2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics

Recognition of Offline Handwritten Chinese Characters Using the Tesseract Open Source OCR Engine

Qi Li, Weihua An, Anmi Zhou, Lehui Ma
College of Information Sciences
Beijing Language and Culture University
Beijing, China
liqihappyday@163.com, anweihua@blcu.edu.cn, 201311681051@stu.blcu.edu.cn, malehuichn@163.com

***Abstract*—Due to the complex structure and handwritten deformation, the offline handwritten Chinese characters recognition has been one of the most challenging problems. In this paper, an offline handwritten Chinese character recognition tool has been developed based on the Tesseract open source OCR engine. The tool mainly contributes on the following two points: First, a handwritten Chinese character features library is generated, which is independent of a specific user's writing style; Second, by preprocessing the input image and adjusting the Tesseract engine, multiple candidate recognition results are output based on weight ranking. The recognition accuracy rate of this tool is above 88% for both known user test set and unknown user test set. It has shown that the Tesseract engine is feasible for offline handwritten Chinese character recognition to a certain degree.**

Keywords-Offline handwritten Chinese characters; Optical Character Recognition; Tesseract

I. INTRODUCTION

Chinese characters recognition has made remarkable achievements in printed and on-line handwritten text over the past few decades. However, offline handwritten Chinese character recognition is still unable to satisfy the user's demands in practice [1]. The main reason is that the handwritten deformation make it hard to accurately conduct the characters segmentation, feature extraction and classification.

To address these problems, many methods have been tried, such as Support Vector Machine, Hidden Markov Model and Artificial Neural Network, etc. However, these technologies are still at the earliest stage of laboratory research.

This paper aims to explore the availability of the Tesseract engine in offline handwritten Chinese character recognition. Our main contribution includes two points: First, a feature library of handwritten Chinese characters, which is independent of some users' writing styles, is trained and generated. Second, a handwritten Chinese character recognition tool with high accuracy is developed by preprocessing the input image and adjusting the Tesseract engine.

II. THE TESSERACT OCR ENGINE

The Tesseract OCR engine was originally developed by a lab of Hewlett-Packard Company at Bristol from 1984 to 1994. As a commercial character recognition engine of HP flatbed scanner, it was among the best in the 1995 UNLV Accuracy Test [2]. Afterwards, HP decided to give up the OCR market, which stopped further development of Tesseract. Until 2005, HP decided to make the Tesseract as an open source software. Presently, the Tesseract has been released on Google Project, and is maintained by open source community.

Nowadays, the Tesseract supports a wide variety of languages, and provides default features libraries for these languages [3]. In addition, it also provides training methods. So users can train their own features library for specific character set.

Due to the expandability of the features library, the Tesseract engine has been applied by researchers to various scenarios. For instance, Hasnat [4] applied it to the recognition of Bengalese printed text. Torabi [5] applied it to the recognition of modern historical manuscript with printed English text, where many font variations exist, such as connection, black body and italic etc., and where some characters are even incomplete. Patel [6] used the engine to the recognition of vehicle license plates, and compared it with other commercial software. And also, it is applied to language category recognition and character direction estimation by Unnikrishanan [7]. For the recognition of Chinese printed text, Wan [8] and Cheng [9] developed two tools based on Tesseract, which are respectively used to recognize business card and ID information.

The Tesseract engine was originally designed for printed text recognition, but some researchers have been trying to apply it to handwritten text. For example, Rakshit [10, 11] applied it to the recognition of handwritten Roman numerals and handwritten English text notes.

In this paper, the availability of the Tesseract engine in offline handwritten Chinese character recognition has been explored. Different from the above mentioned works, the handwritten Chinese characters have the features such as complex structure and various deformations. Therefore, new strategies for dealing with them need be proposed.

III. OUR WORKS

As shown in Fig. 1, the flow chart of our recognition tool includes two parts: the training of the features library and the realization of the recognition tool. These two parts are discussed in detail in the following sections.

