Invited Paper

Recognition of Handwritten Chinese Characters by Self-growing Probabilistic Decision-based Neural Networks*

Hsin-Chia Fu and Y. Y. Xu

Department of Computer Science and Information Engineering,
National Chiao Tung University,
Hsin-Chu, Taiwan, ROC.

ABSTRACT

In this paper, we present a Bayesian decision-based neural networks (BDNN) for handwritten Chinese character recognition. The proposed **Self-growing Probabilistic Decision-based Neural Networks** (SPDNN) adopts a hierarchical network structure with nonlinear basis functions and a competitive credit-assignment scheme. Our prototype system demonstrates a successful utilization of SPDNN to the handwriting of Chinese character recognition on the public databases, CCL/HCCR1 and in house database (NCTU/NNL). Regarding the *performance*, experiments on three different databases all demonstrated high recognition (86~94%) accuracy as well as low rejection/acceptance (6.7%) rates. As to the processing speed, the whole recognition process (including image preprocessing, feature extraction, and recognition) consumes approximately 0.27 second/character on a Pentium-100 based personal computer, without using hardware accelerator or co-processor.

Keywords: B ayesian Decision-based Neural Networks, Self-growing Probabilistic Decision-based Neural Networks, Supervised learning, Optical Character Recognition.

1. INTRODUCTION

In recent years, there is a significant increase in the electronic management of information by multimedia information systems. The handling of multimedia documents consists of the editing and display of texts, graphics, images, and handwriting. Currently, the keyboard and the mouse are still the dominant input devices for personal computer based multimedia systems. However, in preparing a first draft and concentrating on content creation, pencil and paper are often superior to keyboard entry. By incorporating character recognition with a text-to-speech technology, converting handwriting directly to voice will be an interesting multimedia application.

1.1. Overview of Handwriting Recognition

The machine recognition of characters has been a topic of intense research since the 1960s.[1–4,11,14–16,19] After more than 30 years of rigorous attacks, studies in the field of handwriting recognition remain as active as ever. A comprehensive set of references to recent research and developments in *handwriting recognition* can be found in two special issues of the *International Journal of Pattern Recognition and Artificial intelligence (Vols. 1 and 2, 1991)* and *Proceedings of IEEE (July 1992)*. These two issues describe not only the state of the art but also, in many cases, ongoing research efforts. To search for the most updated activities, achievements as well as demonstrations on handwriting recognition, the World Wide Web (WWW) seems an excellent medium for the researcher and in fact for everyone interested.

Current Status and Related Problems

It is well known that Chinese characters (including Japanese Kanji) are unique and different from that of Western languages such as English, in that they are non-alphabetic and have quite complicated stroke structures. Such properties make Chinese character recognition by computer particularly difficult and challenging.

Early work on machine recognition of handwritten Chinese character (HCC) research was originated in Japan –over 25 years ago.[23,17] Since then, hundreds of paper have been appeared on this topic, which assumes both scientific and commercial importance.

*This research was supported in part by the National Science Council under Grant NSC 86-2213-E009-099.
Send correspondence to H. C. Fu: E-mail: hcfu@csie.nctu.edu.tw

Due to the large character set† in Chinese language, until 1988 Shyr et al., built a prototype recognition system for the 5401 printed Chinese characters (PCC) and successfully demonstrated the system with a recognition rate of 95.6%.[18] Four years later (1992), the first commercial PCC recognition machine, based on a 386 personal computer, was announced, which is capable of recognizing four types (Kai, Ming, Black, and Sung) of 5401 characters with an accuracy of 95% at a speed of 7 characters per second. Thus, machine recognition of Chinese characters is by no means a pure research topic. Nevertheless, the machine recognition of HCC is still considered to be a very difficult problem due to the following aspects:

1. large number of characters set
2. complexity of character structures
3. wide variation in personal handwriting
4. many similar characters

Recently, probabilistic decision-based neural networks (PDBNN) which has the merits of both neural network and statistical approaches was proposed to attack face and other biometrics recognition systems.[13] The modular structure in the PDBNN devotes one of its subnets to the representation of a particular person's face images. In each subnets, the discriminate function of a PDBNN is in a form of fixed number of mixture of Gaussian distributions. This yields extremely low false acceptance and rejection rates for the face recognition systems.

However, since the stroke complexity of Chinese characters varies from one stroke up to a few dozens of strokes. Hence, fixed number of mixture of Gaussian distributions is not suitable for the representation of character pixel distribution. In this paper, we propose a new PDBNN, called Self-growing Probabilistic Decision-based Neural Networks (SPDNN) to recognize Chinese characters. For different characters, the discriminate function of a SPDNN is in a form of flexible number of mixture of Gaussian distributions.

This paper is organized as follows: Section 2 presents the mathematical background, the architecture, and the learning rules of the SPDNN. Then, a multistage handwriting recognition system is presented in Section 3. The proposed system consists of three modules, which are all implemented by the SPDNN. A personal adaptive module is recommended to further improve the recognition performance. Experimental results of these modules are provided and discussed in each section.

2. SELF-GROWING PROBABILISTIC DECISION-BASED NEURAL NETWORK

Self-growing Probabilistic Decision-based Neural Networks (SPDNN) is a probabilistic variant of decision-based modular neural network[10] for classification. One subnet of an SPDNN is designed to represent one object class. There are two properties of the SPDNN learning rules. The first one is the decision-based learning rules. Based on the teacher information which only tells the correctness of the classification for each training pattern, an SPDNN performs a distributed and localized updating rules. The updating rule applies *reinforced learning* to a subnet corresponding to the correct class and *antireinforced learning* to the (unduly) winning subnets.

