A Radical Cascade Classifier for Handwritten Chinese Character Recognition

Enzhi Ni and Changle Zhou
Mind, Art and Computation Lab, Fujian Key Lab of the Brain-like Intelligent Systems, School of Information Science and Technology, Xiamen University, Xiamen, China
Email: nienzhi@gmail.com, dozero@xmu.edu.cn

Minjun Jiang
School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China
Email: anna_xmu@sina.com

***Abstract*—Radical extraction is the core technique for radical-based Chinese character recognition. In this paper, we proposed a new method of radical extraction – radical cascade classifier. The radical cascade classifier consists of multiple AdaBoost classifiers. It can detect and extract specific radical from characters. To apply cascade classifier to radical extraction, we focus on two main points: feature selection and radical detection. In this paper, we discussed feature selection for the radical cascade classifier and proposed two methods of radical detection. Based on these works, we constructed the radical cascade classifier and conducted experiments on HITPU databases. The experimental results have shown that our approach is efficient.**

***Index Terms*—handwritten Chinese character recognition, radical extraction, cascade classifier**

I. INTRODUCTION

Radical-based Chinese character recognition is a promising approach for Chinese character recognition. Since the Chinese character has a great amount of categories, the Chinese character recognition is one of the toughest classification tasks. Compared with its large number of categories, the Chinese character is composed of several hundred “elements” called radicals. Based on this observation, the radical-base approach is proposed, which adopts the strategy “divide and conquer”, dividing the character into its radicals and recognizing each of them.

Radical extraction is the core technique of the radical-based approach. Previous works of radical extraction fell into two categories, offline radical extraction and online radical extraction.

In the field of offline Chinese character, Wang and Fan [1] proposed a recursive hierarchical scheme for radical extraction of Chinese character. The proposed scheme includes three modules, which are character pattern detection module, straight cut line detection module, and stroke clustering module. Chung and Ip [2] proposed a radical segmentation algorithm based on D-snakes. D-snakes are used to locate the gaps between radicals, so that characters can be segmented into radicals according to these gaps. Shi et al. [3] used active shape models to extract radicals from Chinese characters. A set of examples of radicals is first represented by landmark points, then radicals are modeled as active shapes using kernel PCA, and finally unseen radicals are matched to the reference models using a genetic algorithm to search for the optimal shape parameters. Chellapilla and Simard [4] proposed the radical-at-location neural network, which is a convolutional neural network that is designed to process the whole character image and recognize radicals at a specific location in the character? Ni et al. [5] proposed a two-stage heuristic search algorithm to extract radicals. The first stage search is applied to extract candidate radicals from a character. And the second stage search is used to find a radical combination that best matches the character among the candidate radicals.

In the field of online Chinese recognition, Xiao and Dai [6] implemented a radical matching algorithm with dynamic programming (DP), but their method only extracts the front radical and the rear radical in a character. Lv et al. [7] segmented radicals by stroke projection. Ma’s method [8] proposed a method for online Chinese character recognition, which includes three aspects: statistical classification of radicals, over-segmentation of characters into candidate radicals and lexicon-driven recognition of characters. So far it is only applied to the characters of left-right structure. Ni et al. [9] also applied the two-stage heuristic search algorithm to online radical extraction.

In this paper, we propose a method of radical extraction in handwritten Chinese character by using radical cascade classifier. Although it has been successfully applied in some fields, cascade classifier is for the first time used in radical extraction. This paper focused on the two main questions: feature selection and radical detection. To choose an efficient feature for the cascade classifier, we present the criteria of feature selection. Two methods of radical detection are presented in this paper. The experiments are conducted on HITPU [3], a loosely constrained handwritten Chinese character database. Our method has achieved average detection rate

doi:10.4304/jsw.7.10.2294-2300

over 96% and average detection time of 0.040s per characters.

The rest of the paper is organized as follows. The cascade classifier and AdaBoost algorithm are introduced in Section II. Feature selection is discussed in Section III. In section IV, the methods of radical detection are presented. The final two sections are experimental results and conclusion.