978-1-5090-0768-4/16 $31.00 © 2016 IEEE
DOI 10.1109/IHMSC.2016.239

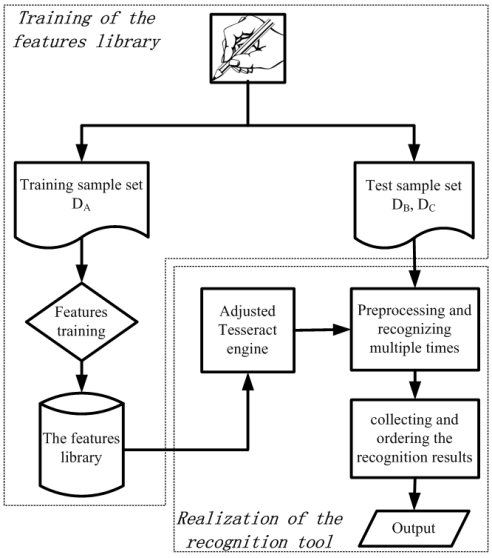

Figure 1. Flow chart of our recognition tool.

For the training of the features library, a character set is determined first; then training samples and test samples are collected for the character set; and finally these samples are trained for generating features library.

For the realization of the recognition tool, the Tesseract engine is adjusted to output multiple candidate results for one input image; then the input image is preprocessed multi-times, and the Tesseract engine is repeatedly called; and finally all the results are collected and orderly output according to their weight ranking.

IV. TRAINING OF THE FEATURES LIBRARY

A. The selection of character set

In this paper, we didn't select a large number of Chinese characters as the recognition set. On the contrary, we hope to analyze the recognition effects for a small number of characters, and to verify the feasibility of the Tesseract engine on the offline handwritten Chinese characters.

Therefore, 45 Chinese characters are selected as the character set for recognition, as shown in Fig. 2. This selection takes into account the mutual relationship between strokes, character structure and number of strokes.

B. Sample collection

The users for sample collection are made up of 20 people, including pupils, undergraduates and Chinese characters teachers. They are divided into two groups (Group I and Group II) with 10 people in each group. For the above selected character set, each user is required to write each character for 50 times.

After data collection, the samples written by Group I is divided equally into two sets and marked as D_A and D_B. Each set contains 25 samples of each character written by each user in Group I. The sample set written by Group II is marked directly as D_C.

贝 不 布 车 大 飞 孤 久 克 口 离 黎 力 猎 流
马 麦 茂 美 门 米 疲 人 日 上 生 手 水 丝 岁
天 土 王 文 希 小 醒 拥 又 于 远 云 正 掷 子

Figure 2. The character set for recognition.

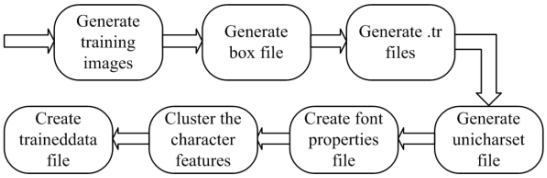

Figure 3. Training process of the features library.

Among the above three sets, D_A is used for the features library training. Because they are collected from 10 users, the obtained features library can reflect common handwriting features, and also achieve the user-independence.

Sets D_B and D_C are both used for recognition test. As the two sets come from different groups, we can analyze recognition effects under the conditions of known users (training set and test set come from the same group) and unknown users (training set and test set come from different groups).

Note that the handwriting requirement is assumed as constrained handwriting in this paper, which requires users to write as much standardly as possible. Scrawling or poor handwriting is not allowed. Additionally, the focus of this paper is single handwritten character recognition, and so we do not consider the segmentation among multiple handwritten characters.

C. Training process

For the sample set D_A, Fig. 3 shows the training process for the features library and it is described below.

1) Generate training images: Firstly, the handwritten samples are scanned to obtain the original images. They are then preprocessed by noise eliminating and binarization. Finally, all the binary images are merged into a .tif format file. The naming standard of .tif file is as follows: [lang].[fontname].exp[num].tif. In this paper, string "chi.written.exp0.tif" is used.