The second property is the Iteratively Supervised Learning and Unsupervised Growing (*ISLUG*). There are two learning phases in this scheme: After each subnet is initialized with one cluster (see Section 2.2.1) or is self-grown with a new cluster (Section 2.2.2), the system enters the supervised learning (SL) phase. In the SL phase, teacher information is used to reinforce or antireinforce the decision boundaries obtained during the initialization or self-growing stages. When the supervised training progress becomes very slow or is trapped in a paralysis state, yet the classification or recognition accuracy is not at a satisfied level, the training enters the Unsupervised growing (UG) phase. In the UG phase, an SPDNN creates a new cluster in a subnet according to the proposed *self-growing* rule. Thereafter, the training enters the Supervised Learning phase again. The ISLUG learning procedure terminates when the training accuracy reaches a predefined satisfaction level. The detailed description of the SPDNN model is given in the following sections.

†There are more than 40,000 Chinese characters, and 5401 characters are used frequently in daily life.

2.1. Discriminant Functions of SPDNN

One of the major differences between PDBNN[13] and SPDNN is that SPDNN extends the fixed number of clusters in PDBNN to flexible number of clusters. That is, the subnet discriminant functions of SPDNN are designed to model the log-likelihood functions of different complexed pixel distribution of handwritten characters. Thus, *reinforced* or *antireinforced* learning is applied to **all** the subnets of the global winner and the supposed (i.e., the correct) winner, with a weighting distribution proportional to the degree of possible involvement (measured by the likelihood) by each subnet. Given a set of iid patterns $\mathbf{X}^+ = \{x(t);\ t = 1, 2, \cdots, N\}$, we assume that the likelihood function $p(\mathbf{x}(t) \mid \omega_i)$ for class ω_i (i.e., a character class) is a mixture of Gaussian distributions.

Define $p(x(t) \mid \omega_i, \Theta_{r_i})$ to be one of the Gaussian distributions which comprise $p(x(t) \mid \omega_i)$, where Θ_{r_i} represents the parameter set $\{\mu_{r_i}, \mathbf{\Sigma}_{r_i}\}$ for a cluster r_i in subnet i.

$$p(\mathbf{x}(t) \mid \omega_i) = \sum_{r_i=1}^{R_i} P(\Theta_{r_i} \mid \omega_i) p(\mathbf{x}(t) \mid \omega_i, \Theta_{r_i}),$$

where $P(\Theta_{r_i} \mid \omega_i)$ denotes the prior probability of the cluster r_i. By definition, $\sum_{r_i=1}^{R_i} P(\Theta_{r_i} \mid \omega_i)$=1, where R_i is the number of clusters in ω_i.

The discriminate function of the multiclass SPDNN models the log-likelihood function

$$\phi(\mathbf{x}(t), \mathbf{w}_i) = \log\, p(\mathbf{x}(t) \mid \omega_i) = \log[\sum_{r_i=1}^{R_i} P(\Theta_{r_i} \mid \omega_i) p(\mathbf{x}(t) \mid \omega_i, \Theta_{r_i})], \tag{1}$$

where $\mathbf{w}_i = \{\mu_{r_i}, \mathbf{\Sigma}_{r_i}, P(\Theta_{r_i} \mid \omega_i), T_i\}$. T_i is the output threshold of the subnet i.

In most general formulation, the basis function of a cluster should be able to approximate the Gaussian distribution with full rank covariance matrix, i.e., $\phi(\mathbf{x}, \omega_i)$ = -$\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}_{r_i}^{-1} \mathbf{x}$, where $\mathbf{\Sigma}_{r_i}$ is the covariance matrix. However, for those applications which deal with high-dimension data but finite number of training patterns, the training performance and storage space discourage such matrix modeling. A natural simplifying assumption is to assume uncorrelated features of unequal importance. That is, suppose that $p(\mathbf{x}(t) \mid \omega_i, \Theta_{r_i})$ is a D-dimensional Gaussian distribution with uncorrelated features

$$p(\mathbf{x}(t) \mid \omega_i, \Theta_{r_i}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{\Sigma}_{r_i}|^{\frac{1}{2}}} \cdot \exp\left[-\frac{1}{2} \frac{(\mathbf{x}(t) - \mu_{r_i})^T (\mathbf{x}(t) - \mu_{r_i})}{\mathbf{\Sigma}_{r_i}} \right] \tag{2}$$

where $\mathbf{x}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \cdots, \mathbf{x}_D(t)]^T$ is the input pattern, $\mu_{r_i} = [\mu_{r_i1}, \mu_{r_i2}, \cdots, \mu_{r_iD}]^T$ is the mean vector, and diagonal matrix $\mathbf{\Sigma}_{r_i} = diag[\sigma^2_{r_i1}, \sigma^2_{r_i2}, \cdots, \sigma^2_{r_iD}]$ is the covariance matrix. As shown in Figure 1, an SPDNN contains K subnets which are used to represent a K-category classification problem. Inside each subnet, an elliptic basis functions (EBF's) is used to serve as the basis function for each cluster r_i:

$$\psi(\mathbf{x}(t), \omega_i, \Theta_{r_i}) = -\frac{1}{2} \sum_{d=1}^{D} \alpha_{r_id}(x_d(t) - \mu_{r_id})^2 + \theta_{r_i} \tag{3}$$

where $\theta_{r_i} = -\frac{D}{2} \ln 2\pi + \frac{1}{2} \sum_{d=1}^{D} \ln \alpha_{r_id}$. After passing an exponential activation function, $\exp\{\psi(\mathbf{x}(t), \omega_i, \Theta_{r_i})\}$ can be viewed as a Gaussian distribution, as described in (2), except a minor notational change: $\frac{1}{\alpha_{r_id}} = \sigma^2_{r_id}$.