II. Cascade Classifier And Adaboost Algorithm

A. Cascade Classifier

Viola and Jones [10] proposed cascade classifier. In the beginning, it was used to rapid face detection. For its excellent performance, it has been applied to varieties of tasks. The idea of the cascade classifier is that a lot of simple and weak classifiers can make up complex and strong classifiers. This method allows non-object regions of an image to be quickly discarded and spends more computation on promising object-like regions. The strong classifiers then form a degenerated tree, which is a cascade classifier. Fig. 1 shows the structure of a cascade classifier.

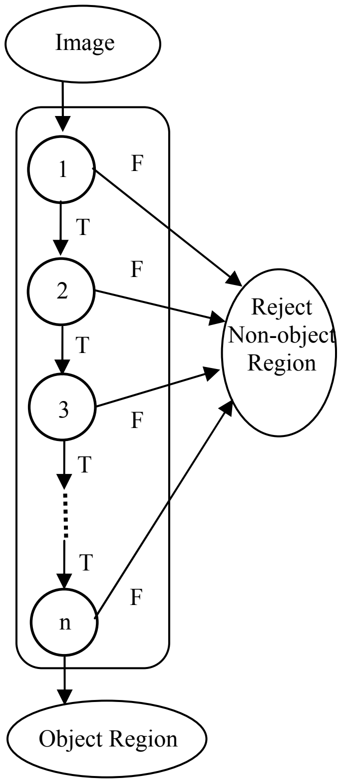

Figure 1. The structure of cascade classifier

In Fig. 1, each stage in the cascade is a strong classifier. The stages are constructed by training classifiers using AdaBoost algorithm and then adjusting the threshold to minimize false negatives. In general, a lower threshold yields higher detection rates and higher false positive rates. Note that the default AdaBoost threshold is designed to yield a low error rate on the training data, therefore the default AdaBoost need to be modified for training cascade classifier [10].

B. AdaBoost Algorithm

The AdaBoost algorithm was introduced by Freund and Schapire [11]. The idea of AdaBoost is to use the weak classifiers to form a highly accurate classifier (strong classifier) by learning weak classifiers repeatedly on different distributions over the training examples. Provided the error rate of these weak classifiers are less than 1/2 (with respect to the distribution on which it was trained), the error rate of the strong classifier decreases exponentially. The AdaBoost has some extensions such as real AdaBoost, LogitBoost and gentle AdaBoost [12]. They have very similar scheme. Gentle AdaBoost is used in the paper. And as mentioned above, the gentle AdaBoost needs to be modified. A modified gentle AdaBoost algorithm can be depicted as follows:

Algorithm 1: Modified gentle AdaBoost algorithm

Given maximum false detection rate $f_{\max}$, minimum detection rate $d_{\min}$ and example $(x_1, y_1), ..., (x_N, y_N)$ where y_i =0,1 for negative and positive examples respectively.

Let f be false detection rate, d be detection rate, t be threshold of classifier.

1. Start with weights $w_i = 1/N, i = 1, 2, ..., N, m = 0,$ $f = 1, H(x) = 0$.
2. If $f > f_{\max}$, $m = m + 1$ and repeat:
   a. Train a weak classifier $h_m(x)$ by weighted least-squares of y_i to x_i with weights w_i , where $h_m(x) = P_w(y = 1 | x) - P_w(y = 0 | x)$.
   b. Update $w_i \leftarrow w_i \exp(-y_i h_m(x_i))$ and renormalize
   c. The current strong classifier is $sign[H(x)] = sign[\sum_{k=1}^{m} h_k(x) - t]$ where t is updated to ensure $d > d_{\min}$.
   d. Update f .
3. Output the strong classifier $sign[H(x)] = sign[\sum_{k=1}^{m} h_k(x) - t]$.

