2013 IEEE International Conference on Systems, Man, and Cybernetics

A faster method for Chinese font recognition based on Harris corner

Shuye Zhang, Lianwen Jin+, Dapeng Tao, Zhao Yang
School of Electronic and Information Engineering
South China University of Technology,China
+Lianwen.Jin@gmail.com

***Abstract*—A new fast font recognition method based on Harris corners is proposed in this paper. According to our observation, we find out that the font discriminative information is mainly hidden in some interesting points of characters. Based on this discovery, we extract the texture feature of the interesting points. The Harris corners are extracted as the interesting points. Experimental results show that our method is 20 times faster than the traditional Gabor features based method while keeping an almost equal accuracy.**

Keywords-Font recognition; Harris corner; local texture analysis; character independent

I. Introduction

OCR has been studied for many years. A lot of problems about character recognition have been solved while little work about font recognition is reported. Font recognition is necessary in many occasions. First, OCR system's performance will benefit from font information. Different fonts will make the character recognition difficult. If the font is known, different fonts will be converted into a specific font. Thus, it can make the character recognition easier. Second, font recognition is a part of document analysis. Title and main body are usually edited in different fonts, so that OCR system can distinguish the document layout on this bias. Third, it is necessary for document recovery. Sometimes, we not only need to get the document content, but also need to know the typeface.

In western character field, most methods are based on typographical features extracted by means of local attribute analysis. Cooperman [2] used some local detectors to get the font properties, such as serif, boldness, etc. Shi and Pavlidis [3] viewed word length and stroke slopes as font feature. Zramdini and Ingold's [4] method was also based on typographical features.

Western character is quite different from Chinese character. Chinese characters have no typographical attributes such as serif which is the intrinsic property of western characters. Researchers working on Chinese characters have put forward some different methods. In Chinese character field, Zhu and Tan [1] used a group of Gabor filters to extract the image's texture feature. The method requests the input image is a text block which is combined by a few characters. Ding's method [5] can work on a single Chinese character. Ding extracted the wavelet feature as font feature. Both of them viewed font recognition as the texture classification. However, both of them are time-consuming. It has difficulties in real-time application. Therefore, we carry out a new method which makes a good trade-off between accuracy and time. Our method consists of training part and testing part. First, we find Harris corners on the images. Then, we extract Gabor feature on these Harris corners. After that, we use the texture feature to train our classifier. At the testing part, we also detect the Harris corners on the input image and extract Gabor feature. Finally, SVM classifier is used to give a result for the input image. Fig.1 shows the flow chart of our method.

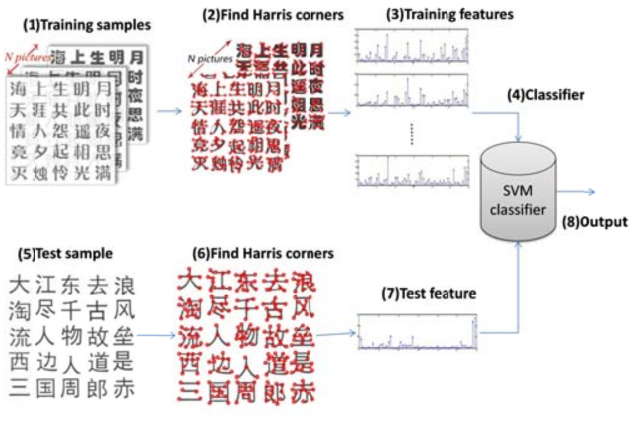

Figure 1. The flow chart of font identification system.

II. Font Feature

Feature extraction is a crucial process in the field of computer vision. Gabor feature has been proved as an efficient and robust feature in font identification. However, it's not fast enough. A practical system usually asks for fast and good feature. We propose a modified method which can get a satisfactory performance in recognition accuracy and running time. The following section will illustrate the feature extraction process in detail.

A. Find interesting points

In our paper, we view font recognition problem as a texture identification problem. We find out that different fonts mainly differ in some interesting points, such as the starting point, ending point and turning point. Font information has no relationship with character order. In other word, it's content-independent. Hence, we focus on the interesting points, instead of a whole image. We regard corners as our interesting points, because corner is the intersection of two edges and it represents a point in which the directions of these two edges change. According to our observation, corner and our interesting points (starting point, ending point, turning point and so on) have many same properties. Corner detection is frequently used in motion detection, image registration, video tracking and object recognition. Rosten [6] provided us with a faster and better method to detect corners. In this paper, we adopt Harris corner detector to find interesting points. Fig.2 shows the interesting points by using Harris corner detector. From the Fig.2, we can see that Harris corners mainly appear in starting points, turning points and ending points of a stroke or overlapping points of different strokes.

