Similar Handwritten Chinese Characters Recognition by Critical Region Selection Based on Average Symmetric Uncertainty

Bo Xu, Kaizhu Huang, Cheng-Lin Liu
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
P.O. Box 2728, Beijing 100190, PR China
Email:{box, kzhuang, liucl}@nlpr.ia.ac.cn

Abstract

We consider the problem of similar Chinese character recognition in this paper. Engaging the Average Symmetric Uncertainty (ASU) criterion to measure the correlation between different image regions and the class label, we manage to detect the most critical regions for each pair of similar characters. These critical regions are proved to contain more discriminative information and hence can largely benefit the classification accuracy for similar characters. We conduct a series of experiments on the CASIA Chinese character data set. Experimental results show that our proposed method is superior to three competitive approaches in terms of both accuracy and efficiency.

1. Introduction

The accuracy of Handwritten Chinese character recognition (HCCR) has been improved substantially from its initial stage of research. However, the improvement may be not enough yet to satisfy the requirements emerging from real applications. Various fundamental problems remain unresolved in HCCR. More particularly, how to distinguish similar characters is still a big challenge. In more details, similar Chinese characters usually share common radicals or have very subtle shape difference in local details. Moreover, the number of similar pairs in HCCR is huge and different similar pairs vary in the location of different strokes. These properties present big difficulties for similar Chinese character recognition.

There have been many proposals to deal with the problem of similar character recognition. One typical way is to adopt a hierarchical structure. Namely, in addition to a global classifier for recognizing normal characters, a local classifier is further engaged to discriminate those similar characters. In the simplest case, the local classifier discriminates only two classes. For example, Ishii Tsutomu [4] used neural networks as the two-class classifier and achieved very good recognition results. Jin's method [5] also showed success in this direction. The compound Mahalanobis function (CMF), proposed by Suzuki [9], making use of minor eigenvectors, can also discriminate pairs of similar characters. Gao et al. [1, 3] proposed the LDA-based compound distance approach that fuses distances in the original feature space and the local subspace.

Different from the previous approaches, in this paper, we propose a novel algorithm based on critical regions to classify similar pairs. Noting the fact that similar pairs usually share common radicals and are just different in some regions, we try to detect those regions which are critical for discriminating two similar characters. Take the similar pair of characters ”埃” and ”挨” as example in Fig 1. ”埃” and ”挨” have the same right radical ”矣”, but are different in the left. Hence we can easily distinguish ”埃” from ”挨” only by its left radical ”土” or recognize ”挨” from ”埃” by the left radical ”扌 ”. This motivates us to distinguish similar pairs by appropriately locating and exploiting the critical region information.

Apparently, the key problem is how to locate the critical region of different similar pairs automatically, since different similar pairs vary in the location of critical region. To solve this problem, we engage Average Symmetric Uncertainty (ASU) to detect critical regions automatically. ASU is a correlation metric used to measure the relevance between a region and the class label and hence can extract the regions which are mostly relevant to classification. After the critical region, often a subtle part of the whole image, is located, the features will be extracted merely from the region and then fed to classification. This strategy presents two appealing advantages. First, since non-cirtical regions basically cannot differentiate two similar characters, ignoring features from these regions could reduce the noise effects and increase the recognition accuracy. Second, exploiting only the critical regions will reduce the size of feature space and hence benefit the efficiency for later classification.

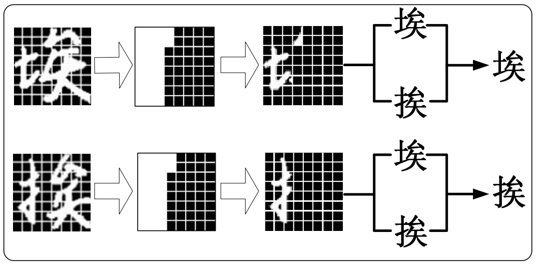

Figure 1. Example of our algorithm

It is noted that Leung [7] et al. also proposed a critical region detection method based on the output of the Fisher's discriminant. However, as shown in experiment, their criterion usually cannot accurately locate the critical regions and hence leads to limited performance.

