Proceedings of International Conference
on Artificial Intelligence

Preclassification for Handwritten Chinese Character Recognition Using Fuzzy Rules and SEART Neural Net

Jyh-Ming Chen, Chin-Chou Lin and Hahn-Ming Lee*
Department of Electronic Engineering
National Taiwan Institute of Technology
Taipei, Taiwan, ROC
E-mail: hmlee@et.ntit.edu.tw

Abstract

In this paper, a method of character preclassification for handwritten Chinese character recognition is proposed. Since the number of Chinese characters is very large (at least 5401s for daily use), we employ two stages to reduce the candidates of input character. In stage I, we try to extract the first primitive features from handwritten Chinese characters and use the fuzzy rules to create the four preclassification groups. The purpose in stage I is to reduce the candidates roughly. In stage II, we extract the second primitive features from handwritten Chinese characters and then use the Supervised Extended ART (SEART) as the classifier to generate the preclassification classes for each preclassification group that we create in stage I. Since the number of characters in each preclassification class is smaller than that in the whole character set, the problem becomes simpler. In order to evaluate the proposed preclassification system, we use the 605 Chinese character categories in the text books of elementary school as our training and testing data. The database used is HCCRBASE (provided by CCL, ITRI, Taiwan). We select the even samples of samples 1—100 as the training set, and the odd samples of them as the testing set. The preclassification rate that characters of testing set can be distributed into correct preclassification classes is 98.11%.

Keywords: preclassification, handwritten Chinese character recognition, fuzzy rules, membership functions, α level cut.

1. Introduction

The difficulties of Chinese character recognition are: large character sets of characters (at least 5401s for daily use), very complex in structure, different writing styles, and many similar characters in shape [1] [2] [3]. Because of the writing habit, the handwritten Chinese character recognition (HCCR) is more difficult. In addition, since we can easily extract the strokes and know the sequence of them, the on-line HCCR is easier than the off-line HCCR. In order to overcome the difficult problems of HCCR that we mentioned above, the researchers often use these following methods [1]: (1) Using preclassification technique to reduce the candidate characters, and then select a character from them. This can relieve the problem of large character sets. (2) Since a Chinese character is composed of several components — radicals, deassemble a character to many radicals may reduce the problem of vary complex in Chinese character structure. (3) Using neural network model and fuzzy concept to lighten the problem of various writing styles. (4) Induce the structure about the shape of Chinese characters can reduce the difficulty of the recognition of similar characters.

Since the scale of Chinese character recognition system is large, we cannot examine each topic of the system and process it nicely. Therefore, in this paper, we focus on the preclassification of off-line HCCR. We use a 2-stage method to accomplish the preclassification. In stage I, we try to extract the first primitive features from handwritten Chinese characters and use the fuzzy rules to create the four preclassification groups. The first primitive features are a kind of statistical feature and they are easy to be extracted. The reasons we adopt the fuzzy concept in stage I to create the four preclassification groups are: fast preclassification speed and overlapping preclassification groups. The purpose in stage I is to reduce the candidates roughly, so we can not spend much time in stage I. Furthermore, in order to make sure the high preclassification rate in stage I, we allow the preclassification groups to overlap each other. The fuzzy concept can help us to achieve these goals. In stage II, we extract the second primitive features from handwritten Chinese characters and then use the supervised extended ART (SEART) [4] as the classifier to generate the preclassification classes for each preclassification group that we create in stage I. The second primitive features have the properties of local and structural features. We also allow the preclassification classes to overlap each other, but we set the overlap degree appropriately to control the preclassification class size. The number of preclassification classes should be large and the class size ought to be small. These two factors are the

* correspondence to this author

performance measure of the preclassification system [5]. Our proposed system will produce a good preclassification result in terms of the above two factors. In our experiment, when we control each preclassification class below 25 character categories, the preclassification rate is acceptable. The preclassification rate that characters of testing set can be distributed into correct preclassification classes is 98.11%.

Following this introduction is a detailed description of the proposed system. Section 3 shows the experimental results. Section 4 includes the advantages and the major reasons for preclassification errors of our proposed system. The final section contains the conclusions of this paper.

