple candidate segments and sending each possibility to a classifier. The final recognition decision is based on the segmentation that produces the strongest classification response. In this section, we review several handwritten numeral recognition methods.

Matan *et al.* [4] built a ZIP code reading system based on the vertical-cut segmentation. The system generates the candidate segments by analyzing the vertical projection of the image. Each candidate segment is sent to the recognizer, and the segmentation that results in the best combined score is chosen. In this method, they reduced the number of candidate segments by analyzing connected components.

Keeler and Rumelhart [5] proposed a neural network that simultaneously segments and recognizes connected characters in an integrated system, called self-organizing integrated segmentation and recognition (SOISR). This neural network uses a backpropagation algorithm, and is able to take position-independent information as targets and self-organizes the activities of the units in a competitive fashion to infer the positional information.

Matan *et al.* [6] proposed another method based on space displacement neural network (SDNN). This is an extension of a backpropagation learning neural network, which recognizes isolated handwritten digits [2]. Because most of ISR methods replicate recognizer at all possible locations across the input, it is redundant to process the same pixels numerous times as part of different, overlapping candidate segments. In order to solve this problem, they proposed a method which passes a whole size-normalized image to the recognition system and segments a feature map, after most of the neural network computation has been done. The novelty in this method is that segmentation is done on the feature maps developed by the SDNN rather than the input space.

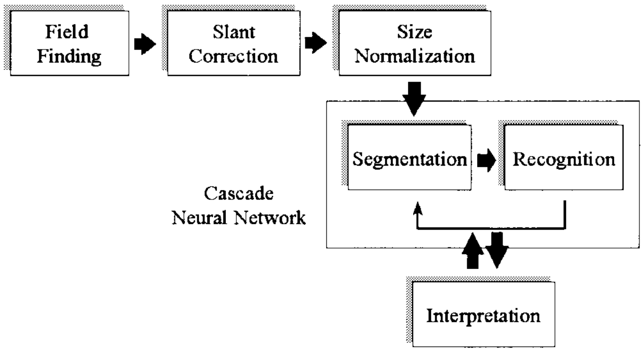

Fig. 1. Schematic of the proposed method.

Martin *et al.* [7] proposed two methods for integrating handwritten character segmentation and recognition within one system, called centered object integrated segmentation and recognition (COISR). Two methods are exhaustive scan and saccadic scan. The exhaustive scan method is that a backpropagation learning neural network exhaustively scans a character field, and is trained to recognize whether its input window is centered over a single character or between characters. When its input window is centered on the character, the network classifies the character. This method yields high accuracy, but suffers from the common weakness of generating too many candidate segments to be efficient. In order to overcome this weakness, the saccadic scan method was developed. For saccadic scan method, the network is trained not only to recognize whether a character is centered on its input window, but also to compute ballistic "eye" movements that enable the input window to jump from one character to the next.

Fujisawa [9] analyzed the stroke shapes of touching pattern and proposed a recognition-based segmentation method for character segmentation and recognition. This is a pattern-oriented segmentation method. Connected handwritten components are extracted instead of a pixel image, and spatial interrelations between components are measured to group them into meaningful character patterns. Stroke shapes are analyzed in case of touching characters. Connected numerals can be separated by using a method of finding the touching position. Ambiguities are handled by multiple hypotheses and verification with recognizer.

III. ISR Using Cascade Neural Network

A. Overview of the Proposed Method

The overview of the proposed method for recognizing connected handwritten numerals is shown in Fig. 1. Preprocessors transform a field into a more usable form by techniques such as slant correction and size normalization. Then, as the input window scans the field, the field is presegmented and the cascade neural network determines whether or not the segmentation is correct. If the segmentation is correct, the network classifies what is centered in the input window.

B. Preprocessing

In order to process the handwritten numerals efficiently, they must be transferred into a more appropriate form by preprocessing. The bounding box of the handwritten numeral is first removed using projection profile. Then the slant correction technique [10] is applied to the handwritten numeral. Each handwritten numeral image is size-normalized [11], with respect to the vertical axis, to the height of 16 pixels keeping the aspect ratio. The images of handwritten numerals for training are also labeled with the horizontal center positions of each character in the field image, as well as with the categories of the characters. This makes it possible to pair each input window with a target output node representing what is centered in the input window (Fig. 2).