2.2. Learning Rules for SPDNN

Recall that the training scheme for a multiclass SPDNN follows the *ISLUG* principle. The ISLUG training scheme contains the following two phases: *Supervised Learning* and *Unsupervised Growing.*

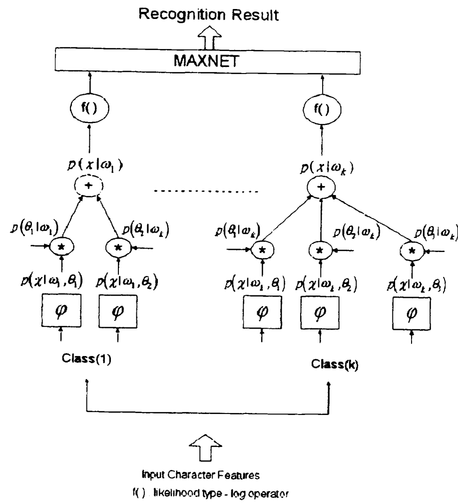

Figure 1. Schematic diagram of a *k-class* SPDNN character recognizer.

2.2.1. Supervised Learning in each subnet

Since the number of cluster in a subnet of an SPDNN can be adjusted in the Unsupervised growing phase, thus each subnet is initialized with one cluster. The values of the parameters (mean and covariance) of a cluster in each subnet are initialized at the beginning of the first Supervise Learning phase. Suppose that $\mathbf{X}_i^+ = \{\mathbf{x}_i(1), \cdots, \mathbf{x}_i(M_i)\}$ is a set of given training characters, which correspond to one of the L classes $\{\omega_i, i = 1, \cdots, L\}$, the mean μ_i and covariance Σ_i of the initial cluster in subnet i can be calculated as

$$\mu_i = \frac{1}{M_i} \sum_{m=1}^{M_i} \mathbf{x}_i(m) \tag{4}$$

$$\sigma_i = \frac{1}{M_i - 1} \sum_{m=1}^{M_i} (\mathbf{x}_i(m) - \mu_i)(\mathbf{x}_i(m) - \mu_i)^T. \tag{5}$$

During the **supervised learning** phase, training data is then used to fine tune the decision boundaries of each classes. The data adaptive scheme of the Supervised Learning for the multiclass SPDNN is the extension of the GS learning in.[13] Each class is modeled by a subnet with discriminant functions, $\phi(\mathbf{x}(t), \mathbf{w}_i), i = 1, 2, \cdots, L$. At the beginning of each Supervised Learning phase, use the still-under-training SPDNN to classify all the training characters $\mathbf{X}_i^+ = \{\mathbf{x}_i(1), \mathbf{x}_i(2), \cdots, \mathbf{x}_i(M_i)\}$ for $i = 1, \cdots, L$. $\mathbf{x}_i(m)$ is classified to class ω_i, if $\phi(\mathbf{x}_i(m), \mathbf{w}_i) > \phi(\mathbf{x}_i(m), \mathbf{w}_k)$, $\forall k \neq i$, and $\phi(\mathbf{x}_i(m), \mathbf{w}_i) \geq T_i$, where T_i is the output threshold for subnet i. According to the classification results, the training characters for each class i can be divided into three subsets:

- $D_1^i = \{\mathbf{x}_i(m); \mathbf{x}_i(m) \in \omega_i, \mathbf{x}_i(m)$ is classified to ω_i (correctly classified set) $\}$;
- $D_2^i = \{\mathbf{x}_i(m); \mathbf{x}_i(m) \in \omega_i, \mathbf{x}_i(m)$ is misclassified to other class ω_j (false rejection set) $\}$;
- $D_3^i = \{\mathbf{x}_i(m); \mathbf{x}_i(m) \notin \omega_i, \mathbf{x}_i(m)$ is misclassified to class ω_i (false acceptance set) $\}$.

The following reinforced and antireinforced learning rules[10] are applied to the corresponding subnets.

$$\text{Reinforced Learning}: \mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \phi(\mathbf{x}_i(m), \mathbf{w}_i) \tag{6}$$

$$\text{Antireinforced Learning}: \mathbf{w}_j^{(m+1)} = \mathbf{w}_j^{(m)} - \eta \nabla \phi(\mathbf{x}_i(m), \mathbf{w}_j) \tag{7}$$

In (6) and (7), η is an user defined learning rate 0< η ≤1, and the gradient vectors $\nabla\phi$ can be computed in a similar manner as proposed in.[13]

For the data set D_2^i, reinforced and antireinforced learning will be applied to class ω_i and ω_j, respectively. As for the false acceptance set D_3^i, antireinforced learning will be applied to class ω_i, and reinforced learning will be applied to the class ω_j where $x_i(m)$ belongs to.

Threshold Updating The threshold value $\mathbf{T}_i$ of a subnet i in the SPDNN recognizer can also be learned by reinforced or antireinforced learning rules.

2.2.2. Unsupervised Growing of a new cluster

The network enters the Unsupervised Growing phase when the Supervised learning reaches a saturated (learning state) but unsatisfied (classification accuracy) situation. There are three main aspects for the self-growing rules:

I1: *When should a new cluster be created?*

I2: *Which cluster should be partitioned to create a new cluster?*

I3: *How to initialize the center and the covariance of the new cluster?*

On Issue $I1$, when the whole training set has been presented for a few times, the train status (especially the recognition accuracy) remains unchanged or unimproved. An extra cluster is suggested to improve the representation power of the SPDNN.

On Issue $I2$, when an extra cluster is needed, a new cluster is suggested to be created from the subnet which caused the most of misclassification during the recent supervised learning processes.

On Issue $I3$, when a new cluster is needed, its initial values of the *center* and *covariance* needs to be properly determined, otherwise poor classification situation may still exist.