In the rest of the paper, we will give an overview of our HCCR system in Section 2. The proposed critical region automatic detection and similar pair selection algorithm are described in Section 3. Section 4 presents our experimental results. Finally, Section 5 gives concluding remarks.

2. System Overview

The diagram of our HCCR system is shown in Fig. 2. The input character image is firstly normalized to a standard size. Then the gradient feature [8] is extracted. After dimension reduction by LDA, the low-dimensional feature is fed to the global classifier, the Modified Quadratic Discriminant Function (MQDF) [6]. MQDF outputs some candidates classes which have higher probabilities. If the value difference of top two candidates is below a predefined threshold, T, and the two candidates are similar pairs (similar pairs are searched in the training stage and saved in a database), then the local classifier is used to choose a class from the two candidates. The local classifier firstly extracts the features from the critical regions (determined in the training stage) and applies two-class LDA [3], then outputs the scores of pair-wise classifier, a two-class MQDF classifier. Finally, the scores of both the global and the local classifiers are fused to output the recognition result. If the output of global classifier cannot meet the above requirements, the top candidate given by MQDF will be output as the final recognition result.

3. Critical Region Based Similar Characters Recognition

In this section, we firstly present the definitions of Average of Symmetric Uncertainty (ASU) and Mean of ASU (M-ASU). Then we introduce our method, Similar Pairs Search (SPS) and ASU-based automatic critical region detection.

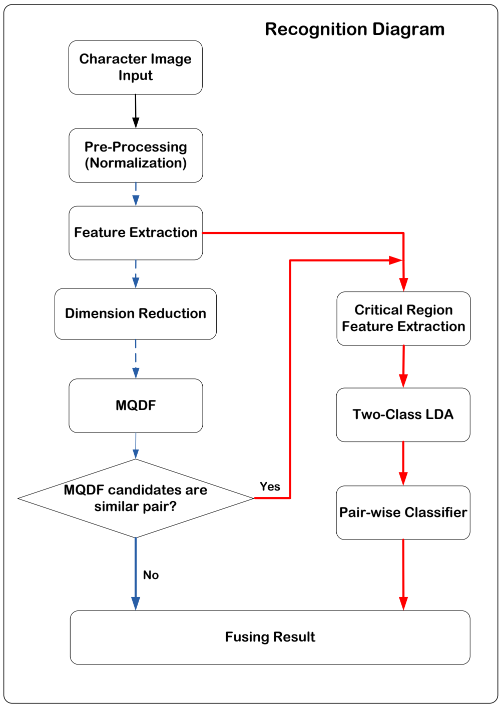

Figure 2. Diagram of HCCR system

3.1 Definition

We first present some definitions. After the normalization, the character image I is divided into $k \times k$ rectangle. Each of the rectangle, called *unit region*, is given a unique number, i, from 1 to N ($N = k \times k$). For example, the last region in the first row in Fig. 3 is defined as I_8, as k is 8 in our experiment. We specify a number of standard directions to decompose the gradient vector of arbitrary direction, e.g., eight directions in the paper, and let $j \in [1, 8]$ denote these standard directions. Our eight standard directions are illustrated in Fig. 3. Let X denote the gradient features of the character image and X_{ij} denote the gradient feature in the j-th standard direction of the i-th region. In addition, let Y denote the class label.

Symmetric Uncertainty [10], defined as the normalization of mutual information, is a measurement of uncertainty between two random variables. In feature selection, it can be used to measure the correlation between a feature and the class label.

$$SU\left(X,Y\right) = 2\left[\frac{I\left(X;Y\right)}{H\left(X\right)+H\left(Y\right)}\right] \qquad (1)$$

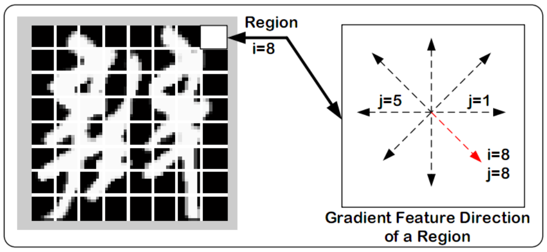

Figure 3. Region Illustration

SU is the symmetric uncertainty of the variable X and Y, $I(X;Y)$ is the information gain of X and Y. $H(X)$ and $H(Y)$ are the entropy of X and Y respectively.