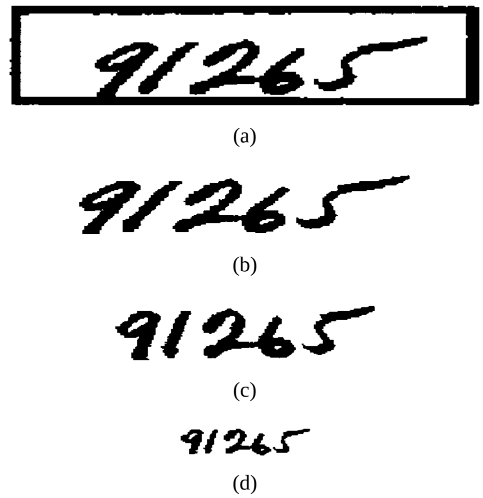

Fig. 2. Example of the preprocessing stages: (a) original character field image, (b) box removal, (c) slant correction, and (d) size normalization.

C. Network Architecture

The architecture of the proposed cascade neural network is shown in Fig. 3. The proposed neural network has three layers. Input and hidden layers are divided into four subnetworks. The input layer of each subnetwork consists of two parts. One is input units which are used as input values, and the other is context units in which the activation values of the previous subnetwork is used as input values. All units of the input layer of the each subnetwork are fully connected to the units in the hidden layer of the corresponding subnetwork. The activation values of the hidden units of each subnetwork are fed into the input units (context units) of its right subnetwork, and used to compute the activation values of the hidden units in its right subnetwork. Therefore, the activation values of the current subnetwork are computed by using the input history of the previous subnetwork as well as the current input. Thus, the information of the spatial dependencies in the input image can be encoded into the network.

The proposed network views the input field on 16×16 window at a time. In order to encode the spatial dependencies of the input pattern, 16×16 input frame window is divided into four local frames. Each subnetwork consists of 124 ($4 \times 16 + 60$) units because the inputs are 4×16 local features for input frame and 60 context units which contain a sequence of features in the past frames. In each subnetwork, all units of the input layer are fully connected to the units in the hidden layer. After each subnetwork processed through time delay, hidden units in all subnetworks are fully connected to output layer. The output layer is composed of 11 output units: one per each class and one noncharacter unit corresponding to the state when the input window is centered between characters or over a blank space.

D. Training

The network is trained on the pattern appeared in the center of the input window and a desired output value representing what is in the center of the input window. The input to the network comes from a two-dimensional input window that scans horizontally across the field image (Fig. 4). It is based on the sliding window concept