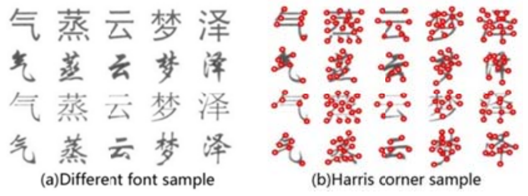

Figure 2. Interesting points found by Harris corner detector.

We analyze a lot of samples and find out that four class points are frequently in Chinese characters. Fig.3 illustrates four typical interesting points of a Chinese character.

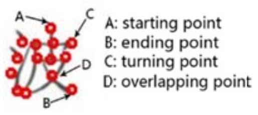

Figure 3. Corners of different positions.

B. *Local Gabor feature*

Gabor transformation is widely used in image process, especially the multichannel Gabor filtering technique ([8], [9]). In the following, we will briefly introduce multichannel Gabor filter technique. The Gabor transformation is:

$$F(x,y;f,\theta)=I(x,y)*g(x,y,f,\theta) \tag{1}$$

where $g(x,y,f,\theta)$ denotes Gabor filter. The Gabor filter is:

$$g(x,y,f,\theta)=\frac{1}{ab}\exp[-\pi(\frac{x_r^2}{a^2}+\frac{y_r^2}{b^2})][\exp(i2\pi f x_r)-\exp(-\frac{\pi^2}{2})] \tag{2}$$

to make calculation easier, Gabor filter is usually divided into even-symmetric part and odd-symmetric part. For a giving interesting point, the output of $g_o(x,y,f,\theta)$ and $g_e(x,y,f,\theta)$ are combined to provide a single channel output. Even-symmetric part $g_e(x,y,f,\theta)$ and odd-symmetric part $g_o(x,y,f,\theta)$ are:

$$g_o(x,y,f,\theta)=\frac{1}{ab}\exp[-\pi(\frac{x_r^2}{a^2}+\frac{y_r^2}{b^2})](\cos 2\pi f x_r-\exp(-\frac{\pi^2}{2})) \tag{3}$$

$$g_e(x,y,f,\theta)=\frac{1}{ab}\exp[-\pi(\frac{x_r^2}{a^2}+\frac{y_r^2}{b^2})](\sin 2\pi f x_r) \tag{4}$$

where f, θ, and σ are the spatial frequency, orientation, and size of the Gauss window, respectively. For a single Gabor filter, we need to set the following parameters in advance. The parameters are:

$$\theta_m=m\frac{2\pi}{M}, m=\{0,...,M-1\} \tag{5}$$

$$f_n=\lambda^{-n}f_{\max}, n=\{0,...,N-1\} \tag{6}$$

$$\lambda=\sqrt{2} \tag{7}$$

$$x_r=x\cos\theta_m+y\sin\theta_m, y_r=-x\sin\theta_m+y\cos\theta_m \tag{8}$$

$$a=b=\frac{1}{f_n} \tag{9}$$

For a given interesting point, convolution of the neighboring area and a single Gabor channel will give a numerical value of the interesting point.

In our paper, we select four orientations which are $0^\circ, 45^\circ, 90^\circ$ and 135°. In the following experiment, we choose 4 scales or 5 scales in order to explore the relationship between scale numbers and recognition rate. We set the maximum spatial frequency $f_{\max}$ as 0.25. In our experiment, first we need to calculate the Gabor mask. It's no need to make the Gabor window size infinitely large [6]. The size of images are 200×200 pixels, our experiments show that 15-pixels width for window size is enough. If we adopt multichannel Gabor filter which is combined by M orientations and N scales, we get a D-dimensional feature on an interesting point ($D=M\times N$). Supposed we find Q interesting points in an image, we can get a $Q\times D$ feature matrix. The mean values (M) and the standard deviations (S) of the feature matrix are chosen to represent the texture features.

There are mainly three advantages in our feature extraction method. It has no constraint on the image size or shape. Compared with the method proposed by [1], which shows the image size is strictly the same size and the image has the same number characters, it can run much faster because the interesting points is much less and the process finding the interesting points just takes a little time. Gabor feature based on interesting points can get a satisfactory recognition rate because it eliminates the impact of redundant pixels which make no contribution in font recognition.