This algorithm trains a strong classifier in example $(x_1, y_1), ..., (x_N, y_N)$, where x is the feature vector of an example, y is the label of an example. Given maximum false detection rate $f_{\max}$ and minimum detection rate $d_{\min}$, the detection rate of the strong classifier has to be greater than $d_{\min}$ after each training. Train the strong classifier until the false detection rate is less than $f_{\max}$. In Step 2, the weak classifier

$h_m(x) = P_w(y=1|x) - P_w(y=0|x)$, where $P_w(y|x)$ is the weighted class probability. We can approximately calculate $P_w(y=1|x)$ and $P_w(y=0|x)$ from the distribution of training set. $h_m(x)$ has already been determined by the training set, the weights of training samples and the form of $h_m(x)$. Therefore, the threshold of the strong classifier t has to be adjusted in order to ensure current detection rate of the strong classifier is greater than $d_{\min}$. During each round of Step 2, the weights of examples are updated to $w_i \exp(-y_i h_m(x_i))$. By updating w_i, training weak classifiers will focus on those examples which still can not be classified correctly.

III. Feature Discussion

A variety of features have been used to recognize handwritten Chinese characters, but it is uncertain whether they are suitable for training the cascade classifier. We argue that a qualified feature satisfies the following three requirements:

1. The feature has to be high dimensional;

As mentioned above, the cascade classifier is formed by many strong classifiers, which are made up of lots of weak classifiers. A weak classifier corresponds to at least one dimension of the feature vector. Consequently, only high dimensional features are able to provide enough weak classifiers. Note that, in the paper the weak classifier is in the form of tree structure which has only one root node and two leaf nodes.

2. The feature has to be scalable or approximately scalable.

For radicals in handwritten Chinese characters are not in fixed sizes, cascade classifier needs to detect in the same images of different sizes. However, image scaling is time-consuming. Thus this method doesn't suitable for rapid radical detection. Instead of scaling the image, we can scale the detection window of the cascade classifier so that the cascade classifier can rapidly detect radicals in different sizes. For this reason, the feature has to be scalable.

Definition 1. *Let $F = (f_1, f_2, ..., f_n)$ be a feature vector of image I. After I scaled, the feature became F'. If for arbitrary scaling, there exists $F' = \lambda F$, where λ denotes arbitrary positive number and is directly proportional to the scale factor, the feature is scalable.*

It is hard to find a feature strictly satisfies above definition. In fact, the feature can be used to train cascade classifier as long as it is approximately scalable.

3. Since handwritten characters have much distortion, the feature must have the ability to reduce the influence of the distortion.

According to the above three criteria, we choose one suitable feature that is used in our methods. The following discussion does not intend to cover many features but two typical features in the field of handwritten Chinese character recognition and object detection.

A. Gradient Feature

The gradient feature is frequently used in handwritten recognition [13, 14, 15]. It can be obtained from gray images and binary images and is not sensitive to noise. Several classifiers based on gradient feature yield very high accuracy in handwritten Chinese character recognition. High dimensional gradient feature can be obtained from images. For example, as to a 20×20 image, the gradient is divided into 8 quantized directions, and then sample the gradient in every direction. Provided the side length of sampling window ranges from 4 to 20 and sampling interval is 1 (see Fig. 2), the dimension of the gradient feature will be 14280. Thus the gradient feature satisfies the first requirement.

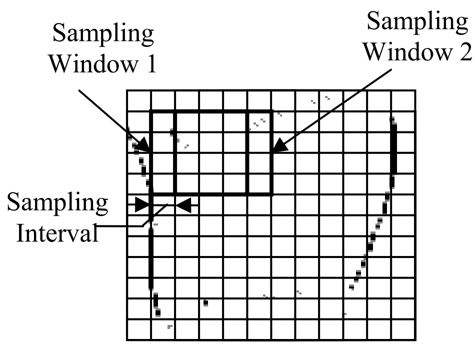

Figure 2. Illustration of sampling the gradient.

It shows the gradient feature of radical "kou" in the direction of 0°. Sampling window 1 and sampling window 2 are two square windows whose side lengths are 4. Sampling interval is 1.

However, the gradient feature does not satisfy the second requirement. We sample the gradient feature of radical "kou" in the direction of 45°. Sampling conduct on two same images in the sizes of 120×120 and 60×60. Sampling windows are 20×20 and 10×10 respectively. After that, we obtained two gradient features which have 36 dimensions. The result is shown in Fig. 3. We can see from this figure that the proportions between corresponding dimensions of the two feature vectors vary considerably. Thus the gradient feature does not satisfy the second requirement.