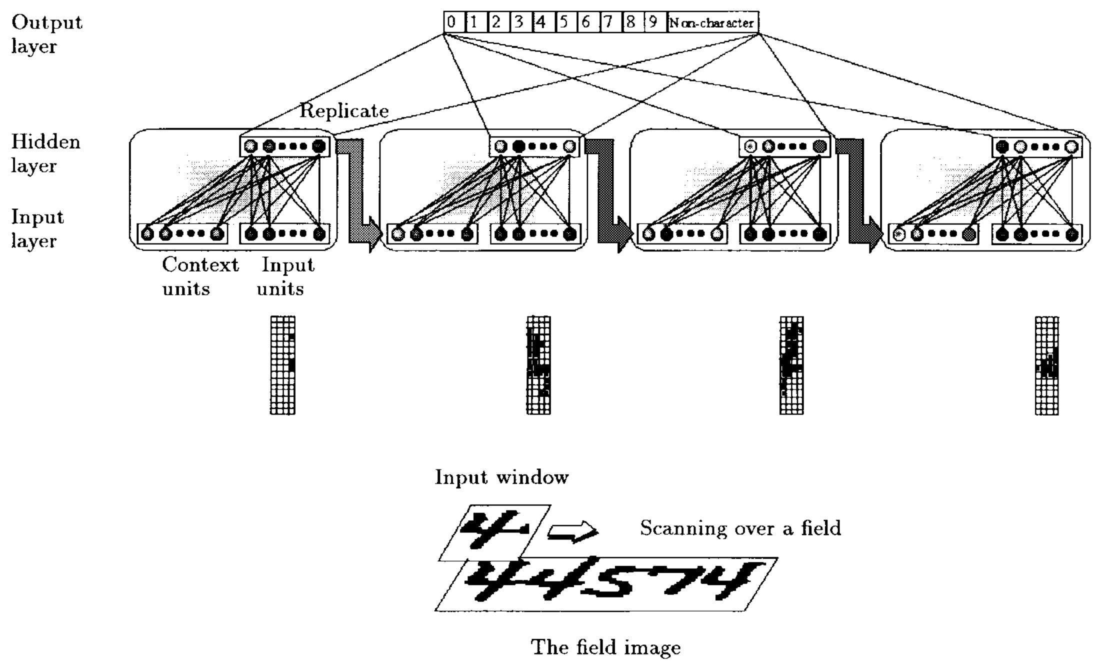

Fig. 3. Network architecture of the proposed cascade neural network.

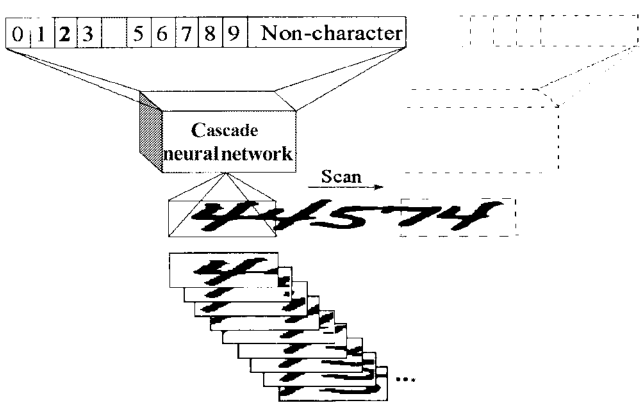

Fig. 4. Scanning over a field.

used in neural network at speech applications, such as time delay neural networks [12]. The input window scans the field at the rate of two pixels. As the center positions of the characters in the field are manually labeled, the center positions are not definitely precise and the scan rate of two pixels is enough. The content of the input window is paired with a target output vector representing what is centered in the input window at each stopping point.

During the forward pass, each subnetwork is processed from left to right. The pattern appeared in the input window is divided into four frames. Initially, the context units of the first subnetwork are initialized to zero. Then, each frame is presented at input units of each subnetwork and results of hidden units are fed into the context units of the next subnetwork. Because the activation values of hidden units are reused in computing the activation values of hidden units of the next subnetwork, the spatial dependencies of each frame are encoded in the network. At the end of a pattern appeared in the input window, the results of all hidden units are fed into the output layer.

All the connections of the network are adaptive, and are trained with a backpropagation learning algorithm [13]. The transfer function at each unit is the familiar sigmoid function as follows:

$$O = \frac{1}{1 + e^{-(\Sigma WX + \text{bias})}} \tag{1}$$

where, O is the output of nonlinear transfer function, W is weight, and X is input to the unit neuron. W and bias are randomly selected in a start state and modified during training. The modification of weights is performed as follows:

$$\Delta W(n) = -\epsilon \partial E / \partial W + \alpha \Delta W(n-1) \tag{2}$$

where, n is the training index, ϵ is learning rate, α is momentum, and E is mean squared error (MSE).

It is also important to determine when the network finishes the training. Over training can cause the network to over-specialize training set and to adapt too closely to the particular features of the training set. This problem is solved by using a validation set. After training the network for a short time, performance of the network is tested on the validation set. The training is repeated until the recognition rate on the validation set starts to deteriorate.

E. Recognition

In recognizing the field image, the input window scans exhaustively across the field image at the rate of two pixels. The activation value of each output unit represents what is centered in the input window. When the input window is centered on a character, the output unit corresponding to that character has a high activation value, and others have a low activation value. When the input window is centered between characters or over a blank space, noncharacter class unit has a high activation value. As the input window scans across the field, the activation values of output units have a trapezoidal shape.