III. CLASSIFIER

Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. For simplicity, we use LIBSVM [7] as our classifier.

LIBSVM provides us with linear kernel, polynomial kernel, RBF kernel and sigmoid kernel. We use linear kernel because linear kernel gets the highest accuracy in our experiments. And we set the penalty coefficient as 10. The rest of parameters keep the default values. We select 25 commonly used fonts. For each font, we generate 1997 image samples. In our experiment, we put the character image together to form a text block. We adopt 300 Tang poems, 283 Song poems and 1414 essay sentences as corpus. Then, we generate 1997 text blocks for one font class. For each font, 1000 training samples and 997 testing samples are used.

IV. Dataset

We edit 3866 commonly used Chinese characters in Microsoft Word software with 25 font style. For convenience, the document typesetting is predetermined. We finely tune the font size, character space and line space in order to make the subsequent segmentation easy. Second, we print them on A4 papers and use scanner to scan the papers and save then in JPG format. Then, we convert the JPG image into gray-scale image. After that, we align the gray-scale images by Hough transformation. Finally, we locate the position of text lines by dilation method. Thus, characters in the same line are connected while characters in different lines still keep separated. At the end, we utilize the gray-scale information to separate the characters. In this step, we compute the vertical projection profile (VPP). The peak corresponds to the split position of two characters. In this method, we save the Chinese character one by one in certain format. We obtain 25 font classes(Hei, YaHei, XiHei, YueHei, MeiHei, YaYuan, XingKai, Kai, FangSong, Song, ZhongSong, ZongYi, HuoYi, CuQian, GuangBiao, HuangCao, HuPo, LingXin, Shu, WeiBei, XinWei, YaoTi, YouYuan, LiShu, ShuangXian). A font class contains 3866 Chinese characters. Fig.4 shows some samples of Kai.

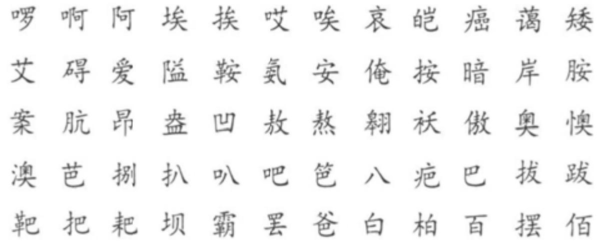

Figure 4. Kai samples in dataset.

We use the segmentation characters to assemble text block. Fig.5 presents 25 classes of different font sample. The text block is the same size 200×200 . The between-characters and between-lines space can be set according to our requirement.

V. Experimental Results

A. Recognition of Different Font

Experiments were carried out to examine the performance of our method. We have used Gabor filters of 16 channels or 20 channels. We also perform Zhu's method [1] in order to make a comparison. Zhu's method extracted the global Gabor feature of an image as the texture feature [1]. Our method extracts the Gabor feature of interesting points. Fig.6 shows four text blocks of different size. The character size in each text block is fixed while the text block size is determined by character number. Table I shows the experimental result. The experimental result is an overall recognition rate of 25 classes. In Table I, gabor_*m*_*n* means the method is combined by a group of Gabor filters which are *m* orientations and *n* scales, as well as corner_gabor_*m*_*n*. The experiments in this paper were all implemented in C++ using Microsoft visual studio 2010 on PC.

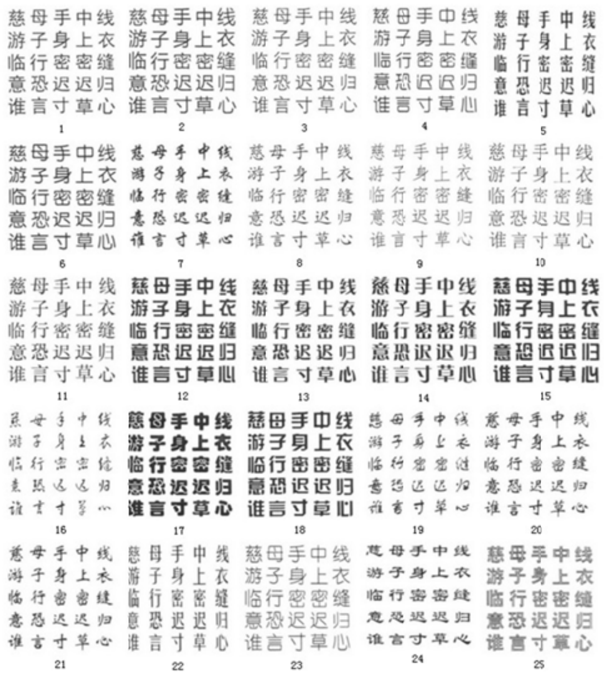

Figure 5. 25 classes of text blocks.