2. System architecture

The system architecture of our proposed system is illustrated in Fig. 1. In the training phase, the training pattern is sent to the preprocessing unit to accomplish the normalization, thinning, and dividing of image. Then the processed image is sent to the feature extractor. The feature extractor is responsible for extracting the first and second primitive features. After extracting the features from all the training patterns, we analyze the first primitive features and define some membership functions and fuzzy rules. In summary, we use two stages to create the preclassification classes. In stage I, we use the first primitive features, membership functions, and the fuzzy rules to categorize the training patterns into four preclassification groups. In stage II, the second primitive features of the training patterns that belong to each preclassification group are sent to the corresponding SEART neural net model. After training the corresponding SEART of each group, we can create the preclassification classes from each output node of SEARTs.

In the testing phase, the pattern is sent to the preprocessing unit and feature extractor to finish the extraction of first and second primitive features. In stage I, by using the first primitive features, membership functions, and the fuzzy rules, we can decide which preclassification group this pattern belongs to. In stage II, the second primitive features of this pattern will be sent to the corresponding SEART of the group that is chosen in stage I. The SEART classifier will then determine which preclassification class this pattern belongs to. We will detail the components of the Fig. 1 in the following subsections.

2.1 Preprocessing unit

The preprocessing unit is composed of normalization, thinning, and dividing. At first, we normalize the input character image to the same size, i.e., 72×72 in our system. Thinning is then applied to find the skeleton of the character image. We adopt Chens's modified thinning algorithm [6] for its advantages: fast speed and immunity the contour noise.

Before extracting the second primitive features, we divide the character image. Bai [7] have made an interesting experiment. He selects 29 printed Chinese characters and purges the central region of the character image. Only the four peripheral regions of these characters are left to be identified in three seconds by persons. The average recognition rate in this experiment is 83%. It shows that the information of the peripheral regions in Chinese character images plays an important role in Chinese character recognition. In Wang's four corner method [8], he founded the four corner codes to represent Chinese characters by their four corners. It is successful to separate the Chinese characters into many different categories. Tham and Lee [9], Chang and Wang [10], and Bai [7] have extracted the features from the peripheral regions or from four corners of the Chinese characters and the recognition result is acceptable. For this reason, we decide to emphasize the peripheral parts of the Chinese characters. In addition, we adopt the dividing method used in Lee and Sheu [11].

The character image is divided into nine blocks

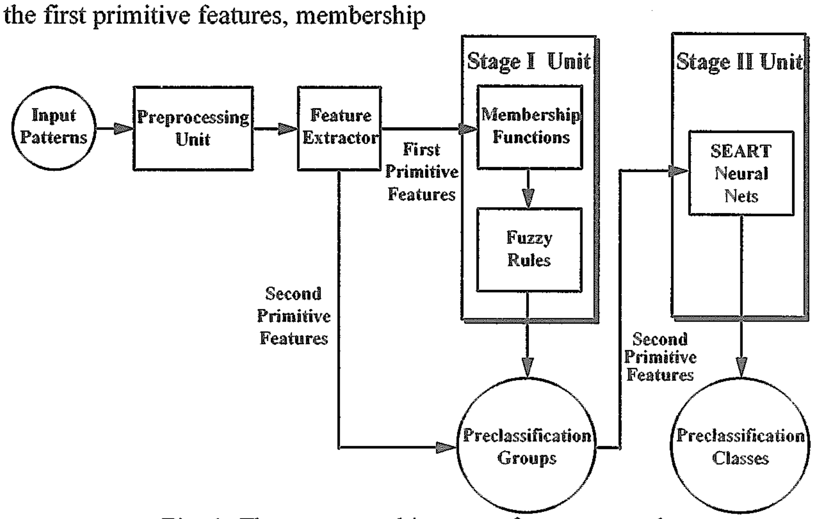


Fig. 1. The system architecture of our proposed system.

equally. Fig. 2 shows the nine dividing blocks used. We extract the second primitive features from the shadowy regions shown in Fig. 2. We emphasize the peripheral parts in the nine dividing blocks. The central part and the periphery overlap a little, i.e., 5 pixels from each central part edge used in our system, to avoid the writing position variations.