The activation values of the output units make a trace as shown in Fig. 5 as the network scans across the field image. This output trace is parsed to determine an ASCII string corresponding to the digits in the field. When the activation value of noncharacter class unit falls below a threshold, the activation values of each output category are summed until the activation value of noncharacter class unit again exceeds the threshold. This drop and subsequent rise in

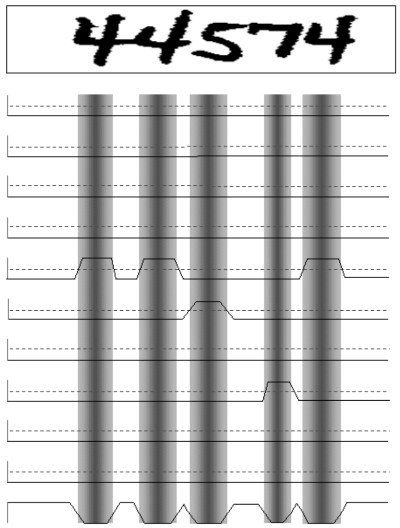

Fig. 5. Activation values of the output units as the network scans the field horizontally over time.

the activation value of noncharacter class unit signifies that the input window has just passed over a character. The recognition system classifies the character by determining which output unit of character category has the highest summed activation value. Then, the sum of each output value is re-initialized to zero, and the process continues as the horizontal scan.

The recognition results are reported in terms of field-based error rates. If the network misclassifies one character in a field, the entire field is considered as misclassified. Error rates pertain to the fields remaining, after rejection. Rejections are based on placing a threshold for the acceptable distance between the highest and the next highest running activation value.

IV. Experimental Results and Analysis

In this section, we present experimental results and the performance of the proposed method is compared with that of previous ISR methods. Experiments have been carried out on Cray Y-MP supercomputer.

A. Database

We tested the performance of the proposed method on the set of handwritten numerals collected and distributed by the NIST. This database contains approximately 273 000 samples of handwritten numerals. Each of 2,100 Census workers filled in the form. There were 50 different forms used in the study, each with 33 fields, 28 of which contain handwritten numerals raging in length from 2–10 digits per field. The scanning resolution of the samples was 300 pixels/in. We used training set, which consists of about 80 000 digits from 20 000 fields and testing set, which consists of about 20 000 digits from 5,000 fields. The training data fields were labeled with the horizontal center positions of each character in the field, as well as with the categories of the character. The horizontal center positions of each character in the training data fields were manually extracted.

B. Experiment 1: Vertical Nonlinear Normalization

Variations in height and position of the numerals are not suitable for training all spatial dependencies in connected handwritten numerals. If the numerals in a field are similar in height, they can be located in a similar position at the input sliding window of the system. In order

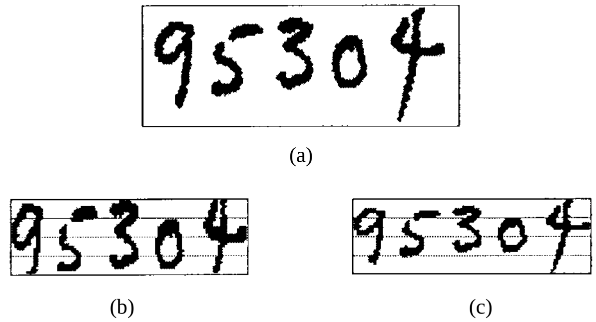

Fig. 6. Example of vertical normalization: (a) input image, (b) linear normalization, and (c) vertical nonlinear normalization.

TABLE I
Field-Based Recognition Rates of the Linear Normalization Method and Nonlinear Normalization Method. Training With 2500 Fields and Testing with 496 Fields

Field size	Linear normalization	Nonlinear normalization
2-digits	71.43%	72.45%
3-digits	67.68%	73.74%
4-digits	69.00%	73.00%
5-digits	61.62%	65.57%
6-digits	59.00%	64.03%

TABLE II
Field-Based Recognition Rates of the Proposed Method and the Fujisawa's Method

Field size	Fujisawa's method	Proposed method
2-digits	89.79%	95.23%
3-digits	84.64%	88.01%
4-digits	80.63%	80.69%
5-digits	76.05%	78.61%
6-digits	74.54%	70.49%

to absorb the vertical variations of numerals, we applied nonlinear normalization method based on dot density [14] to input data with respect to vertical axis. Through vertical nonlinear normalization, the numerals can be centered at input sliding window in the proposed method. In Fig. 6, it is evident that the vertically nonlinear normalized numerals are more regular in height than the linear normalized ones. The recognition rates for the linearly normalized data and nonlinearly normalized data are also compared in Table I.