2) Generate box file: The Tesseract engine annotates the handwritten samples in the training images by using .box file, which is a text file with the labeling format shown in Fig. 4. In this figure, each line denotes annotated information of one handwritten sample. The contents are: the first column is the corresponding character of the sample; the second and third columns are the coordinates of the bounding box; the fourth and fifth columns are the width and height; and the last column is the number of the image file.

To create a .box file, the following command in the Tesseract engine can be executed. In this command, the first parameter is the file name of the training image, and the second parameter is the prefix name of the box file to be created. The third parameter "-l chi_sim" means using the

```
贝 1002 727 1033 795 0
贝 1124 730 1159 796 0
贝 1242 732 1281 798 0
贝 1364 732 1392 798 0
贝 1485 731 1517 797 0
贝 1602 731 1633 793 0
贝 1724 731 1756 801 0
```

Figure 4. An example of the box file.

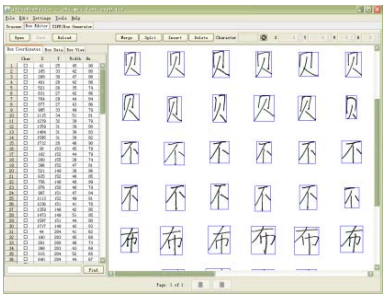

Figure 5. Adjusted results with jTessBoxEditor tool.

default Chinese Song typeface features library to recognize the samples.

tesseract fontfile.tif fontfile -l chi_sim batch.nochop makebox

The generated .box file can be opened with jTessBoxEditor tool, and the inaccurate annotation results can be corrected. The adjusted results are illustrated in Fig. 5. Each blue bounding box indicates each segmented sample. The column on the left indicates the annotation results of the samples.

3) Generate .tr files: The Tesseract engine stores the shape features of handwritten samples by using .tr files. For a .tif image and its related .box file, the following command is used to generate the .tr file. The second parameter in the command is the prefix name of the .tr file and it is generally the same as the name of the .tif file.

tesseract fontfile.tif fontfile nobatch box.train

4) Generate unicharset file: The unicharset file is used to save the complete character set for recognition. The command to generate the unicharset file is given below. It can combine multiple box files and generate one unicharset file.

unicharset_extractor fontfile_1.box fontfile_2.box …

5) Create font properties file: The name of the font properties file is font_properties. This file is used to store the font information of the training image. It only contains one text line with the following format.

<fontname> <italic> <bold> <fixed> <serif> <fraktur>

The first item is same as the [fontname] in the .tif filename, and the values of the following five items are 1 or 0, which indicates whether the training image have the corresponding font effects. In this paper, the content of this file is "written 0 0 0 0 0".

6) Cluster the character features: Based on the results of the previous five steps, the character features need to be clustered to generate four auxiliary files: shapetable, inttemp, pffmtable and normproto. As shown below, the first command is used to generate the shapetable file, the second command is used to generate the inttemp file and pffmtable file, and the third command to generate normproto file.

shapeclustering -F font_properties -U unicharset fontfile_1.tr fontfile_2.tr …

mftraining -F font_properties -U unicharset fontfile_1.tr fontfile_2.tr …

cntraining fontfile_1.tr fontfile_2.tr …

7) Create the traineddata file: A unified prefix is set for the generated five files (shapetable, inttemp, pffmtable, normproto and unicharset), such as "written". Then the following command is used to generate the final features library.

combine_tessdata written.

The name of the generated features library is "written.traineddata". After placing it into the installation directory of the Tesseract engine, the handwritten Chinese characters can be recognized.