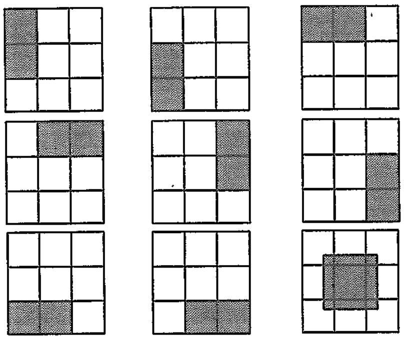


Fig. 2. The nine dividing blocks used for the second primitive feature extraction.

2.2 Feature extractor

The feature extractor is responsible for extracting the first and second primitive features. We explain the features we use in the following subsections.

2.2.1 The first primitive features

Neocognitron was used in alphanumeric character recognition by Fukushima et al. [12] There are 12 templates in the S1 layer of Fukushima's Neocognitron. It categorizes the 12 templates into 8 categories. Fig. 3 shows the 12 templates and the corresponding 8 categories (S1 to S8). Lee and Sheu [11] have employed them to be the features in HCCR.

The 8 categories can be treated as 8 stroke directions. We adopt part of them to be the first primitive features in our system. The category S1 of the 12 templates is a pure horizontal template. The categories S2 and S8 are the variations of S1. On the other hand, the category S5 is a pure vertical template. The categories S4 and S6 are the variations of S5. In order to count the numbers of horizontal and vertical templates as well as their variations, a 3×3 window is used to trace the character

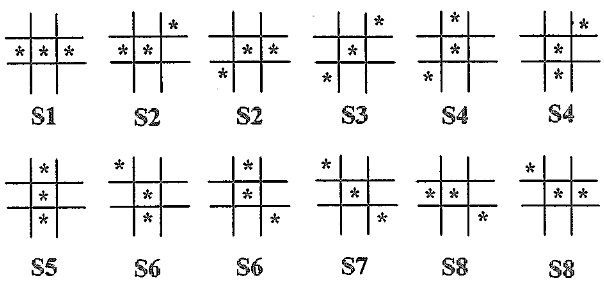


Fig. 3. The 12 templates and corresponding 8 categories used in Neocognitron.

image from left to right and from top to bottom. We match the 3×3 window with the categories S1, S2, S8, S4, S5, and S6. Then we obtain the following two equations:

$$H = \frac{\text{The number of categories S1, S2, and S8 in the character image}}{\text{The number of black pixels in the character image}} \quad (2.1)$$

$$V = \frac{\text{The number of categories S4, S5, and S6 in the character image}}{\text{The number of black pixels in the character image}} \quad (2.2)$$

Horizontal density H and vertical density V in the above equations are our first primitive features.

2.2.2 The second primitive features

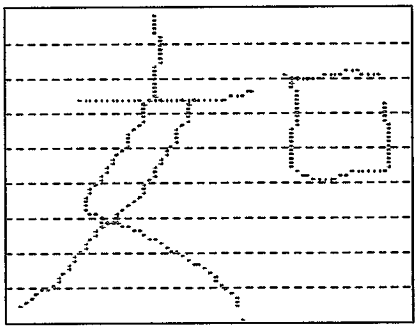
The second primitive features are composed of stroke direction features and stroke density features. Lee and Sheu [11] also used them to be the features in HCCR.

(1) Stroke direction features

As we mentioned above, the 8 categories in Fig. 3 can be treated as 8 stroke directions. Besides, these 8 categories are tolerant of the local variations. For example, the category S4 may be a variation of S3 or S5. We adopt the 8 categories in Fig. 3 to be the stroke direction features in our proposed system. In order to count the numbers of 8 categories in each shadowy dividing block mentioned in Fig. 2, a 3×3 window is used to trace the character image from left to right and from top to bottom. We match the 3×3 window with the 12 templates. After we count the numbers of the 8 categories for each shadowy dividing block, we normalize these numbers to [0,1]. The normalized values are our 72 stroke direction features.