As shown in Table I, the vertical nonlinear normalization method is more powerful than linear normalization method.

C. Experiment 2: ISR versus Non-ISR

We also made an experiment to compare the ISR method with a non-ISR method of Fujisawa's [9]. The Fujisawa's method segments the connected handwritten numerals into primitive segments. Each primitive segment is a subimage of the original handwritten numerals and ideally consists of either a numeral or a part of a numeral. The recognition process decides the best numeral string by verification using recognizer. The recognition accuracy of this system depends on the accuracy of the single digit recognizer. We used the Multilayer Cluster Neural Network [15] as the single digit recognizer. The recognition rates of the Fujisawa's method and the proposed method are are compared in Table II.

D. Experiment 3: NIST Numeral Database

In order to verify the performance of the proposed method, we made experiments with NIST numeral database. Table III shows the field-based rejection rates for 1% field-based error rates.

TABLE III
FIELD-BASED REJECTION RATES

Field size	Error rate	Rejection rate
2-digits	1.0%	3.8%
3-digits	1.0%	11.1%
4-digits	1.0%	18.5%
5-digits	1.0%	20.6%
6-digits	1.0%	28.8%

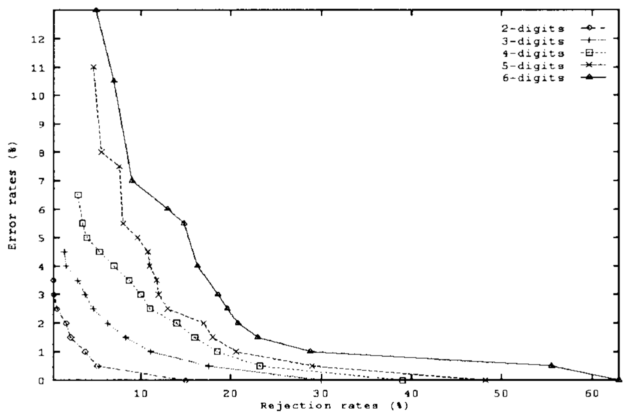

Fig. 7. Error rates versus rejection rates of the proposed method.

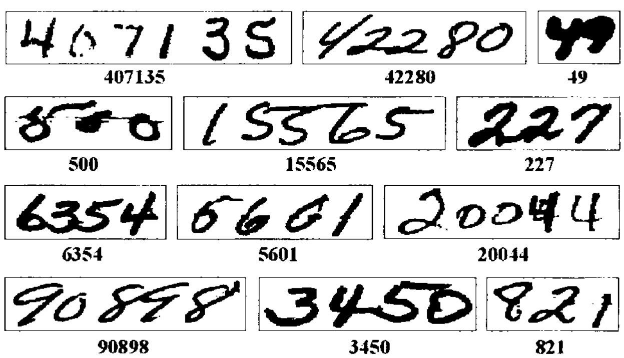

Fig. 8. Examples of correctly recognized fields.

As shown in Table III, for the testing set, the rejection rate is 20.6% with an accuracy rate of 99% overall on fields of length 5. We have also analyzed the error rates versus rejection rates to evaluate the discrimination performance of the proposed method on testing set. The results are shown in Fig. 7. Fig. 8 shows the examples of fields containing touching or broken characters that were correctly recognized by the proposed method, and Fig. 9 shows the examples of misclassified fields.

E. Discussion

In order to demonstrate the performance of the proposed method objectively, five kinds of ISR methods have been compared, in which SDNN used USPS database and the others used NIST numeral database. Tables IV and V show the field-based error rates on fields of length 5 and network complexities of each method, respectively.