Assume that a training character $\mathbf{x}$ corresponding to class ω_i is presented to an SPDNN classifier: the cluster Θ_i is in class ω_i, and the cluster (say Θ_j) is in the class ω_j which corresponds to the largest response among the classes other than ω_i. Let o_i and o_j be the output of $\mathbf{x}$ from class ω_i and class ω_j, respectively. According to the retrieving scheme of the proposed SPDNN, if o_j is larger than or equal to o_i, the retrieving result for the training character $\mathbf{x}$ must be wrong. As shown in Figure 2(a), clearly the best position for the center of the new cluster should be located at $\mathbf{x}$, i.e., $\mu_0 = \mathbf{x}$, so that the class with new cluster Θ_i' will generate the maximal output o_i for the training character $\mathbf{x}$. To determine the covariance matrix $\mathbf{\Sigma}_0$, first let $\mathbf{\Sigma}_0 = \sigma_0 \mathbf{I}$ and σ_0 be a positive constant (to be determined). As shown in Figure 2(b), if the σ of the new cluster Θ_i' is not properly determined, the class with the new cluster will have its largest possible output $o_i(\mathbf{x})$ to be smaller than the output $o_j(\mathbf{x})$ of class ω_j. In other word, the cluster Θ_i' is *overwhelmed* by cluster Θ_j, where μ_j and $\mathbf{\Sigma}_j$ are the center and covariance of cluster Θ_j. To prevent the *overwhelming* problem, Figure 2(c) presents a properly initiated new cluster Θ_i'. The following two constraints are suggested for a proper initial value of σ.

$$o_j(\mathbf{x}) = \frac{P(\Theta_j \mid \omega_j)}{(2\pi)^{\frac{D}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\frac{(\mathbf{x}-\mu_j)^T(\mathbf{x}-\mu_j)}{\mathbf{\Sigma}_j}\right] + \epsilon_j \ < o_i(\mathbf{x}) = \frac{P(\Theta_0 \mid \omega_i)}{(2\pi\sigma)^{\frac{D}{2}}} + \epsilon_i \tag{8}$$

$$o_i(\mu_j) = \frac{P(\Theta_0 \mid \omega_i)}{(2\pi\sigma)^{\frac{D}{2}}} \exp\left[-\frac{1}{2}\frac{(\mathbf{x}-\mu_j)^T(\mathbf{x}-\mu_j)}{\mathbf{\Sigma}_i}\right] + \epsilon_i' \ < o_j(\mu_j) = \frac{P(\Theta_j \mid \omega_j)}{(2\pi)^{\frac{D}{2}} |\Sigma_j|^{\frac{1}{2}}} + \epsilon_j' \tag{9}$$

where $P(\Theta_0 \mid \omega_i)$ and $P(\Theta_j \mid \omega_j)$ are the prior probability of cluster Θ_i' and Θ_j, respectively. ϵ_i and ϵ_j represent the partial output of classes ω_i and ω_j from the clusters other than Θ_i and Θ_j at $\mathbf{x}$. ϵ_i' and ϵ_j' are the partial output at μ_j of the clusters other than Θ_i and Θ_j. The prior probability $P(\Theta_0 \mid \omega_i)$ of cluster Θ_i' can be initialized as $(\sigma/\bar{\sigma_i})P(\Theta_i \mid \omega_i)$, where $\bar{\sigma_i} = (1/R_i)\sum_{r_i=1}^{R_i} \sigma_{r_i}$, in which σ_{r_i} is the covariance of cluster r_i in class ω_i. Since ϵ_i,ϵ_j, ϵ_i', and ϵ_j', are very small at $\mathbf{x}$ and μ_i, they are ignored in the following σ estimation.

These two constraints imply that cluster Θ_i' and cluster Θ_j will not overwhelm each other. To satisfy (8), σ is initialized to be lesser than $(\frac{P(\Theta_i \mid \omega_i)}{(2\pi)^{D/2}\bar{\sigma}_i o_j(\mathbf{x})})^{\frac{2}{D-2}}$. Then, σ can be iteratively decreased by a small value η $(0 \le \eta \le 1)$ until (9) is satisfied, and the final value of σ can be a proper initial value of σ_o for the new cluster Θ_i'.

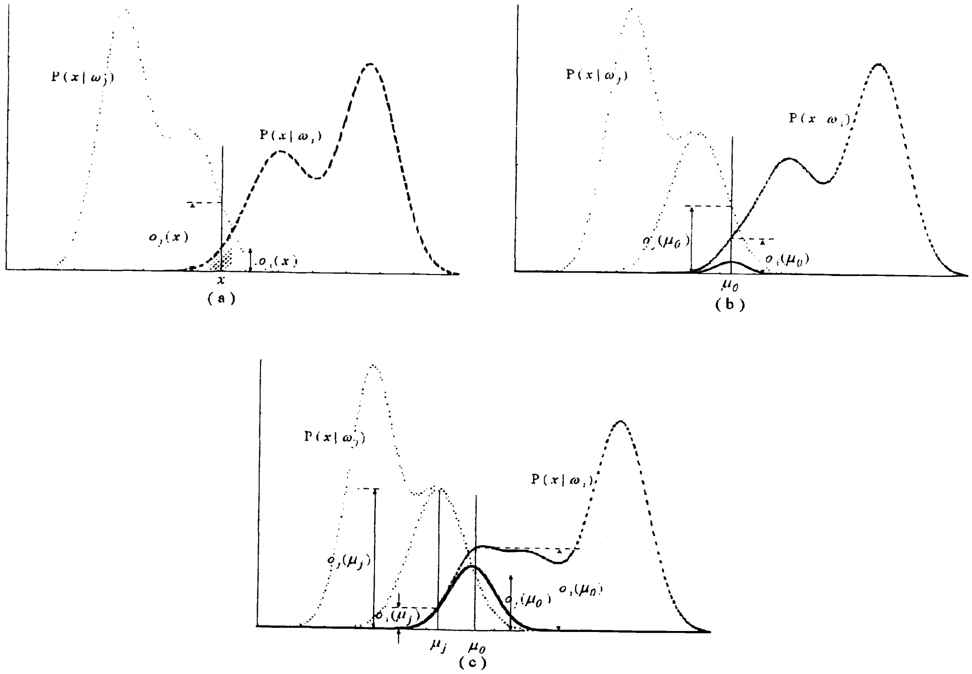

Figure 2. Example of creating a new cluster in a mixture of Gaussian distributions. (a) For $x(t) \in \omega_i$ the $x_i(t)$ is not correctly classified, since $o_i(x(t))$ is smaller than $o_j(x(t))$. A new cluster Θ_i' is needed in ω_i. (b) The new cluster Θ_i' is *overwhelmed* by the cluster Θ_j, i.e., $o_i(\mu_0)$ is still smaller than $o_j(\mu_0)$. (c) By having initialized with proper μ_0, σ_0 and $P(\Theta_0 \mid \omega_i)$, the new cluster Θ_i' can contribute enough to support class ω_i. For example, $o_i(\mu_0)$ is larger than $o_j(\mu_0)$, and $o_i(\mu_j)$ is smaller than $o_j(\mu_j)$.