(2) Stroke density features

A thin stroke of row means that the number of continuous pixels in the same row is less than 4, and the number of continuous pixels in the same row greater than 4 is a bold stroke of row. In order to count the numbers of thin and bold strokes of row, respectively, we horizontally scan the character image for each row. We accumulate the numbers of thin strokes and bold strokes of row for 8 rows to reduce the input dimension. Since we normalize the source image to the same size, i.e., 72×72, there are 9 clusters after scanning the 72 rows. Each cluster has 2 features: the numbers of thin strokes and bold strokes of row. The 18 numbers are normalized to [0,1] and used as the horizontal one of the stroke density features. Similarly, we can define the other two terms: thin stroke and bold stroke of column, and obtain the other 18 features in vertical scan. Therefore, we have 36 stroke density features. In Fig. 4, the example of Chinese character "如" (if) is scanned horizontally to count the thin and bold strokes. Because the number of stroke direction features is 72 and the number of stroke density features is 36, the total number of second primitive features is 108.



Number of strokes	
Thin	Bold
7	0
10	2
23	2
32	0
29	2
16	0
14	1
16	0
13	0

Fig. 4. Chinese character example "如" (if) in horizontal scan. Scan horizontally and count the numbers of thin strokes and bold strokes of row for 8 rows.

2.3 Membership functions and fuzzy rules

In stage I of the preclassification process, we want to reduce the candidates roughly. The method we adopt is to define some membership functions and then use the fuzzy rules to create four preclassification groups.

2.3.1 The membership functions

In the training phase, after extracting the features from all the training patterns, we analyze the first primitive features and define some membership functions. In the testing phase, the membership functions and α level cut [13] are used to determine the types, i.e., Low or High of horizontal density H and vertical density V. In order to simplify our question, we adopt the trapezoidal-shaped membership function. Fig. 5 (a) shows the membership function of fuzzy set Low for horizontal density H. We sort horizontal density H of all training patterns ascendently by their values. We then pick the top 20% of the sorted horizontal density H and membership values of them to fuzzy set Low are set as 1.0. Therefore, the coordinate of point a in Fig. 5 (a) can be determined. Since the value of horizontal density H is not more than 1.0, the coordinate of point b in Fig. 5 (a) is (1.0, 0.0). Once the coordinates of points a and b are determined, we can sketch the Fig. 5 (a) completely. Similarly, the membership function of fuzzy set High for horizontal density H can be illustrated as Fig. 5 (b). Also, we can obtain the membership functions of fuzzy sets Low and High for vertical density V.

The α level cut is used to determine the types, i.e., Low or High of horizontal density H and vertical density V. For example, if the α level is set to be 0.8, and one character gets the horizontal density H degree 0.7 for Low and 0.85 for High, then we treat the horizontal density H as type High rather than Low. However, if we set α to be 0.6, the horizontal density H can be treated to be both Low and High.

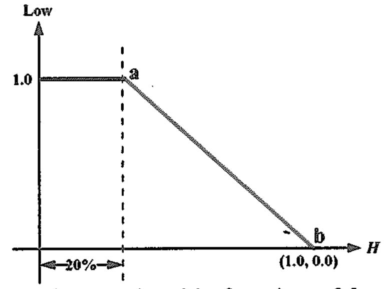


Fig. 5 (a) The membership function of fuzzy set Low for horizontal density H.

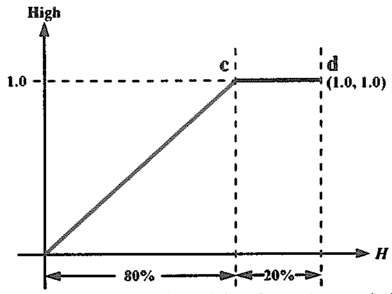


Fig. 5 (b) The membership function of fuzzy set High for horizontal density H.

2.3.2 The fuzzy rules

The fuzzy rules used in the stage I of proposed system are given as follows:

(1) If the horizontal density H of the character is **High** and the vertical density V of the character is **High**, then the character belongs to **Preclassification Group 1**.

(2) If the horizontal density H of the character is **High** and the vertical density V of the character is **Low**, then the character belongs to **Preclassification Group 2**.

(3) If the horizontal density H of the character is **Low** and the vertical density V of the character is **High**, then the character belongs to **Preclassification Group 3**.

(4) If the horizontal density H of the character is **Low** and the vertical density V of the character is **Low**, then the character belongs to **Preclassification Group 4**.