As shown in Table IV, the rejection rate of the proposed method is 20.6% with an error rate of 1.0% on fields of length 5. However, for the SOISR method and the exhaustive scan method on the same NIST numeral database, the rejection rate is 28.0 and 23.4%, respectively. And, for the saccadic scan method, the rejection rate is 23.2% with the

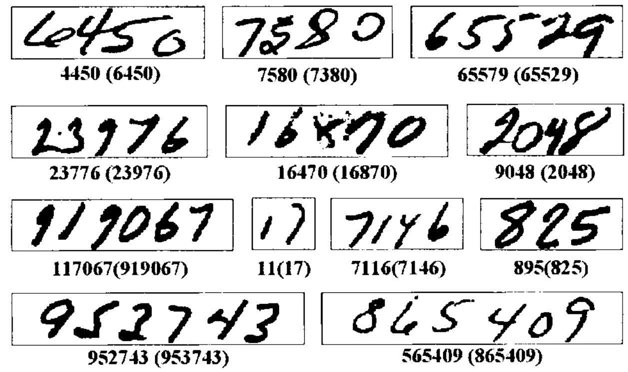

Fig. 9. Examples of misclassified fields.

TABLE IV
FIELD-BASED ERROR RATES OF EACH METHOD

Method	Rejection rate	Error rate	Training data	Test data
SOISR [5]	28.0%	1.0 %	NA	5,000
SDNN [6]	0%	33.7 %	6,000	3,000
Exhaustive scan [7]	23.4%	1.0 %	20,000	5,000
Saccadic scan [7]	23.2%	1.1 %	20,000	5,000
Proposed method	20.6%	1.0 %	20,000	5,000

TABLE V
NETWORK COMPLEXITIES OF EACH METHOD

Method	# of nodes	# of connections	# of weights
SOISR [5]	2,532	130,680	5,508
	1,920	107,424	6,552
SDNN [6]	4,634	94,592	2,536
Exhaustive scan [7]	2,927	157,068	61,164
Saccadic scan [7]	2,947	160,668	64,764
Proposed method	747	33,400	33,400

error rate of 1.1%. These results illustrate that the proposed method has much better discrimination power than the other ISR methods have.

In order to compare the efficiency of several ISR methods, we compared the size of networks and the number of weights. As shown in Table V, the size of the proposed network is much smaller than that of other ISR methods.

V. CONCLUSION

As conventional methods for recognizing connected handwritten characters usually involve a segmentation step prior to a recognition step, their recognition accuracy depends on the accuracy of the underlying segmentation algorithm. That is, the separation between segmentation and recognition becomes unreliable where the characters are touching each other, touching bounding boxes, broken, or noisy. Therefore, in order to overcome this problem, researches for integrating segmentation and recognition have been in progress vigorously. However, most ISR methods based on the feedforward neural network can not train spatial dependencies in connected handwritten characters, they are inefficient for real data with various connection styles.

In this paper, we proposed an ISR method to recognize handwritten numerals with cascade neural network. The proposed method offers a framework suitable to train spatial dependencies in connected handwritten numerals because the network is able to encode, store, and process the context information about the input history of the network.

In order to verify the performance of the proposed method, experiments with the NIST numeral database have been carried out and the performance of the proposed method has been compared with that of the previous ISR methods. The experimental results revealed that the proposed method had much better discrimination and generalization power than the previous ISR methods.

REFERENCES

[1] J. S. Denker *et al.*, "Neural network recognizer for handwritten ZIP code digits," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, vol. 1, pp. 323–331.

[2] Y. Le Cun *et al.*, "Constrained neural network for unconstrained handwritten digit recognition," in *Proc. 1st Int. Workshop Frontiers Handwriting Recogn.*, Montreal, P.Q., Canada, Apr. 1990, pp. 145–154.

[3] G. L. Martin and J. A. Pittman, "Recognizing hand-printed letters and digits," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 405–414.

[4] O. Matan *et al.*, "Reading handwritten digits: A ZIP code recognition system," *IEEE Comput. Mag.*, vol. 25, no. 7, pp. 59–63, July 1992.

[5] J. Keeler and D. E. Rumelhart, "A self-organizing integrated segmentation and recognition neural network," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 496–503.

[6] O. Matan, J. C. Burges, Y. Le Cun, and J. S. Denker, "Multi-digit recognition using a space displacement neural network," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 488–495.

[7] G. L. Martin, M. Rashid, and J. A. Pittman, "Integrated segmentation and recognition through exhaustive scans or learned saccadic jumps," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 831–847, 1993.

[8] K. Imai, K. Gouhara, and Y. Uchikawa, "Recognition of letters in lateral printed strings using a three-layered BP model with feedback connections," *Syst. Comput. Jpn.*, vol. 23, no. 12, pp. 83–92, 1992.