B. Haar-like Feature

Haar-like features were originally used in face detection [10] and gave good performance. The features are made up of several rectangles (see Fig. 4). The value of the features is the difference of the weighted sums of the pixels between black and white rectangular regions. The weights are in reciprocal proportion to the areas of rectangles. In Fig.4 there are 12 haar-like features which are commonly used.

Haar-like features are high dimensional. For example, as to a 20×20 image, the number of its haar-like features is more than 120,000. Meanwhile, haar-like features are scalable. For example, we obtain rectangular features from radical "kou". The two same images of radical "kou" are in the sizes of 120×120 and 60×60. The sizes of the rectangles are 20×20 and 10×10 respectively. The sampling intervals are 20 and 10 respectively. We obtained 36 rectangular features. The result is shown in Fig. 5. We can see that corresponding rectangular features are in proportion to their areas (4:1 in this case). Haar-like features are computed from these rectangular features according to the definition of the haar-like feature, so haar-like features are scalable.

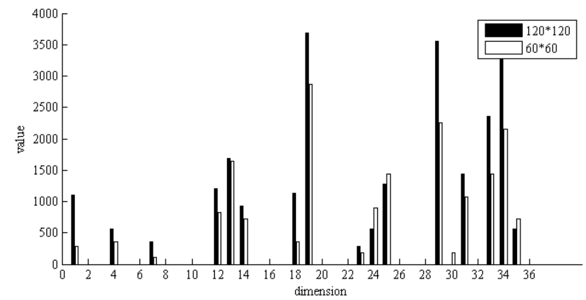

Figure 3. The sampling gradient features of two same images of two different sizes

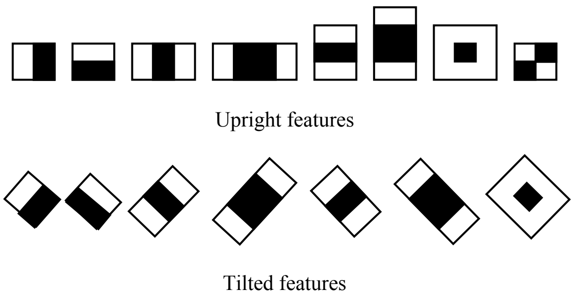

Figure 4. Haar-like Features

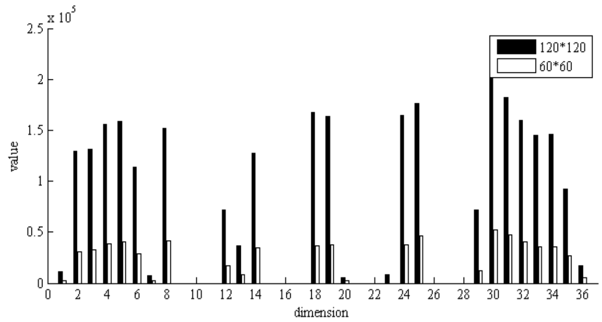

Figure 5. The rectangular features of two same images of two different sizes

Moreover, haar-like features have an advantage in radical detection. As we know, there is much distortion on handwritten Chinese characters. Over-accurate features are likely to yield bad results. Some methods, such as blurring images, character regulation and flexible templates, can reduce the influence of the distortion. As to haar-like features, they can tolerate the distortion by expanding their rectangles. The expansion has the same effect with flexible features. As shown in Fig. 6, the haar-like feature is expanded to tolerate the difference among the three radicals due to the handwritten distortion. Thus haar-like features satisfy all three requirements we mentioned above.

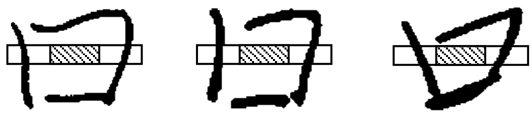
Figure 6. The same haar-like features of three different radical "kou"

Compared with the gradient feature, haar-like features are scalable, which guarantees the high speed of radical detection. And also they can reduce the influence of the distortion by expand their rectangles. Therefore haar-like features are used in this paper.