Here we define the Average of Symmetric Uncertainty (ASU) to measure the dissimilarity of a unit region between similar pairs.

Definition 1 ***ASU (Average of Symmetric Uncertainty)*** *is defined as the Symmetric Uncertainty between a unit region and the class label, valued by the mean of SU_{ij}. The formulation of ASU is given as below:*

$$ASU_i = \frac{1}{8}\sum_{j=1}^{8} SU_{ij}, i \in [1, 64]. \tag{2}$$

$$SU_{ij} = 2\left[\frac{I(X_{ij};Y)}{H(X_{ij}) + H(Y)}\right]. \tag{3}$$

ASU measures the dissimilarity of a unit region between similar pairs. If the ASU value of a unit region is large, this signifies that the strokes in the unit region are more easily to distinguish the similar pairs; otherwise, the region may be less discriminative for differentiating two similar characters .

Definition 2 ***M-ASU (Mean of ASU)*** *is the mean of ASU in all the regions. It measures the similar degree of similar pairs. It is used as a filter in the paper to find the critical regions. The formulation of M-ASU is given as follows:*

$$MASU = \frac{1}{N}\sum_{i=1}^{N} ASU_i. \tag{4}$$

3.2. Similar Pair Selection

A similar pair is two characters which tend to be confused during recognition. In the hierarchical system, similar pairs are distinguished by a local classifier to improve the accuracy. However, a large number of similar pairs will bring heavy burden for the system. Hence we should balance the size of similar pairs and the recognition accuracy.

In this paper, we propose an algorithm that can find similar pairs effectively. The algorithm is listed in Table 1. Input parameters include the output scores of training samples, $S(Y_1, Y_2)$, the class number N, the absolute difference threshold AD and appearance times threshold AT. The output is the similar pair list (SPL). Before the description of the algorithm, we introduce the output scores firstly. We apply 5-fold cross validation in training data and record both the top three output scores and the corresponding candidate class label of each data sample, which is denoted by $S(Y_1, Y_2)$. Y_1 is the genuine class label, Y_2 is the estimated class label by the classifier. $S(Y_1, Y_2)$ is the output score. All the output scores are collected in the so-called score subset. Obviously, the number of the subset is the same as the number of training samples. We then take a specific class S_0 as an example to illustrate our SPS algorithm. To find the similar pairs of the class S_0, we firstly collect all the output scores from the score subset where $Y_1 = S_0$ and calculate the average score $\bar{S}(S_0, S_0)$. In order to decide whether the class S_0 and class S_i are a similar pair, we estimate the average score $\bar{S}(S_0, S_i)$ and the frequency of (S_0, S_i), denoted as $f(S_0, S_i)$. We take (S_0, S_i) as similar pairs if $|\bar{S}(S_0, S_0) - \bar{S}(S_0, S_i)|$, denoted as $D(S_0, S_i)$, is larger than AD and $f(S_0, S_i)$ is larger than AT; otherwise, (S_0, S_i) is not the similar pairs.

The number of similar pairs is controlled by parameters AD and AT. AD controls the similarity between similar pairs. AT controls the frequency of being misclassified or being easily misclassified. When AD is decreased and AT is increased, we select more similar pairs. The combination of AT and AD can help to find similar pairs that tend to be misclassified.

3.3. Automatic Critical Regions Detection

Human being could automatically locate the different strokes or regions of similar pairs to distinguish one from another. This inspires us to design a similar character recognition system that imitates the recognition process of human being. Thus we propose a novel algorithm based on Average Symmetric Uncertainty (ASU), a measurement between the feature from the unit region and the class label, to automatically detect the critical regions (different strokes or shape) of similar pairs.