2.3.3 The creation of the four preclassification groups

After determining the membership functions, fuzzy rules, and α level, we can get the types of horizontal density H and vertical density V for each training pattern, and then create the four preclassification groups. The lower value of the α level is, the larger size and overlap degree of preclassification groups have. It makes the preclassification rate higher in stage I, but the large group size leads to the large number of candidates. In stage II, the large group size may lead to the bad total result. On the other hand, the higher value of the α level is, the smaller size and overlap degree of preclassification groups have. It makes the low preclassification rate in stage I and may lead to the bad total result, too. Therefore, appropriately setting α value is useful for obtaining better result.

Once we accept a testing pattern and extract its features, we can get the values of its horizontal density H and vertical density V. If the horizontal density H (or vertical density V) degree for Low is larger than it for High, we treat the horizontal density H (or vertical density V) as type Low, otherwise we treat it as type High. Then we can use the fuzzy rules defined in Section 2.3.2 to determine which preclassification group the testing pattern belongs to.

2.4 The SEART classifier

In stage II, we adopt the Supervised Extended ART (SEART) neural network model [4] as our classifier to create the preclassification classes. After creating the four preclassification groups in stage I, the second primitive features of the training patterns that belong to each preclassification group are sent to the corresponding SEART neural net model. After training SEARTs, the preclassification class for each output node will be set. In Section 2.4.2, we will detail the method we use in setting the preclassification classes. Furthermore, an example will be given to explain it.

2.4.1 The SEART architecture

The SEART neural net architecture is shown in Fig. 6. The two lower layers are essentially fuzzy ART [14] with complement coding, but with different connection weights and match control. The input vector is transmitted through the hyperbox connections and the prototype connections from F1 input field to F2 category field. Then, a node in F2 is chosen as a candidate category for the input vector according to the choice function. Next, the concept contained in the category is read out through the hyperbox connections. During training, the candidate category needs to be checked by the match control unit. If the category is accepted, the concept will be generalized via including the input vector into the original concept.

A layer of Grossberg's Outstars [15] is superimposed on the top of F2 layer. During learning, the Outstar connection weights learn the desired output for the category chosen on F2. Initially, all categories are said to be uncommitted. After a category is selected and meets the matching criteria, it becomes committed. If the chosen category is committed, F3 layer will read out the predicted output corresponding to the category. During training, match signals that reflect the match degree between desired output and predicted one are computed in F3. The point is that another category must be chosen for the input if the desired output is different from the predicted one. The match control unit is in charge of this.

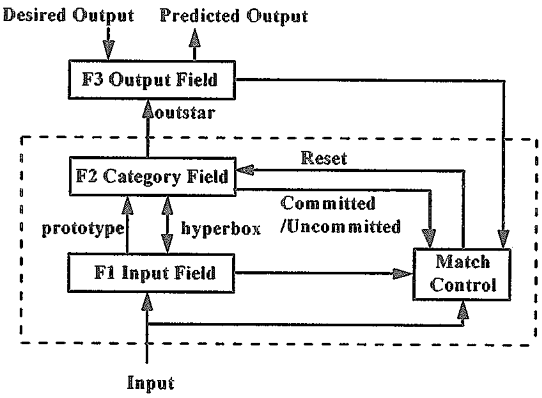


Fig. 6. Supervised Extended ART (SEART) architecture.

2.4.2 The creation of the preclassification classes

The number of SEART's input nodes is equal to the number of the second primitive features, i.e., 108. The number of SEART's output nodes is equal to the number of character categories in the corresponding preclassification group. In the original SEART model, for each input vector I, node J in F2 layer is chosen for candidate category according to the certain choice function C_j. The choice function will consider the Euclidean distance between the input vector I and the hyper-rectangle corresponding to the jth node in F2. In addition, it will also calculate the distance between the input vector and the prototype vector B_j. Thus, the choice function considers the learned concept and the statistic characteristic in the hyper-rectangle. The node in F2 that has the smallest choice function value is selected. The activation value of it is set to be one, and the activation values of the other nodes are set to be zero.

After training each SEART, we use the training patterns again to test the SEART. In order to create the preclassification classes, when we input a training pattern to SEART, we keep the top 10 categories that have the smallest choice function value. We give an example below to explain how we create the preclassification classes.