IV. RADICAL DETECTION

Radical detection is to search for all regions matched with a specific radical template within an image. In this paper, a radical template is a radical cascade classifier, which is composed of multiple strong classifiers and a lot of weak classifiers. Therefore radical detection is to apply a radical cascade classifier at sub-regions within an image, and output regions passed by the radical cascade classifier and discard regions rejected. The region passed is called radical-object-like window.

Radicals are flexible in handwritten Chinese characters. They have the following characteristics: the positions of the same radicals in different characters can be different (see Fig. 7a, 7b, 7c); the same radicals can exist in different positions in a character (see Fig. 7d); even the sizes of the same radicals can be different when the radicals are in different positions. In addition, the aspect ratios of the same radicals might change greatly (see Fig. 7a, 7b, 7e).

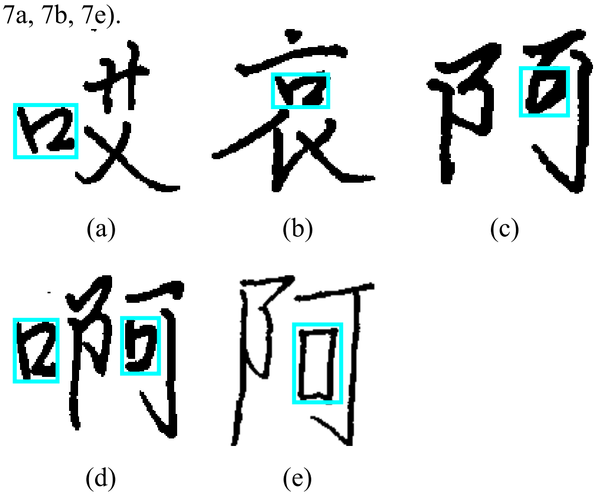

Figure 7. Radical "kou" in different characters

Because of these characteristics, radical detection has to be carried out in the whole images of Chinese characters by using radical templates in different sizes and aspect ratios. Changing the size or aspect ratio of a template is to change the sizes or aspect ratios of all features in the template. Since haar-like features are scalable, the templates using haar-like features are scalable. As a result, the templates in different sizes and aspect ratios can be generated from the original template by scaling all features in the original template.

By using integral image, haar-like features can be computed very rapidly. However, after the aspect ratios of templates being changed, the tilted haar-like features are not rectangular anymore and their tilt angles are also changed. As a result, their values cannot be computed with the original integral graph. Computing integral graphs with different tilt angles needs lots of time. So it is not suitable for rapid radical detection. There are two solutions for this problem. One is to change the aspect ratio of character images instead of the aspect ratios of radical templates (see Fig. 8); the other is to avoid using the tilted haar-like features (see Fig. 9). Obviously, the first one reduces the efficiency of detection, since image transformation and computing integral graph are both time-consuming. And the second one discards many efficient weak classifiers. We implemented these two methods and made a comparison between them in the following experiments. Two algorithms of these two methods are given as below.

Algorithm2: Radical detection algorithm (method 1)

Given an image *I* and a radical template *T*.

Let *R* be the range of aspect ratio, *S* be the range of scale.

1. For $r \in R$

a. Change the aspect ratio of image *I* to *r* and get image I'.

b. Compute the integral graph g of image I'.

c. For $s \in S$

(1) Scale radical template *T* with *s* and get radical template T' with size $w \times h$.

(2) For each position (*x*, *y*) at image I'

(a) Match the image within window (*x*, *y*, *w*, *h*) with template T', if they are matched, add this window to radical-object-like window set W.

2. Combine neighbor radical-object-like windows in W and output W.

Algorithm3: Radical detection algorithm (method 2)

Given an image *I* and a radical template *T*.

Let *R* be the range of aspect ratio, *S* be the range of scale.

1. Compute the integral graph g of image *I*.

2. For $(r, s) \in R \times S$

a. Change the aspect ratio of template *T* to *r*, scale it with *s* and get radical template T' with size $w \times h$.

b. For each position (*x*, *y*) at image *I*.