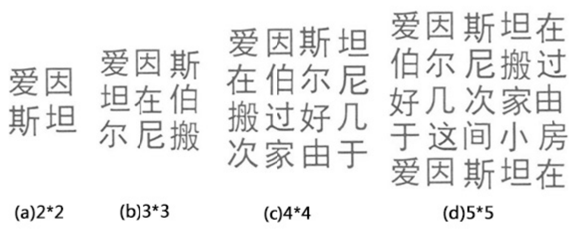

Figure 6. Text blocks of different size. A text block of N*N corresponds to an imge which has N lines and N characters in each line.

TABLE I. Recognition Results for our method and Zhu's in different size image

Methods	2×2	3×3	4×4	5×5
gabor_4_4	90.78	97.02	98.88	**99.23**
corner_gabor_4_4	90.06	97.37	99.14	**99.43**
gabor_4_5	94.40	98.36	99.44	**99.70**
corner_gabor_4_5	90.78	97.91	99.35	**99.57**

B. Robust Testing

Noise is inevitable in our real world, so it’s important to design an algorithm of a good robustness. We add pepper and salt noise to simulate contaminating effect in the real-world. We use noise density to measure different noise level. Supposed the original image from scanner is clean, we add different level noise to the clean image to generate images at different noise levels. Fig.7 presents a group of different level noise images.

$$NoiseLevel = \frac{\text{number of contaminated pixels}}{\text{number of total pixels}} \tag{10}$$

TABLE II. RECOGNITION ACCURACY ON DIFFERENT NOISE LEVEL

Methods	0	0.02	0.04	0.08
gabor_4_4	**99.24**	98.95	98.72	97.97
corner_gabor_4_4	**99.43**	97.79	96.01	93.32
gabor_4_5	**99.70**	99.45	99.26	98.78
corner_gabor_4_5	**99.57**	98.22	96.76	94.58

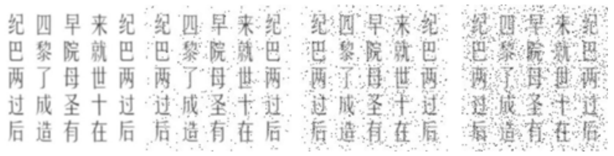

Figure 7. Text blocks at four noise level.

Our method can still achieve a recognition rate beyond 90 percent. It shows that our method works well though the images are severely contaminated.

C. Time testing

Speed is our method’s biggest advantage. As we know, Gabor feature extraction is time-consuming. We test our method and Zhu’s in a same way. Table III shows the duration time. Our method consists of two parts, which are the process of finding Harris corner and the process of extracting local Gabor feature. We respectively calculate duration time of them.

TABLE III. RESULT FOR FEATURE EXTRACTION’S TIME TEST

Methods	gabor_4_4	gabor_4_5	corner_gabor_4_4		corner_gabor_4_5	
			corner	Gabor	corner	Gabor
Average time for per-sample(ms)	451	571	16.6	7.1	16.6	8.9

Our experiments were implemented on a PC with Intel(R) Core(TM)2 Duo CPU E8400 3.00GHz processor and 2 GB RAM. In this experiment, we extract the features of 10,000 samples and get an average time for one image, which is showed in Table III. The image consists of 25 characters (5 rows and 5 columns). Its size is 200×200. The experimental result shows that our method is approximately 20 times faster than Zhu’s method.

D. Font confusion

Table IV shows the font confusion matrix for top ten fonts (the rest of 15 fonts is omitted for the sake of space). The table can tell us that the most similar fonts are YaHei and YaYuan.

For the sake of the paper layout, we choose the top 10 font classes. We select 1000 training samples and 997 testing samples. From the confusion matrix, we can see that 1.81% YaYuan samples is mistaken for YaHei and 2.11% YaHei samples is mistaken for YaYuan. Fig.8 shows the YaYuan and YaHei samples. It’s also difficult for human beings to distinguish the two fonts. YaHei and YaYuan samples are similar except the turning places.