Suppose there are 15 character categories in the first preclassification group, then the corresponding SEART has 15 output nodes. We then generate the preclassification class for each output node. After training the SEART, we use the training patterns again to test it. Suppose the training pattern $\tilde{X}_0$ is entered, and the top 10 categories, for examples 0, 2, 4, 6, 8, 10, 11, 12, 13, 14, that have the smallest choice function values are kept. Then this training pattern is included to the preclassification classes 0, 2, 4, 6, 8, 10, 11, 12, 13, and

14 of this SEART model. In order to control the size of each preclassification class, when a character category is included to some preclassification class, we count the number of times that the character category appears to that preclassification class. After creating the preclassification classes, we reserve the character categories with higher appearance rate for each preclassification class.

3. Experimental results

In order to evaluate the proposed system, we use the 605 Chinese character categories in the text books of elementary school as our training and testing data. The database used is CCL/HCCR3 of HCCRBASE provided by Computer and Communication Research Laboratories of Industrial Technology Research Institute, Taiwan (CCL, ITRI, Taiwan). In the following experiments, the net parameters: choice parameters, vigilance parameters, and nest parameters used in SEART, are set to 0.3, 0.6, 0.5, respectively. We keep the top 10 categories that have the smallest choice function values for each input training pattern to generate the preclassification classes.

We select the samples 1 — 100 of CCL/HCCR3 to evaluate the proposed system. We adopt the even samples as the training set, and the odd samples as the testing set.

(1) Stage I

In stage I, we use the training set to create the membership functions and preclassification groups. We have introduced how we define the membership functions in Section 2.3.1. Since the size of training set is 30250 characters, 20% of them is 6050 characters. The value of the 6050th sorted horizontal density H is 0.34, therefore, we can determine the membership function of fuzzy set Low for horizontal density H easily. Similarly, we can obtain the other membership functions of fuzzy sets for horizontal density H and vertical density V. Table 1 lists the different α values and their effect on each preclassification group. In this experiment, the 0.8 for α value seems to be the best choice.

Table 1. The different α values and their effect on each preclassification group.

α value		0.6	0.7	0.8	0.85	0.9
the size (character categories) of preclassification groups	Group 1	578	496	278	48	0
	Group 2	583	550	475	407	341
	Group 3	596	542	480	411	368
	Group 4	595	545	411	178	57
preclassification rate in stage I		100%	99.99%	99.99%	95.22%	89.21%

(2) Stage II

In stage II, the second primitive features of the training patterns that belong to each preclassification group are sent to the corresponding SEART neural net model. Table 2 lists the preclassification rates of various preclassification class sizes in each preclassification group (the α level is set to be 0.8). From Table 2, we can find that the character categories of each preclassification class less than 25 seems to be the best choice. Table 3 lists the detailed experimental results when we control the character categories of each preclassification class less than 25 in each preclassification group. The total preclassification rate is 98.11%.

Table 2. The preclassification rates of various preclassification class sizes in each preclassification group (the α level is set to be 0.8).

preclassification rate	↘	preclassification groups			
		Group 1	Group 2	Group 3	Group 4
≤ N character categories in each preclassification class	50	98.13%	99.16%	98.91%	99.42%
	40	98.82%	98.91%	98.83%	99.21%
	30	98.58%	98.62%	98.20%	99.01%
	25	98.26%	98.32%	97.76%	98.92%
	20	98.10%	96.96%	96.89%	98.38%
	10	96.05%	94.36%	94.57%	95.85%

Table 3. The detailed experimental results of each preclassification group in stage II (we control the preclassification class size below 25 character categories).

	Group 1	Group 2	Group 3	Group 4
the number of classes	278	475	480	411
the size of the smallest class	6	7	4	5
the size of the largest class	25	25	25	25
the average size of class	24.45	24.67	24.68	24.78
the number of testing patterns that are distributed into corresponding group	1266	13098	13669	2217
the number of testing patterns that are not distributed into the proper group	1	0	2	1
the number of testing patterns that cannot be distributed into correct class	21	220	304	23
preclassification rate	98.26%	98.32%	97.76%	98.92%
total preclassification rate	98.11%			

4. Discussion

In what follows, we discuss the characteristics of the proposed system and the problems we met during the experiments.