(1) Match the image within window (*x*, *y*, *w*, *h*) with template T', if they are matched, add this window to radical-object-like window set W.

2. Combine neighbor radical-object-like windows in W and output W.

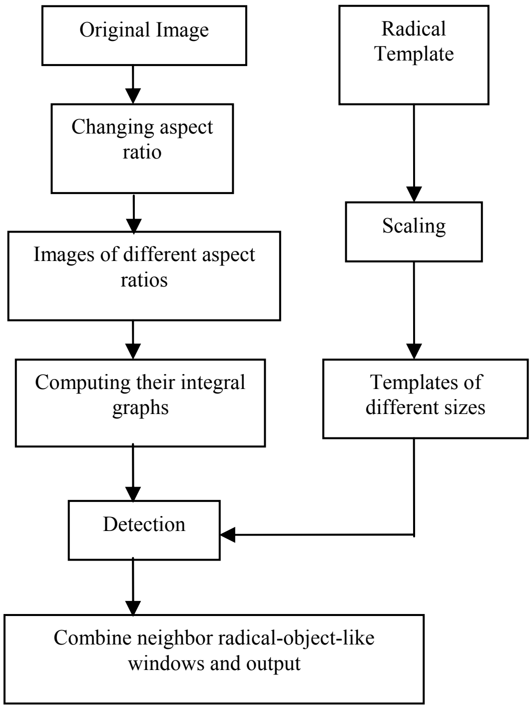

Figure 8. Method 1: detection in the same images of different aspect ratios

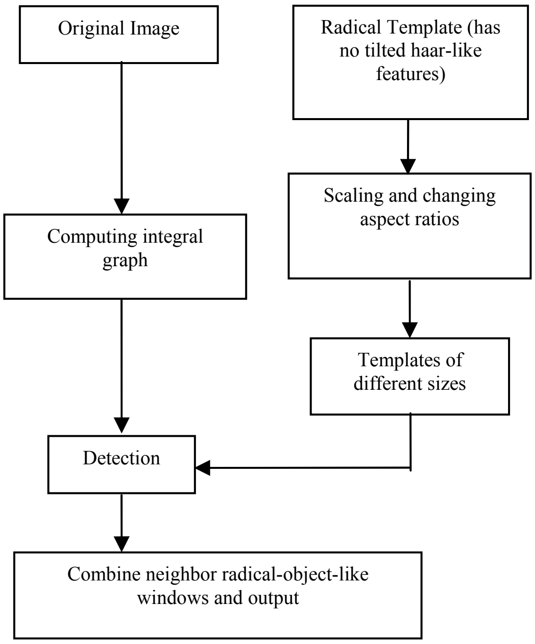

Figure 9. Method 2: detection with the same templates of different aspect ratios

The ranges of the aspect ratios and sizes of different radical categories vary widely. For instance, radical “kou” (口) can be narrower or wider than radical “wang” (王). To set a uniform range for all radicals is not ideal. For this reason, we calculated the ranges of each radical category from their own training samples.

V. EXPERIMENTS AND RESULTS

Experiments are performed on Intel Pentium 4 CPU (3.00GHz) equipped with 1 Gigabytes RAM. The source code is written in C++ with OpenCV (a computer vision library originally developed by Intel). All experimental samples come from HITPU, a database collected by Harbin Institute of Technology and Hong Kong Polytechnic University, which comprises a collection of 751,000 loosely constrained handwritten Chinese characters, consisting of 3755 categories written by 200 different writers.

We choose the following six radical categories: “kou” 口, surrounding “kou” (囗), “mu” (木), “wang” (王), “yu” (玉) and “zu” (足). The reason for choosing these radical categories is that they have much distortion in handwritten characters. Besides, as to radical “kou” (口), “mu” (木) and “wang” (王), their sizes and aspect ratios could change greatly. And they can appear at many positions of characters. As to radical surrounding “kou” (囗), it is a surrounding radical, which contains an inner radical.