3. MULTISTAGE HANDWRITTEN CHARACTER RECOGNITION SYSTEM

An SPDNN-based Handwritten character Recognition system is being developed in the Neural Networks Laboratory of National Chiao Tung University. The system configuration is depicted in Figure 3. All the three main modules, including preprocessing and feature extraction module, coarse classifier, character recognizer and its personal adaptive module are implemented on a Pentium-100 based personal computer.

The system built upon the proposed SPDNN has been demonstrated to be applicable under reasonable variations of character orientation, size, and stroke width. This system also has been shown to be very robust in recognizing characters written by various tools; such as pencils, ink pens, mark pens as well as the Chinese calligraphy brushes. The prototype system takes 270 ms in average to identify a character image out of a commonly used Chinese character set (e.g., the 5401 sect[21]) on a Pentium-100 based personal computer. For alphanumerical character recognition, the recognition is about three times faster.

3.1. Image preprocessing and Feature Extraction

Image preprocessing of a Chinese character recognition is by no means of any different from the monolanguage character recognition. Character segmentation on free format handwritten character is a very difficult task, thus it is usually an interactive task between segmentation and recognition. Since the Chinese character recognition is already a complicated recognition problems, and the interactive segmentation methods would slow down the processing speed, we must restrict our handwritten character domain to be free format on Chinese characters and handprinted characters on alphanumerics. Thus, an *Interactive Rule-Based Character Segmentation*[5] is applied to slice and separate the whole page image into a sequence of character image. Basically, this method is based on some heuristic rules to combine several isolated connected-components into a separated character.
The binary images of a handwritten character are then passed through a series of image processing stages, such as

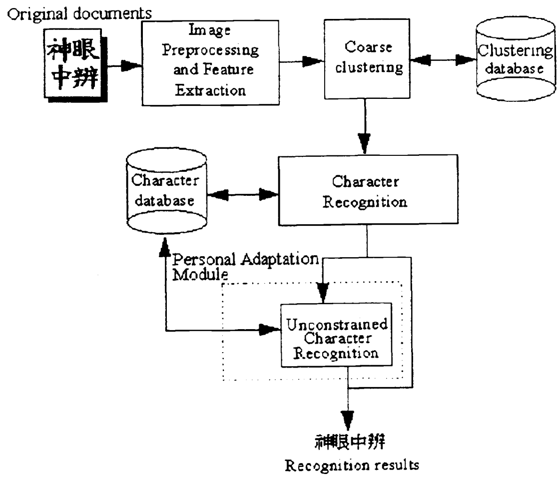

Figure 3. System configuration of the multistage character recognition system. Character recognition system acquires images from a scanner. The coarse classifier determines an input character image to be one of the predefined subclass. The character recognizer matches the input character with a reference character. The personal adaptive module learns the user's own written style to enhance the recognition accuracy.

boundary smoothing, noise removing, space normalization, and stroke thinning operations. Figure 4 depicts a series of preprocessing results of some Chinese and English characters.

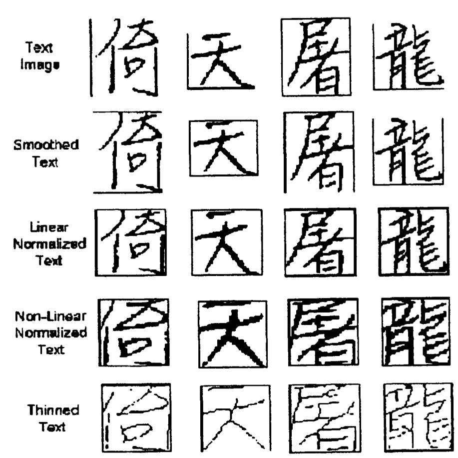

Figure 4. Image preprocessing on handwritten characters: (from top) original text image, smoothed text, linear normalized text, nonlinear normalized text and thinned text.

Feature extraction Using statistical features in pattern recognition has been very successful for a long time. A character can be well represented by a 2-D image pattern, thus many statistical pattern recognition techniques have been applied in this type of character recognition. Among various statistical features,[6] we selected the *crossing count* (CCT), *belt shape pixel number* (BSPN), and *stroke orientation feature* (STKO) as candidate features for the proposed character recognition system. As shown in Figure 5, features such as *CCT* and *BSPN* represent the stroke complexity and the pixel density of a character image. In,[9] Kimura et al. proposed the directional code histogram, and used this feature for Chinese character recognition successfully. As shown in Figure 7, the *STKO* is a simplified version of Kimura's directional code histogram.

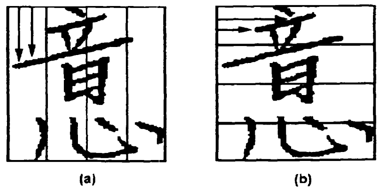

Figure 5. Extracting the CCT and BSPN features from a Chinese character.