4.1 Feature extraction

Generally speaking, poor thinning will cause the stroke extraction fail and thus lead to bad recognition result. Fortunately, it does not have significant effects in our system. The features used in our proposed system are a kind of statistical feature. They also have the properties of the local and structural features. The feature extraction is to count the numbers of the 8 categories (see Fig. 3) and the numbers of the continuous pixels in each row and column. It is easy to achieve and just spend a little time.

4.2 Fault tolerance

The variations, distortions and noises will affect the stability of features. Favorably, the features used in our proposed system are stable and insensitive to the conditions mentioned above. Because we use the global concept to extract the second primitive features in the large dividing block, it has better tolerance ability than the general local features do. Since the neural networks have the fault tolerance and generalization ability [16], we use the SEART neural network model to handle imperfect or incomplete data, providing the tolerance for variations and distortions.

4.3 Performance measure

The number of preclassification classes should be large and the class size should be small. These two factors are the performance measure of the preclassification system [5]. In our system, the number of preclassification classes is equal to the total size of all preclassification groups. When we control the preclassification class size below 25 character categories, the preclassification rate is acceptable. Our proposed system will produce a good preclassification result in terms of the above two factors.

4.4 The reasons for preclassification errors

From the experiments, we observe that there are two major reasons to cause the preclassification errors. In what follows, we discuss these two reasons, respectively.

4.4.1 Character displacement

Although we adopt the dividing blocks and SEART mechanisms to avoid the influence of writing habit, the primitive features we employed are still sensitive in character displacement. For example, all the Chinese characters "午" (noon) in the training set look like the Fig. 7. The vertical stroke of the training character "午" is in the three central blocks. However, the vertical stroke of the Chinese character "午" in the sample 20 of testing set shown as Fig. 8 is in the three right blocks. When we extract the stroke direction features from the dividing blocks mentioned in Fig. 2, the features extracted from the character "午" of sample 20 are quite different from the features extracted from the character "午" of the other samples. Thus, the preclassification error occurs in testing character "午" of sample 20.

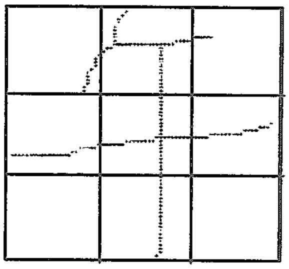


Fig. 7. The preprocessing result of Chinese character "午" (noon) in the sample 1 of database CCL/HCCR3.

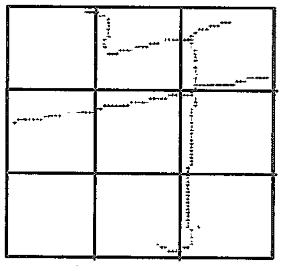


Fig. 8. The preprocessing result of Chinese character "午" (noon) in the sample 20 of database CCL/HCCR3.

4.4.2 Serious damage of the stroke shape

Normalization and thinning processes may lead to damage of the stroke shape. If the damage is serious, the preclassification error may take place. Fig. 9 shows the preprocessing result of Chinese character "多" (much) of sample 1 in training set. Most of the characters "多" in the training set look like the Fig. 9. However, the Chinese character image "多" in sample 30 of database CCL/HCCR3 is quite long and narrow. The normalization process damages the stroke shape seriously and causes the character displacement. The preprocessing result of this character sample is shown in Fig. 10. The damage of stroke shape causes the changes of the stroke directions of many pixels. Thus, we can not extract the correct stroke direction features. In addition, since the character displacement, the influence mentioned in Section 4.4.1 will take place. These make the preclassification error.

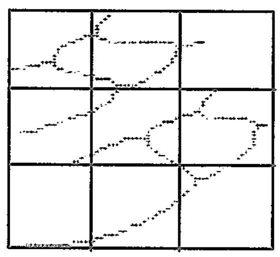


Fig. 9. The preprocessing result of Chinese character "多" (much) in the sample 1 of database CCL/HCCR3.

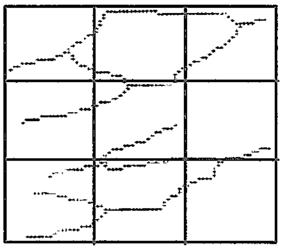


Fig. 10. The preprocessing result of Chinese character "多" (much) in the sample 30 of database CCL/HCCR3.