Our algorithm is listed in Table 2. The input parameters include feature $X_{ij}^t, t = 1, 2, ...n$, which is extracted from n training images of a similar pair. The number of unit region N, the threshold α and class label $Y^t, t = 1, 2, ...n$. The output parameter is the critical region subset (CRS). Firstly, CRS is initialized as an

Table 1. Similar Pair Selection Algorithm

```
Similar Pair Selection (SPS)

Input: $S(Y_1, Y_2)$, class Number N, AD, AT

Output: Similar pair list SPL

Initialize: Set SPL = $\emptyset$

For $i = 1 : N$
    $S_0 = i$
    $S = \{S_1, S_2, ...S_k\}$ (Candidate Set)
    For $j = 1 : k$
        Compute $f\left(S_0, S_j\right)$ and $D\left(S_0, S_i\right)$
        if $f\left(S_0, S_j\right) > AT$ and $D\left(S_0, S_j\right) < AD$
            Updata SPL:
            if $[S_0, S_i]$ or $[S_j, S_0]$ not in SPL
                $SPL = SPL \cup \{[S_0, S_j]\}$
            end
        end
    end
end
```

empty set. After SU between feature X_{ij} and class label Y is evaluated, ASU in each unit region is computed by Eq. (2) and $M\text{-}ASU$ is estimated by Eq. (4). Then the threshold, T, for the detection of critical regions, is decided by Eq. (5). Next, we traverse all the unit regions to compare the ASU_i and threshold T. We then add the sequence number of the region whose ASU is higher than the threshold to CRS. When the traverse is ended, CRS selects all the sequence number of the critical regions.

$$T = \alpha * MASU, \alpha > 0. \tag{5}$$

4. Experiments

In this section, we compare the recognition performance of our method with three competitive methods, the traditional MQDF [6], the LDA-based compound distance method [3], and the method proposed in [7] on the CASIA data set. As the LDA-based compound distance method [3], and the method proposed in [7] contain different parameters, for fair comparisons, we conduct evaluations of our method separately with these two methods. We first report the experimental setup in the following.

4.1 Data and Pre-processing

We exploit the CASIA data set for comparison. The CASIA data set was collected by the Institute of Automation of Chinese Academy of Sciences, contains 3755 Chinese characters of the level-1 set of the standard GB2312-80, 300 samples per class. We choose 250 samples per class for training and the remaining 50 samples per class for testing.

Table 2. Automatic Critical Region Detection Algorithm

```
Critical Region Automatic Detection Algorithm

Input: feature $X_{ij}$, unit region number N, $\alpha$, class Label Y
Output: Critical Regions subset (CRS)

Initialize: Set CRS= $\emptyset$

1.Compute $SU$ between feature $X_{ij}$ and class Label $Y$

    $SU_{ij} = SU\left\{X_{ij}, Y\right\}$    (6)

2.Use Eq. (2) to estimate $ASU_i$ of each zone.
3.Use Eq. (4) to estimate $MASU$.
4.Set $T = \alpha * MASU$

Update CRS :
for $i = 1 : N$
    $if\ ASU_i > T$
        $CRS = CRS \cup \{i\}$
    end
end
```

During the pre-processing and feature extraction, each binary image was normalized to gray-scale image of 64 × 64 pixels by the bi-moment normalization methods. Then the 8-direction gradient direction features were extracted. The resulting 512-dimensional feature vector was projected onto a 160-dimensional subspace learned by the global LDA. The 160-dimensional projected vector was then fed to the MQDF classifier. For similar characters discrimination, features from differential regions were firstly extracted, then fed into the two-class LDA classifier. The final results were given by the compound distance of MQDF and two-class LDA classifier.

4.2. Parameter Setup

Three types of parameters need to be set in our system. They are the parameters for similar pairs searching, critical region detection, and final results fusion. We implement three types of experiments to search the best parameter sets.

Firstly, we investigate the impacts of AT and AD in

Table 3. Similar pair number based on different AT and AD

AD	$AT = 5$	$AT = 10$	$AT = 20$	$AT = 50$
30	171	162	160	154
70	8257	6497	4997	3280
100	23002	16026	10909	5960
200	32884	21296	13512	6784
500	32903	21307	13519	6800

Table 4. Recognition rate on different AT and AD

AT	AD =5	AD =10	AD =20	AD =50
100	98.46%	98.46%	98.38%	98.38%

our Similar Pairs Selection algorithm. We vary AT and AD to filter the similar pairs. The number of similar pairs with the different parameter is listed in Table 3 and the recognition rate is listed in Table 4. To balance the number of similar pairs and the accuracy of recognition, in our system, the parameters (AT, AD) is finally set to $(100, 10)$.