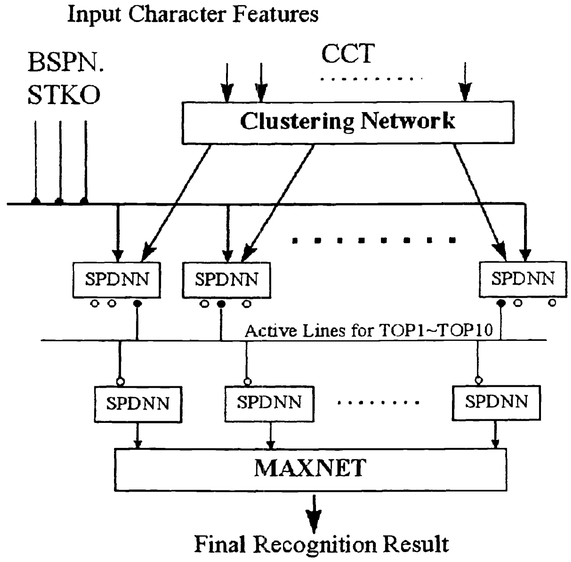

Figure 6. The architecture of the three-stage recognition system.

3.2. Multistage character recognition

Since there are as many as 5401 commonly used characters, and 62 alphanumerics and symbols in a Chinese document, it is desirable to perform a coarse classification (or clustering) to reduce the number of candidate characters for the *character recognition*. With a smaller candidate set, not only the overall recognition speed and recognition accuracy can be greatly improved, but also the training on the SPDNN character recognizer can be much easier and faster. The architecture of a multistage SPDNN character recognizer is depicted in Figure 6.

3.2.1. Coarse classification

In order to achieve a balanced recognition performance in a multistage recognition system, the coarse classifier needs to maintain a very high accuracy, (e.g., $\geq$ 99.9%). Although this is a difficult task, we proposed to use the *CCT* feature and the SPDNN with ***overlapped*** boundaries to implement the coarse classifier. This design is to achieve low sensitivity in personal writing style and high classification rate among characters. By applying the two public databases suggested in Section 3.2.3 and the *ISLUG* principle, we have trained the proposed coarse classifier to achieve this goal. The training and testing results are listed in Table 1. At the end of the retrieving phase of the coarse classifier, the number of candidate characters with respect to the input character is reduced to 516 characters in average.

3.2.2. Character recognition

The design of the character recognizer is also based on the SPDNN model. For a K-character recognition problem, an SPDNN character recognizer consists of K subnets. A subnet i in the SPDNN recognizer estimates the distribution on the patterns of character i only, and treats those patterns which do not belong to character i as the "non i" patterns. The combined features such as CCT, BSPN, and STKO are used in the SPDNN character recognizer. The training of the character SPDNN was conducted with the *ISLUG* principle. During the retrieving phase, each of the

Table 1. THE TRAINING AND TESTING RESULTS OF COARSE CLASSIFICATION ON THE CCL/HCCR1 AND THE CEDAR DATABASES. HALF OF THE RANDOMLY SELECTED CHARACTERS IN EACH OF DATABASES ARE USED FOR TRAINING AND THE OTHER HALF ARE USED FOR TESTING.

Number of cluster	Ave. No. of characters in a cluster	Training Accuracy	Testing Accuracy
61	516	99.9 %	99.8%

subnets corresponding to candidate characters from coarse classifier produces a score according to its discriminate function $\phi(\mathbf{x}(t), \mathbf{w}_i)$. The subnet which produces the highest score is the winner and its corresponding reference character is considered as the result of the character recognizer.

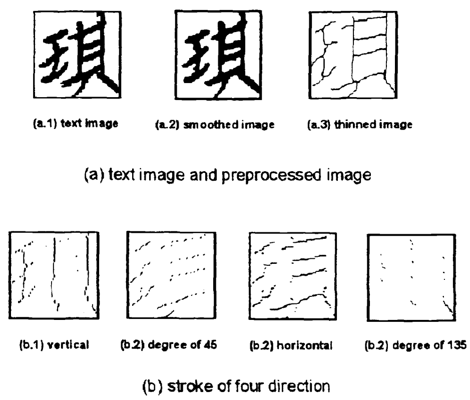

Figure 7. Preprocessing and STKO feature extraction of a Chinese character.

3.2.3. Experimental Results

In this section, experimental results for handwriting recognition will be discussed. We will use the CCL/HCCR1[21] handwritten characters database, which have been used by several handwritten character recognition research groups.[5,6,12]

Handwritten Chinese Recognition: We have conducted experiments on the CCL/HCCR1 database, which contains more than 200 samples of 5401 frequently used Chinese characters. The samples were collected from 2600 people including junior high school and college students as well as employees of ERSO/ITRI. According to the most recent survey on handwriting recognition,[4,14,20] most of the handwritten Chinese OCR were done on small database, i.e., training and testing on very small size of character set, e.g., a few hundred characters. As far as studies conducted on the recognition on a complete set of commonly used Chinese characters, Xia[22] developed an experimental system on the 3755 character set and achieved a 80% recognition rate. Recently, Li and Yu[12] reported a 93.43% recognition accuracy on the CCL/HCCR1 database. Our SPDNN-based system achieved a slightly lower accuracy performance (90.11%). Table 2 summarizes the performance comparison of these two systems based on CCL/HCCR1 database. We would like to comment on why the SPDNN has inferior performance. First, compared to the huge number of character feature sets (400~500) used by,[12] the features used by SPDNN only consists of 96 sets. A more relevant comparison could be made if comparable number of training and testing features for these two systems were available. In fact, the SPDNN character recognizer is designed to use no more than 100 sets of features, since more feature set means more memory storage and longer recognition time. Two reasons explain why SPDNN can live with lesser features and yet achieve a comparable performance. (1) The mixture of Gaussian-based discriminant function permits the SPDNN to learn the precise decision boundaries. (2) The self-growing rules allow for just enough number of Gaussian clusters to represent the character image distribution. Therefore, the proposed SPDNN system is very suitable to implemented on a personal computer system. Li and Yu's system was implemented on a Sparc-2 workstation.

Table 2. PERFORMANCE OF DIFFERENT HANDWRITTEN CHARACTER RECOGNIZERS ON THE CCL/HCCR1 DATABASE. THE *Li-yu1* SYSTEM WAS IMPLEMENTED BY THEIR MAIN FEATURES, AND *Li-yu2* SYSTEM WAS IMPLEMENTED BY THEIR FULL SET OF FEATURES[12].