V. REALIZATION OF THE RECOGNITION TOOL

This paper aims to recognize single handwritten character. For a single handwritten Chinese character image, the recognition command provided by the Tesseract engine is shown below, in which "character.jpg" is the image filename and "result" is the name of the output text file.

tesseract character.jpg result -l written -psm 10

By default, the above command can only output one character as the recognition result. For a handwritten sample, a single result cannot guarantee high recognition accuracy. Moreover, the feedback form of single result is not convenient for subsequent editing. Therefore, an improved recognition tool is realized based on the Tesseract engine. As shown in Fig. 6, the improvements include the following two points.

A. The adjustment of Tesseract engine

Fig. 6 shows the process flow of the Tesseract engine [12]. The first step is page layout analysis for detecting the text area in the input image. The second step is blob finding, which divides the detected text area into a series of blobs. A blob is a putative classifiable unit, which may contain several characters or may be a part of some character. The third step is to determine the text lines, and to incorporate the blobs into a series of words according to the blank spacing.

The above mentioned three steps are prepared for the word recognition. The fourth step is to recognize each word including two passes. On each pass, the Tesseract engine tries to conduct various splitting and merging operations to the blobs in one word and forms a series of character outlines to be recognized. On pass 1, the engine recognizes those character outlines with the static classifier based on the features library. If the recognition result has high confidence, it will be passed to the adaptive classifier as training data. On pass 2, the engine once again recognizes those words with low confidence on pass 1, thus to improve the recognition accuracy.

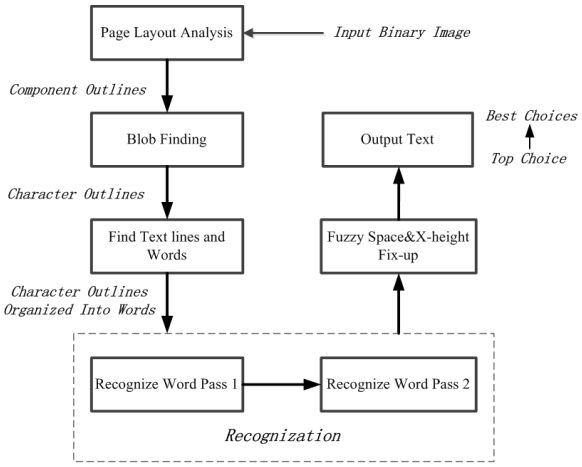

Figure 6. Process flow of the Tesseract engine.

The function of the fifth step is to adjust the recognition results for the words with low recognition accuracy according to the fuzzy-space and character height. For each word to be recognized, the fifth step generates multiple candidate recognition results. This step stores the results and their confidence values in a table. By default, the sixth step chooses only one result with the highest confidence as the output. In this paper, this step is modified to output all the candidate results.

B. Preprocessing the input image

During the features library training process, the handwritten samples come from ten people. Even so, it cannot represent all the possible handwriting deformation. Accordingly, during the character recognition, the input image should reflect the existing features in the features library as much as possible.

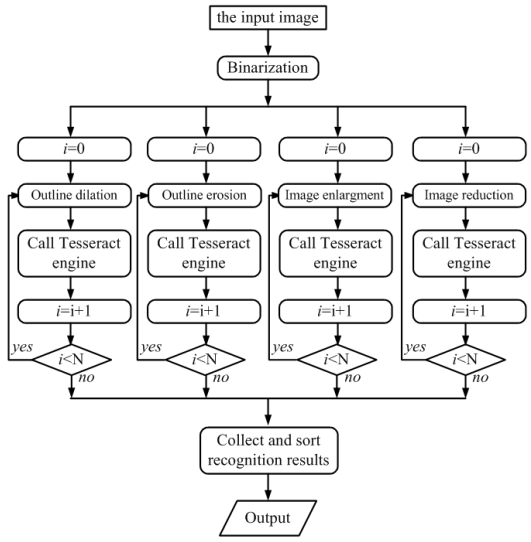

Figure 7. Process flow of the input image.

Therefore, an image preprocessing strategy is proposed as shown in Fig. 7. First of all, the input image is processed with binarization. Then, the following four operations are applied on the binary image: outline dilation and erosion, image enlargement and reduction. Each operation is repeated for N times (where N is 5 in this paper). After each execution, the Tesseract engine is called to obtain multiple candidate results.