We then examine the effects of the Automatic Critical Regions Detection threshold, α. We set α to 0,0.8, 1.0 and 1.2 and the experimental results are in listed in Table 5. Obviously, $\alpha = 0.8$ generally outperforms the other values.

Finally, we justify the impacts of fusion parameter β. The final recognition result is the fusion of the outputs from the global classifier and the local classifier if the local classifier is used. We apply the fusing algorithm in [3].

$$\left\{ \begin{array}{l} S\left(X, Y_i\right) = (1 - \beta) * S_1\left(X, Y_i\right) + \beta * S_2\left(X, Y_i\right) \\ S\left(X, Y_j\right) = (1 - \beta) * S_1\left(X, Y_j\right) + \beta * S_2\left(X, Y_j\right) \end{array} \right. .$$

We vary β from 0 to 1 with the step 0.1. Experimental results are also listed in Table 5. The results reveal that $\beta = 0.5$ is the optimal choice for the final fusion.

Table 5. Recognition rate using different α and β. d means the average feature dimension.

β	$\alpha = 0$	α =0.8	α =1.0	α =1.2
0	98.32	98.35	98.30	98.20
0.1	98.35	98.39	98.36	98.28
0.2	98.38	98.43	98.40	98.30
0.3	98.41	98.44	98.43	98.37
0.4	98.43	98.47	98.46	98.40
0.5	98.46	98.48	98.46	98.40
0.6	98.45	98.45	98.43	98.38
0.7	98.45	98.45	98.43	98.38
0.8	98.37	98.32	98.38	98.34
0.9	98.21	98.16	98.14	98.11
1	97.89	97.89	97.89	97.89
d	512	260	198	149

Table 6. Recognition rate (%) under different K. $d = d' = 198$.

K	MQDF	Gao et al.'s method	Our method
K=10	97.66	98.26	**98.36**
K=20	97.89	98.36	**98.46**
K=30	98.01	98.39	**98.50**
K=40	98.06	98.42	**98.53**
K=50	98.04	98.41	**98.53**

4.3. Comparison with [2]

We firstly compare the recognition accuracy of our system with the LDA-based Compound distance approach [2]. This algorithm distinguishes similar pairs by projecting features to a subspace learning by global LDA. Our method classifies similar pairs by extracting features from critical regions. For a fair comparison, the dimension of local feature subspace d in [2] is set to the same value as the average number of features from critical regions for all of the similar pairs in our method. By varying K, the number of the principle vectors of the MQDF global classifier, we obtained the recognition accuracy on CASIA as shown in Table 6. From Table 6, we can see that our method outperforms the MQDF+MD method from $K = 10$ to 50. The corresponding similar pair number is listed in Table 7. As observed, our approach uses much fewer similar pairs, but achieves better performance than Gao et al.'s approach [2]

In order to further examine the performance of our proposed approach against [2] in different local feature subspaces, we fix K and vary d. Table 8 shows the recognition rate under different d's. The dimension of d is decided by setting the threshold parameter α to $0.8, 1.0$ and 1.2. In each experiment, we see that our approach is always better than Gao et al.'s approach.

4.4. Comparison with [7]

As mentioned before, Leung et al. also proposed a method to detect critical regions for similar character recognition [7]. To evaluate the performance of our approach against their method, we conducted another exper-

Table 7. Similar pair number with varying K

Method	K=10	K=20	K=30	K=40	K=50
Gao et al.	70098	70904	71867	66378	71784
Our	**15910**	**16026**	**17245**	**17369**	**17369**

Table 8. Recognition rate (%) based on different d.

Method	$d = 149$	$d = 198$	$d = 260$
Gao et al.	98.32	98.36	98.39
Our	**98.40**	**98.46**	**98.48**

Table 9. Recognition rate (%) of critical region detection algorithm.