Systems	Accuracy	Feature sets	Train/Test ratio
SPDNN1	90.12%	96	50-50
Li-yu1	88.65%	400	50-1
Li-yu2	93.43%	500	50-1
SPDNN2	94.02%	194	50-50 with 6% rej.

3.3. SPDNN for Personal Adaptive Recognition

In order to further enhance the recognition accuracy, we proposed personal adaptation on the SPDNN character recognizer.

Unconstrained freehand-written character recognition

Most of the recently announced handwritten character recognition systems claimed their benchmarking recognition performance to be higher than 90%. However, when they were tested on unconstrained freehand-writing, most of their recognition accuracy fell between 40% and 50%.[7] Hence, we suggested an unconstrained freehand-writing recognition module to adaptively fine tune the parameters of the SPDNN character recognizer in order to learn the user's own writing style. When input characters were misclassified, the erroneous recognition results will be manually corrected by a user. In the mean time, the parameters or the decision boundaries of the corresponding character SPDNN are modified and improved by performing the reinforced and antireinforced learning processes. In addition, when it is necessary, clusters in a character SPDNN may be created (self-growing rules) to better approximate the partition boundaries. In order to prevent the excessive learning of the designated character boundary, the adaptive learning process usually include a verification process. Naturally, the reinforced and antireinforced learning processes are applied to the SPDNNs associated with the mismatched character and its similar characters (the TOP 10 candidates). When more and more unconstrained freehand-written characters are presented to the system, each character SPDNN will gradually learn the user's personal writing style.

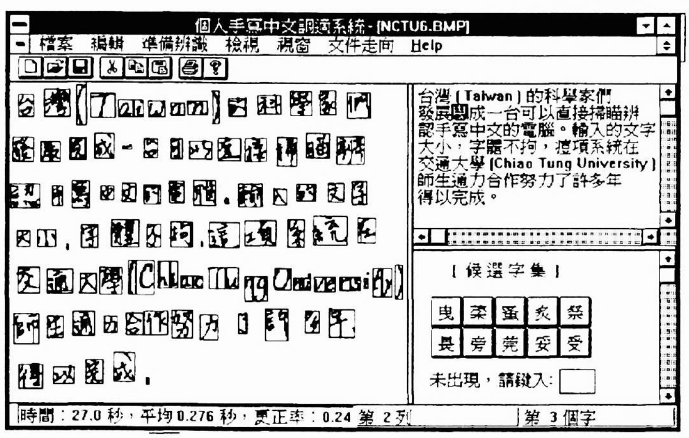

Figure 8. The user interface and a recognition snapshot of the proposed three-stage recognizer.

Experimental results and Performance evaluation

In order to evaluate the performance of the unconstrained freehand-writing recognition module for its adaptation and recognition capabilities, we prepared our in house database (NCTU/NNL) in the following manner. We first selected the most commonly used 300 characters from the Chinese textbooks for the elementary schools in Taiwan. And then, these 300 Chinese characters and the alphanumerics were written without any restriction on the writing style by several students in our university for 10 times in several days. We intended to simulate a natural and general unconstrained freehand-written database in this manner. The testing results for 5 user's adaptation processes are illustrated in Table 3. The recognition rates was raised from 44.09% to 82.2% during the 5 learning cycles. And the performance may finally increase up to 90.03% in 10 learning cycles. Figure 8 depicts the user interface and a snapshot of recognition results of the prototype system.

Table 3. BY APPLYING 300 COMMONLY USED CHARACTERS WRITTEN WITHOUT ANY CONSTRAINTS BY FIVE STUDENTS, THE PROPOSED ADAPTIVE SYSTEM SHOWS SIGNIFICANT IMPROVEMENT ON THE RECOGNITION ACCURACY DURING THE 10 LEARNING CYCLES.

Trial	user#1	user#2	user#3	user#4	user#5	avg.
1st	50.6%	33.6%	38.1%	52.3%	45.7%	44.0%
2nd	67.7%	69.0%	55.5%	56.6%	61.1%	62.0%
3rd	78.6%	80.0%	69.9%	71.5%	72.7%	74.5%
4th	84.3%	78.7%	69.3%	75.9%	86.0%	79.5%
5th	84.6%	87.6%	73.9%	79.9%	85.1%	82.2%
6th	81.9%	89.0%	76.2%	80.2%	84.0%	82.2%
7th	86.5%	89.6%	79.9%	78.5%	84.6%	83.8%
8th	89.5%	90.3%	79.5%	86.9%	89.3%	87.1%
9th	90.5%	90.6%	81.2%	87.7%	89.3%	87.9%
10th	93.6%	91.4%	84.6%	90.5%	90.1%	90.0%

4. PROTOTYPING AND USER INTERFACE DESIGN

The proposed multi-stage recognition system has been implemented on a Pentium-100 based personal computer. The hardware equipment and software environment are:

- hardware: a Pentium-100 personal computer with 8-16 M bytes RAM, and an optical scanner;
- software: TWAIN scanner interface, and Chinese Windows 3.1, or 95.

User interface design Since this system may require an user to correct recognition errors, thus a friendly user interface becomes necessary. Figure 8 depicts the user interface and a snapshot of recognition results of the prototype system.

The user interactive window contains three partitions: the left partition displays the binary image of an A4 sized handwritten document, the upper-right partition displays the recognition results and the lower-right partition contains an interactive error correcting interface. When a user intends to correct some errors from the primary recognition results, one can move the cursor to point the misclassified word first, then the top 10 recognition results will be displayed at the buttons. If a correct word is displayed in one of the button, then the user can point and click the cursor to make the correction. If the correct character is not in the top 10 list, then the user can manually type in a character to correct the misclassified character.