TABLE IV. FONT CONFUSION MATRIX (PERCENTAGE)

Classes	Hei	YaHei	XiHei	YueHei	MeiHei	YaYuan	XingKai	Kai	FangSong	Song
Hei	99.1	0.9	0	0	0	0	0	0	0	0
YaHei	0.6	97.3	0	0	0	2.1	0	0	0	0
XiHei	0	0	1	0	0	0	0	0	0	0
YueHei	0	0.1	0	99.9	0	0	0	0	0	0
MeiHei	0	0	0	0	99.9	0	0.1	0	0	0
YaYuan	0	1.8	0	0	0	98.2	0	0	0	0
XingKai	0	0	0	0	0	0	1	0	0	0
Kai	0	0	0	0	0	0	0	1	0	0
FangSong	0	0	0	0	0	0	0	0	1	0
Song	0	0	0	0	0	0	0	0	0	1

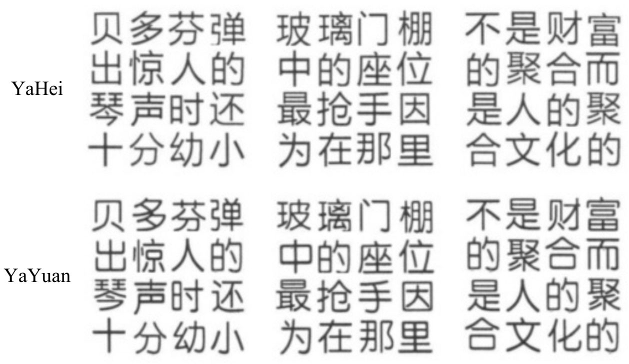
YaHei
YaYuan
贝多芬弹 玻璃门棚 不是财富
出惊人的 中的座位 的聚合而
琴声时还 最抢手因 是人的聚
十分幼小 为在那里 合文化的

Figure 8. Samples of YaHei and YaYuan.

VI. CONCLUSIONS

We have proposed a fast method for automatic font recognition. Our method is content-independent and shape-independent. The training and testing image samples are not required to be a same shape because we just extract the texture feature of the interesting points. At the meanwhile, the feature extraction process is very fast. It’s 20 times faster than Zhu’s method [1]. The average recognition accuracy of 25 typefaces over 20000 samples is almost the same as Zhu’s method. However, it just spends 23.7ms for a testing sample of 200 pixels by 200 pixels. Its high performance and fast speed makes the real-time font recognition become possible.

ACKNOWLEDGMENT

This research is supported in part by the National Natural Science Foundation of China (No.61075021), the National Science and Technology Support Plan (2013BAH65F01-

2013BAH65F04), GDNSF (No.S2011020000541), GDSTP (No.2012A010701001), SZSTP (No.CXY201106300148A), and the Doctoral Fund of MOE(No.20120172110023).

REFERENCES

[1] Y. Zhu, T.-N. Tan, and Y.-H. Wang, “Font Recognition Based on Global Texture Analysis,”*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, Oct. 2001

[2] R. Cooperman, “Producing Good Font Attribute Determination Using Error-Prone Information”, *Electronic Imaging'97. International Society for Optics and Photonics,* vol. 3027, pp.50-57,1997.

[3] H.Shi and T.Pavlidis, “Font Recognition and Contextual Processing for more Accurate Text Recognition,”*Proc. Fourth Int’l Conf. Document Analysis and Recognition,* pp. 39-44, Aug. 1997.

[4] A.Zramdini and R. Ingold, “Optical Font Recognition Using Typographical Features,”*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no.8, pp.877-882, Aug.1998.

[5] X.-Q. Ding, C. Li, W. Tao,“Character independent font recognition on a single Chinese Character,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2,pp. 195-204, 2007

[6] E. Rosten, R. Porter, and T. Drummond,“A Machine Learning Approach to Corner Detection,”*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105-119, 2010.

[7] C.-C. Chang and C.-J. Lin,“LIBSVM : a library for support vector machines,”*ACM Trans. Intelligent Systems and Technology*, 2:27:1—27:27, 2011.

[8] M.Turner, “Texture Discrimination by Gabor Functions,” *Biological Cybernetics*, vol. 55,pp. 71-82, 1986.

[9] A.Bovik, M. Clark, and W. Geisler, “Multichannel Texture Analysis Using Localized Spatial Filters,”*IEEE Trans. Pattern Anal. Mach. Intell.*, vol.12, no. 1, pp.55-73, Jan.1990.

[10] T.Tan, “Texture Feature Extraction via Cortical Channel Modeling,”*Proc.11th Int’l Conf. Pattern Recognition*, vol.III, pp. 607-610,1992.