Outline dilation and erosion can change the connected areas in the binary image, thus to change the overlapping relations between the strokes. Image enlargement and reduction can change the strokes' thickness and the character size. Repeating these operations can make the character shape in the input image to match with the training features of the target character as much as possible.

As shown in Fig. 7, our tool collects all the candidate recognition results and calculates their weights using the following equation.

$$w_a = \sum_{i=1}^{m_a} w_a^i \qquad (1)$$

where w_a denotes the final weight of the candidate recognition result a, m_a denotes the appearing times of a, and w^i_a denotes the corresponding confidence when the Tesseract engine output a at the i_{th} recognition.

Finally, we sort all the candidate results by weights. Three results with the largest weights are output as the final outputs in this paper.

VI. EXPERIMENTAL RESULTS

The test set D_B and training set D_A are different sample sets from the same people. Therefore, the recognition test of D_B may reflect the effects under the known user condition. Fig. 8(a) represents the recognition accuracy rate of D_B. The lower polyline is calculated under the condition that the features library in Section IV and the original Tesseract engine are used. The top polyline is calculated under the condition that both the features library in Section IV and the improvement strategy in Section V are used.

As seen from Fig. 8(a), without the improvement strategy in Section V, the accuracy rate are irregularly distributed, with the highest 100% and the lowest 4%. The average accuracy rate of all samples is 64%, which is not ideal. The reason is that all the deformation of handwritten characters cannot be completely covered during the features training. After adopting the strategy in Section V, the recognition accuracy rate of each Chinese character reaches over 92%.

The test sets D_C and D_A come from different people. Therefore, the recognition test of D_C may reflect the effects under the unknown user condition. Fig. 8(b) presents the statistical results of D_C. As seen from the figure, the recognition accuracy rate is jumbled without adopting the improvement strategy. In the case of adopting the improvement strategy, the recognition accuracy rates are all above 88%, which is consistent with the results of D_B.

Note that our character set is small. So, our tool has not been tested on some well-known offline handwritten Chinese character databases [13, 14]. But from the above results, it

can be seen that the offline handwritten Chinese character recognition based on the Tesseract engine is feasible. In the future, we will modify our tool and test it on a large scale of Chinese character sets.

VII. CONCLUSION

In this paper, the Tesseract engine is improved for the application of offline handwritten Chinese character recognition. Firstly, a features library independent of specific users is trained and generated. Secondly, the preprocessing strategy for the input images is proposed. Finally, the Tesseract engine is adjusted to output multiple recognition results. The experiment shows that the offline handwritten Chinese character recognition based on the Tesseract engine is feasible at a certain degree.

Our further work contains the training of features library for a large scale of character set, realizing the recognition of multiple text lines and exploring the selection of the candidate recognition results according to the text context.

ACKNOWLEDGMENT

We thanks the support of National Natural Science Foundation of China (Grant No.61202249), WuTong innovation platform of Beijing Language and Culture University (No.16PT04) and the youth academic backbone support program of Beijing Language and Culture University. This paper is also supported by the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (No.16YCX168).

REFERENCES

[1] Ji-yin Zhao, Rui-rui Zheng, Bao-chun Wu, Min Li. A Review of Offline Handwritten Chinese Character Recognition. Acta Electronica Sinica, 2010, 38(2): 405-415

[2] S.V. Rice, F.R. Jenkins, T.A. Nartker. The Fourth Annual Test of OCR Accuracy, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995

[3] R. Smith, D. Antonova, D.-S. Lee. Adapting the Tesseract Open Source OCR Engine for Multilingual OCR. In Proceedings of the International Workshop on Multilingual OCR, MOCR '09, pages 1:1–1:8, New York, NY, USA, 2009.