All training radical samples are manually extracted from HITPU. The training set for each radical consists of 1000 positive samples (500 samples for radical “yu” 玉) which are radical images and 6000 negative samples which are character images not contain this radical. The test set for each radical consists of 1000 samples which contain this radical (53 samples for radical “yu” 玉) and 5000 samples which do not. The test samples cover a large number of character categories from different writers.

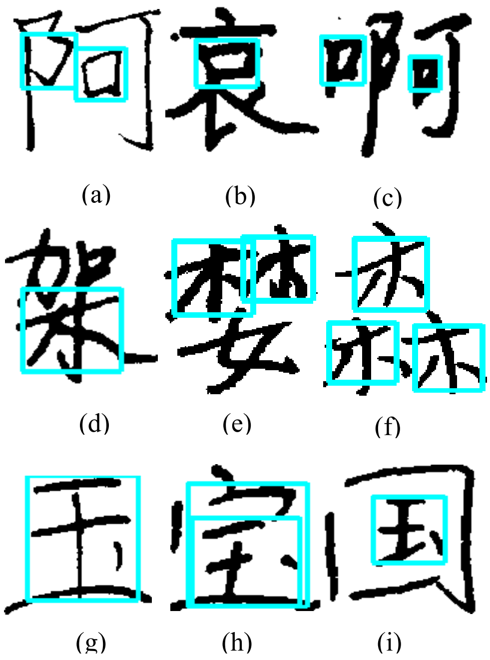

Figure 10. Some results of radical detection

Our proposed method has achieved detection rates of 94.29% (method 1) and 96.14% (method 2) and average detection time per character of 0.085s (method 1) and 0.040s (method 2). The detailed experimental results are showed in Table 1 and Table 2. In the tables, the object-radical-like windows denote the areas of detected radicals. The results show that our methods are effective to extract radicals from characters. Besides, the results also show method 2 is more rapid than method 1, meanwhile its performance is slightly better than method 1. Some results of radical detection are showed in Fig. 10 and some haar-like features used in the cascade classifier of radical “kou” are showed in Fig. 11.

Figure 11. Some haar-like features used in the cascade classifier of radical “kou”

Fig. 10 includes five situations in which radicals are hard to be extracted.

- The radicals have different sizes and aspect ratios (fig. 10a, 10b, 10d, 10e).
- The radicals have great distortion (fig. 10c, 10f).
- There is adhesion within radicals (fig. 10b, 10d, 10e).
- There are more than one same radical in a character (fig. 10c, 10e, 10f).
- Inner radicals (fig. 10i).

TABLE I.

EXPERIMENTAL RESULTS OF METHOD 1

Radical	**Detection rate (%)**	**Avg. detection time per character (s)**	**Avg. number of object-radical-like windows per character**	
			samples containing object radical	**samples not containing object radical**
口	91.70	0.144	2.02	1.49
Surrounding 口	98.51	0.100	1.44	0.88
木	95.35	0.113	1.69	0.66
王	94.73	0.039	2.26	1.06
玉	96.23	0.067	0.96	0.09
足	89.19	0.047	2.39	1.61

The results in Fig. 10 illustrate the distortion and adhesion has less influence on the performance of our method. Our method can detect radicals in different sizes and aspect ratios. And also our method is able to detect radicals in random positions of characters, including surrounding radicals and inner radicals.

TABLE II.

EXPERIMENTAL RESULTS OF METHOD 2

Radical	Detection rate (%)	Avg. detection time per character (s)	Avg. number of object-radical-like windows per character	
			samples containing object radical	samples not containing object radical
口	94.50	0.076	2.41	1.81
Surrounding 口	98.86	0.041	1.44	0.81
木	94.03	0.055	1.71	0.70
王	97.99	0.017	2.34	1.02
玉	94.34	0.030	1.06	0.31
足	97.10	0.019	1.88	0.71

The haar-like features showed in Fig. 11 are in accordance with our knowledge about the structure of radical "kou" (口). They represent the most typical characteristics of this radical, so they can be used in the radical cascade classifier and yield high accuracies.