Algorithm	$d = 129$	$d = 176$	$d = 240$	$d = 512$
Leung et al.	95.92	97.18	97.88	98.56
Our	**98.70**	**98.67**	**98.52**	98.56

iment. [1] We compare the recognition accuracy and the average time of these two different algorithms on similar pairs. Firstly, we choose 1093 similar pairs from CASIA by setting parameter $(AD, AT) = (50, 100)$. Thus the number of total training samples reaches to $1093{*}250{*}2 = 546500$ and corresponding testing number is $1093{*}50{*}2 = 109300$. Then the critical regions of each similar pair are detected by different algorithms. In [7], features are extracted from the regions with larger absolute weights of two-class LDA projection vectors. Thus we compute the projection vector by Eq. (7)

$$\begin{aligned}\omega &= \mathrm{S}_w^{-1}\left(m_i - m_j\right)\\ &= \sum_{n=1}^{d}\frac{1}{\lambda_n}\phi_n\phi_n^T\left(m_i - m_j\right), \lambda_n > 0.1\end{aligned} \; . \qquad (7)$$

S_w is the within-class scatter matrix, m_i and m_j are class center, λ_n and ϕ_n are eigenvalue and eigenvector of S_w. Next the gradient features from those regions are fed to the two-class LDA classifiers for training. During testing, for each two-class classifier, samples from those two classes in the test data set are collected. Then the features from the critical regions are extracted and are fed to the two-class classifier. The average recognition accuracy of all the two-class classifiers is taken to compare the effectiveness of critical region detection algorithm. Meanwhile, the average time for detecting a similar pair is recorded.

From the results, obviously, our algorithm is better in terms of both the recognition accuracy or the detecting time. Especially, the accuracy achieves 98.70%, which is higher than the accuracy using all the features. In addition, Leung et al.'s algorithm computed the S_w^{-1} to detect the critical regions, which takes more time than computing the symmetric uncertainty as in our algorithm.

5. Conclusion

In this paper, we proposed a novel method to distinguish the similar characters by features from the critical regions. The critical regions of similar pairs were automatically detected by our algorithm based on Average Symmetric Uncertainty (ASU). Furthermore, we also presented an algorithm, SPS, to effectively select the similar pairs. Experiments on CASIA demonstrated the superiority of our method over both the traditional MQDF and the other two competitive approaches.

Table 10. Computational time ($sec.$) of different critical region detection algorithms.

Algorithm	$d = 129$	$d = 176$	$d = 240$
Leung et al.	5.69	5.67	5.62
Our	**0.34**	**0.35**	**0.34**

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no.60825301 and no.60933010.

References

[1] T.-F. Gao and C.-L. Liu, "LDA-based compound distance for handwritten Chinese character recognition", *Proceedings of Ninth International Conference on Document Analysis and Recognition*, 2007, volume 2, pp 904–908.

[2] T.-F. Gao and C.-L. Liu, "Combining Quadratic Classifier and Pair Discriminators by Pairwise Coupling for Handwritten Chinese Character Recognition", *Proceedings of 19th International Conference on Pattern Recognition*, 2008, pp 1–4.

[3] T.-F. Gao and C.-L. Liu, "High accuracy handwritten Chinese character recognition using LDA-based compound distances", *Pattern Recognition*, 41(11):3442–3451, 2008.

[4] T. Ishii, Y. Waizumi, N. Kato and Y. Nemoto, "Recognition System for Handwritten Characters by Alternative Method using Neural Network.", *IEICE Transactions on Information and Systems*, 83(3):988–995, 2000.

[5] Y. Jin and S. Ma, "Pairwise classifier combination and its application on Chinese character recognition", *Proceedings of Fifth World Congress on Intelligent Control and Automation*, 2004, volume 5, pp 4075–4078.

[6] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):149–153, 1987.

[7] K. Leung and C. Leung, "Recognition of handwritten Chinese characters by critical region analysis", *Pattern Recognition*, 43(3):949–961, 2010.

[8] C.-L. Liu, "High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction", *Proceedings of 18th International Conference on Pattern Recognition*, 2006, volume 2, pp 942–946.

[9] M. Suzuki, S. Ohmachi, N. Kato, H. Aso and Y. Nemoto, "A discrimination method of similar characters using compound Mahalanobis function", *Trans IEICE Japan*, J80-D-II(10):2752–2760, 1997.

[10] I.-H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with Java implementations", *ACM SIGMOD Record*, 31(1):76–77, 2002.

[1] [7] also partitions the critical regions into finer cells for extracting detailed features. Here we only compare with its critical regions detection algorithm.