[4] M. A. Hasnat, M. R. Chowdhury, and M. Khan. An Open Source Tesseract Based Optical Character Recognizer for Bangla Script. In Proceedings of the International Conference on Document Analysis and Recognition, pages 671–675, 2009.

[5] K. Torabi, J. Durgan, B. Tarpley. Early Modern OCR Project(eMOP) at Texas A&M University: Using Aletheia to Train Tesseract. In Proceedings of the 2013 ACM symposium on Document engineering, Pages 23-26, New York, USA, 2013

[6] C Patel, A Patel, D Patel. Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. International Journal of Computer Applications (0975 – 8887), Pages 50-56, Volume 55–No.10, October 2012Electronic Publication: Digital Object Identifiers (DOIs):

[7] R. Unnikrishnan, R. Smith. Combined Script and Page Orientation Estimation using the Tesseract OCR engine. In Proceedings of the International Workshop on Multilingual OCR, MOCR '09, New York, NY, USA, 2009

[8] S. Wan. Research and implementation of biz card recognition system based on Tesseract-OCR engine. Master Dissertation, South China University of Technology, Guangzhou, China, 2014

[9] Y.H. Cheng. The Design and Development Of Identification Recognizing System Based On Tesseract. Master Dissertation, Donghua University, Shanghai, China, 2014

[10] S. Rakshit, A. Kundu, M. Maity, S. Mandal, S. Sarkar, S. Basu. Recognition of handwritten Roman Numerals using Tesseract open source OCR engine. In Proceedings of the International Conference on Advances in Computer Vision and Information Technology, Pages 572-577, 2009

[11] S. Rakshit, S. Basu, H. Ikeda. Recognition of Handwritten Textual Annotations using Tesseract Open Source OCR Engine for information Just In Time (iJIT). In Proceedings of the International Conference on Information Technology and Business Intelligence (2009) 117-125

[12] R. Smith. An Overview of the Tesseract OCR Engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition, 2: 629-633, 2007

[13] J. Guo, Z.Q. Lin, H.G. Zhang. A new database model of offline handwritten Chinese character and its application. Acta Eletronica Sinica, 2000, 28(5):115 -116

[14] T. Su, T. Zhang, D. Guan. Corpus-based HITMW database for offline recognition of general-purpose Chinese handwritten text. International Journal on Document Analysis and Recognition, 2007, 10(1):27-38

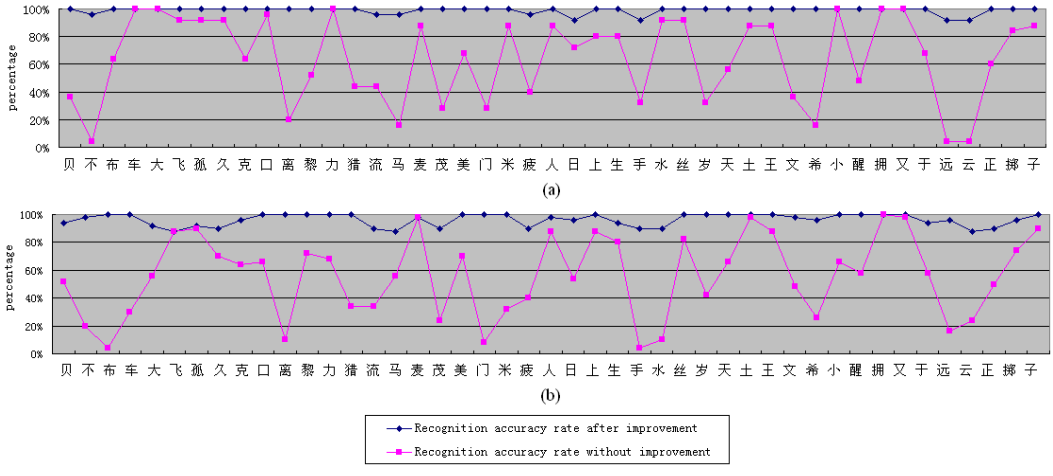

Figure 8. Recognition results of the test set D_B and D_C.