VI. Conclusions

Handwritten Chinese character recognition is a very complicated classification task. The main cause is that Chinese characters have a large amount of categories. Radical-based approach regards a character as a combination of some radicals, recognizing the whole character by recognizing radicals within the character. As a result, the classification task with a large amount of categories is transformed to an easier classification task with only several hundred categories.

The most important step of radical-based approach is radical extraction. In this paper, we proposed a method of radical extraction that detects radicals within characters by using the radical cascade classifier. The method includes two parts: the construction of the radical cascade classifier and the methods of radical detection. The radical classifier consists of multiple strong classifiers, which are composed of a lot of weak classifiers. A weak classifier is represented by a haar-like feature. We construct the radical cascade classifier by using AdaBoost algorithm. Two methods of radical detection are proposed according to the characteristics of radicals. They adopt two different ways to ensure the high speed of radical detection. Experiments conducted on HITPU database showed our method has a good performance, especially in some difficult situations.

Acknowledgement

The work described in this paper was supported by National Nature Science Foundation of China under grant No. 60975076 and grant No. 60903129. Thank Prof. Shi for providing HITPU database to us, which is very helpful to our research.

References

[1] A. B. Wang, K. C. Fan, "Optical recognition of handwritten Chinese characters by hierarchical radical matching method," *Pattern Recognition*, vol. 34(1), pp. 15-35, 2001.

[2] F. L. Chung, W. W. S. Ip, "Complex character decomposition using deformable model," *IEEE Trans. Systems Man. Cybernet. Part C: Appl. Rev.*, vol. 31(1), pp. 126-132, 2001.

[3] D. Shi, G. S. Ng, R. I. Damper, S. R. Gunn, "Radical recognition of handwritten Chinese characters using GA-based kernel active shape modelling," *IEE Proceedings of Vision, Image & Signal Processing*, vol. 152(5), pp. 634-638, 2005.

[4] K. Chellapilla, P. Simard, "A new radical based approach to offline handwritten East-Asian character recognition," in: *Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition*, La Baule, France, 2006.

[5] E. Z. Ni, M. J. Jiang, C. L. Zhou, "Decomposition of reduced character of Chinese guqin notation," in: *Proceedings of 5th International Conference on Intelligent Systems and Knowledge Engineering*, Hangzhou, China, November 2010.

[6] X. H. Xiao, R. W. Dai, "On-line handwritten Chinese character recognition directed by components with dynamic templates," *Advances in Oriental Document Analysis and Recognition Techniques*, pp. 143-157, 1998.

[7] X. Q. Lv, D. S. Huang, E. M. Song, P. Li and C. S. Wu, "One radical-based on-line Chinese character recognition (OLCCR) system using support vector machine for recognition of radicals," in: *Proceedings of 1st International Conference on Bioinformatics and Biomedical Engineering*, Wuhan, China, pp. 558-561, 2007.

[8] L. L. Ma, C. L. Liu, "A new radical-based approach to online handwritten Chinese character recognition," in: *Proceedings of 19th International Conference on Pattern Recognition*, Tampa, Florida, pp. 1-4, 2008.

[9] E. Z. Ni, M. J. Jiang, X. J. Ding and C. L. Zhou, "Handwriting Input System of Chinese Guqin Notation," *ACM Journal on Computing and Cultural Heritage*, vol. 3(3), pp. 9-22, March 2011.

[10] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," in: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol.1, pp. 511-518, 2001.

[11] Y. Freund, R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sci.*, vol.55, pp.119-139, 1997.

[12] J. Friedman, T. Hastie, R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, pp. 337-407, 2000.

[13] J. X. Dong, A. Krzyżak, C. Y. Suen, "An improved handwritten Chinese character recognition system using support vector machine," *Pattern Recognition Letters*, vol. 26(12), pp. 1849-1856, 2005.

[14] T. F. Gao, C. L. Liu, "High accuracy handwritten Chinese character recognition using LDA-based compound distances," *Pattern Recognition*, vol. 41, pp. 3442-3451, 2008.

[15] Y. P. Wang, D. Q. Miao, D. X. Yue, "Off-line handwritten character recognition based on hierarchical classification," *Computer Science*, vol. 36(12), pp. 203-209, 2009.