5. Conclusion

This paper proposes a handwritten Chinese character preclassification technique based on fuzzy concept and neural net model. The experiments on 605 categories of handwritten Chinese characters are encouraging. Each preclassification class is less than 25 character categories. The advantages of our technique are the large number of preclassification classes and the small preclassification class size.

Although the proposed system can be used as the preclassification approach for HCCR, there are some issues needed to be investigated. We summarize them as follows:

(1) For completing the character recognition, fine classification strategy is needed. Since the preclassification class size is small, the fine classification may adopt scheme which is powerful but computation intensive such as relaxation [17].

(2) Although we adopt the dividing blocks and SEART mechanisms to avoid the influence of writing habit, the primitive features we employed are still sensitive in character displacement and serious damage of the stroke shape. Thus, how to extract the features that can relieve the problems as discussed in Section 4.4 need to be further investigated.

(3) We also hope that we can extend our system to deal with 5401 frequently used handwritten Chinese characters in future.

References

[1] Fang-Hsuan Cheng and Wen-Hsing Hsu, "Research on Chinese OCR in Taiwan," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 5, no. 1 & 2, pp. 139-164, 1991.

[2] Chen-Tsun Chuang and Lin Yu Tseng, "A heuristic algorithm for the recognition of printed Chinese characters," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, no. 4, pp. 710-717, 1995.

[3] Chen-Tsun Chuang and Lin Yu Tseng, "A stroke extraction method for multifont Chinese characters based on the reduced special interval graph," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, no. 7, pp. 1171-1178, 1995.

[4] Hahn-Ming Lee and Chia-Shing Lai, "Supervised extended ART: a fast neural network classifier trained by combining supervised and unsupervised learning," *Applied Intelligence*, vol. 6, no. 2, pp. 117-128, 1996.

[5] Rei-Heng Cheng, Chi-Wei Lee, and Zen Chen, "Preclassification of handwritten Chinese characters based on basic stroke substructures," *Pattern Recognition Letters*, vol. 16, no. 10, pp. 1023-1032, 1995.

[6] Yung-Sheng Chen and Wen-Hsing Hsu, "A modified fast parallel algorithm for thinning digital patterns," *Pattern Recognition Letters*, vol. 7, pp. 99-106, 1988.

[7] Gang Bai, "Multifont Chinese character recognition using side-stroke-end feature," *Proc. of the Second Int. Conf. on Document Analysis and Recognition*, 1993, pp. 794-797.

[8] Yun Wuu Wang, "Four-corner method," *Commercial Press, Taipei, Taiwan, ROC*, 1967 (in Chinese edition).

[9] Yiu-Man Tham and Tong Lee, "Four corner code based pre-classification scheme for Chinese character recognition," *ISIPNN '94: 1994 Int. Symposium on Speech, Image Processing and Neural Networks*, 1994, vol. 1, pp. 213-216.

[10] Hong-De Chang and Jhing-Fa Wang, "Preclassification for handwritten Chinese character recognition by a peripheral shape coding method," *Pattern Recognition*, vol. 26, no. 5, pp. 711-719, 1993.

[11] Hahn-Ming Lee and Chung-Chieh Sheu, "A handwritten Chinese characters recognition method based on primitive and fuzzy features via SEART neural net model," *1995 IEEE Int. Conf. on Systems, Man, and Cybernetics Intelligent System for the 21st Century*, 1995, vol. 2, pp. 1939-1944.

[12] Kunihiko Fukushima and Nobuaki Wake, "Handwritten alphanumeric character recognition by the Neocognitron," *IEEE Trans. on Neural Networks*, vol. 2, no. 3, pp. 355-365, 1991.

[13] Hans-Jürgen Zimmermann, "Fuzzy set theory and its applications," *Kluwer Academic Publishers*, 2nd ed., 1991.

[14] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen, "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759-771, 1991.

[15] Gail A. Carpenter, "Neural network models for pattern recognition and associative memory," *Neural Networks*, vol. 2, pp. 243-257, 1989.

[16] Dan Hammerstrom, "Neural networks at work," *IEEE Spectrum*, June, p. 26-32, 1993.

[17] K. Yamamoto and A. Rosenfeld, "Recognition of hand-printed Kanji characters by a relaxation method," *Proc. 6th Int. Conf. on Pattern Recognition, Munich, Germany*, 1982, pp. 395-398.