5. CONCLUDING REMARKS

In this paper, a neural network handwritten Chinese character recognition system is proposed and implemented on a Pentium-100 based personal computer. This system performs classification, character recognition, and unconstrained freehand-writing recognition. The SPDNN, a Bayesian Decision-based Neural network, was applied to implement the major modules of this system. This modular neural network deploys one subnet to take care one object (character), and therefore it is able to approximate the decision region of each class locally and precisely. This locality property is attractive especially for personal freehand-writing or signature identification applications. A personal adaptation for the character recognition module is proposed and implemented to improve the recognition performance on the unconstrained freehand-writing. On the other hand, due to the enormous number of variations involved, handwriting recognition applications still require more work before they can reach comparable performance by a human. Therefore, document analysis and recognition become an interesting and fascinating research topic in the field of intelligent information processing.

Acknowledgments

The authors acknowledge Prof. S. -Y. Kung, Dr. S. -H. Lin for their helpful suggestions regarding the probabilistic DBNN and statistical pattern recognition methods.

REFERENCES

1. H.S. Baird, "Feature identification for hybrid structural/statistical pattern classification," *Computer Vision, Graphics, and Image Processing*, vol. 42, pp. 318-333, 1988.
2. R.M. Bozinovic and S.N. Srihari, "Off-line cursive word recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 68-83, Jan 1989.
3. D.J. Burr, "Designing a handwriting reader," *IEEE Trans. Pattern Anal. Machine Intell.,* vol.PAMI-5, pp554-559, 1983.
4. F.H. Cheng and W.H. Hsu, " Research on Chinese OCR in Taiwan," *Int. J. of Pattern Recognition and Artificial Intelligence*, Vol. 5, No. 1 & 2 (1991), pp. 139-164.
5. Cheng-Chin Chiang, Tze Cheng, Shiaw-Shien Yu, "An Iterative Rule-Based Character Segmentation Method for Chinese Documents," in *Proc. of Int. Conf. on Chinese Computing'96,* June 4-7, 1996 Singapore.
6. H.C. Fu and K.P. Chiang, "Recognition of Handwritten Chinese characters by Multi-stage Neural network classifiers," in *Proc. of 1995 IEEE Int. Conf. Neural Networks*, Perth, Australia.
7. H.C. Fu, S.C. Chuang, Y.Y. Xu, W.H. Su, and K.T. Sun, " A Personal Adaptive Module for Unconstrained Handwritten Chinese Characters Recognition, " in *Proc. of the International Symposium on Multi-technology Information Processing,* Hsinchu, Taiwan, ROC, Dec., 1996.
8. Jonathan J. Hull, "A Database for Handwritten Text Recognition Research," in *IEEE Trans. Pattern Analysis Mach. Intell.*, Vol. 16, No. 6, pp. 550-554, 1994.
9. F. Kimura, K.Takashina, S. Tsuruoka and Y. Miyake, "Modified quadratic discrimininant functions and Application to Chinese character recognition," in *IEEE Trans. Pattern Analysis Mach. Intell.*, Vol. 9, No. 1, pp. 149-153, 1987.
10. S.Y. Kung, and J.S. Taur, "Decision-based Hierarchical Neural Networks with Signal/Image Classification Applications," in *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 170-181, Jan 1995.
11. Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.R. Howard, W. Hubbard, and L.D. Jackel, "Handwritten zip code recognition with multi-layer networks," in *Proc. 10th Int. Conf. Pattern Recognition*, 1990, pp.35-40.
12. Tze-Fen Li, and Shiaw-Shian Yu, "Hand-printed Chinese Character Recognition Using the Probability Distribution Feature," *Int. J. of Pattern Recognition and Artificial Intelligence*, Vol. 8, No. 5, (1994), pp. 1241-1258.
13. Shang-Hung Lin, S.Y. Kung, and L.J. Lin, "Face Recognition/Detection by Probabilistic Decision-based Neural Networks," in *IEEE Transactions on Neural Networks, special issue on Artificial Neural Network and Pattern Recognition*, Vol. 8, No. 1, pp. 114-132, 1997.
14. S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of IEEE*, vol. 80, no. 7, pp. 1029-1058, July, 1992.
15. G. Nagy, "Optical character recognition," *Handbook of Statistics*, P. R. Krishnaiah, L. N. Kanal, Eds., pp. 621-649, North Holland, 1982.
16. G. Nagy, "Chinese character recognition: A twenty-five year retrospective," in *Proc. 12th Int. Conf. Pattern Recognition*, pp163-167, 1988.
17. H. Ogawa and Y. Tezuka, "Recognition of handwritten Chinese characters by means of hierarchical representation," in *Proc. Int. Comput. Symp.*, Taipei, 1975, Vol. 1, pp. 411-420.
18. I.-S. Shyu, L.-T. Tu, M.-Y. Chen, K.-H. Chou, and W.-W. Lin, "Design of a decision tree and its application to large-set printed Chinese character recognition," in *Proc. Int. Conf. Comp. Processing of Chinese & Oriental Lang.*, Toronto, Canada, 1988, pp. 126-130.
19. C.Y. Suen, M. Berthod, and S. Mori, "Automatic recognition of hand-printed character - the state of the art," *IEEE Proceedings*, vol. 68, pp. 469-487, April 1980.
20. Ju-Wei Tai "Some Research Achievements on Chinese Character Recognition in China," *Int. J. of Pattern Recognition and Artificial Intelligence*, Vol. 5, No. 1 & 2 (1991), pp. 199-206.
21. L.T. Tu, Y.S. Lin, C.P. Yeh, I.S. Shyu, J.L. Wang, K.H. Joe, and W.-W. Lin, "Recognition of hand-printed Chinese characters by feature matching," in *Proc. of 1991 First National Workshop on Character Recognition*, Taipei, ROC, 1991, pp. 166-175.
22. Yin Xia, "Research Report on interactive self-learning system of handwritten Chinese characters," Department of Computer Science, QingHua University, Nov. 1989.
23. M. Yoshida and M. Eden, "Handwritten Chinese character recognition by A-b-s method," in *Proc. 1st Int. Joint Conf. Pattern Recognition*, Nov. 1973, pp. 197-204.