[9] H. Fujisawa and Y. Nakano, "Segmentation methods for character recognition: From segmentation to document structure analysis," *Proc. IEEE*, vol. 80, pp. 1079–1092, July 1992.

[10] R. M. Božinovič and S. N. Srihari, "Off-line cursive word recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 68–83, Jan. 1989.

[11] S.-W. Lee and J. S. Park, "Nonlinear shape normalization methods for the recognition of large-set handwritten characters," *Pattern Recognit.*, vol. 27, no. 7, pp. 895–902, 1994.

[12] K. Lang, A. Waibel, and G. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, pp. 23–43, 1990.

[13] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. Rumelhart and J. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[14] Y. Yamashita, K. Higuchi, Y. Yamada, and Y. Haga, "Classification of handprinted Kanji characters by the structured segment matching method, *Pattern Recognit. Lett.*, vol. 1, pp. 475–479, 1983.

[15] S.-W. Lee, "Off-line recognition of totally unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 648–652, June 1996.

Analytical Analysis and Feedback Linearization Tracking Control of the General Takagi-Sugeno Fuzzy Dynamic Systems

Hao Ying

***Abstract*—Takagi–Sugeno (TS) fuzzy modeling technique, a black-box discrete-time approach for system identification, has widely been used to model behaviors of complex dynamic systems. Analytical structure of TS fuzzy models, however, is presently unknown, nor is its possible connection with the traditional models, causing at least two major problems. First, the fuzzy models can hardly be utilized to design controllers for control of the physical systems modeled. Second, there lacks a systematic technique for designing a controller capable of controlling any given TS fuzzy model to achieve desired tracking or setpoint control performance. In this paper, we provide solutions to these problems. First of all, we have proved that a general class of TS fuzzy models is nonlinear time-varying Auto-Regressive with the eXtra input (ARX) model. The fuzzy models in this study are general because they use arbitrary continuous input fuzzy sets, any types of fuzzy logic AND operators, TS fuzzy rules with linear consequent and the generalized defuzzifier which contains the popular centroid defuzzifier as a special case. Furthermore, we have established a simple necessary and sufficient condition for analytically determining local stability of the general TS fuzzy dynamic models. The condition can also be used to analytically check quality of a TS fuzzy model and invalidate the model if the condition warrants. More importantly, we have developed a feedback linearization technique for systematically designing an output tracking controller so that output of a controlled TS fuzzy system, which may or may not be stable, of the general class achieves perfect tracking of any bounded time-varying trajectory. We have investigated stability of the tracking controller and established a necessary and sufficient condition, in relation to stability of nonminimum phase systems, for analytically deciding whether a stable tracking controller can be designed using our method for any given TS fuzzy system. Three numerical examples are provided to illustrate the effectiveness and utility of our results and techniques.**

***Index Terms*—AR models, feedback linearization, fuzzy control, fuzzy modeling, stability.**

I. INTRODUCTION

Numerous successful industrial applications have shown the power of Takagi–Sugeno (TS) fuzzy modeling approach [10], which is a black-box discrete-time modeling approach developed for modeling complex dynamic systems [1], [5], [16], [17], [19]. Compared with the conventional black-box modeling techniques [7] that can only utilize numerical data, TS modeling approach allows one to take advantage of both qualitative and quantitative information [8]. This advantage is practically important and even crucial in many circumstances. Qualitative information, such as expert/operator knowledge and experience about a physical system to be modeled, can readily be incorporated into TS fuzzy models in the form of fuzzy sets, fuzzy logic, or fuzzy rules. Virtually all the TS fuzzy models in the literature use linear functions of input variables as consequent of the fuzzy rules. Many learning schemes have been developed to automatically configure one or more components of TS fuzzy models so that a TS fuzzy model can quickly be established when qualitative/quantitative information is available [6], [15]. Despite of

Manuscript received November 10, 1996; revised September 17, 1998. This work was supported in part by Texas Higher Education Coordinating Board Grant 004952-054.

The author is with the Department of Physiology and Biophysics, Biomedical Engineering Center, The University of Texas Medical Branch, Galveston, TX 77555-0456 USA.

Publisher Item Identifier S 1094-6977(99)02771-6.