A Convolution-Based Approach for Fixed-Pattern Noise Removal in OCR

Jiawei Mo, Baohua Wang, Zezhou Zhang, Zhengze Chen, Zeyin Huang, Jiale Zhang, Xuanhui Ni
College of Mathematics and Statistics
Shenzhen University
Shenzhen, China, 518060
e-mail: 245994626@qq.com, bhwang@szu.edu.cn, aaronzzz@126.com, 809999640@qq.com, 13427978254@163.com, 506894980@qq.com, 1378959973@qq.com

***Abstract*—There is some fixed-pattern noise in the OCR text image and the denosing is needed to improve the accuracy of recognition. In this paper, a convolution-based approach for the fixed-pattern noise removal in OCR is proposed. The approach identifies the location of text content pixels and removes noise pixels based on the convolution kernel. The experiment shows that the approach is an effective way to remove the underline and improve the accuracy of recognition. For its generality, the algorithm is also applicable of removing other type of fixed-pattern noise.**

Keywords-fixed-pattern noise; denoising; convolution

I. INTRODUCTION

OCR is short for Optional Character Recognition, which translates the text image information of books, newspapers, manuscripts, forms, and other printed materials obtained by scanners, cameras, and other optical input methods into text information that can be recognized and processed by computers. [1] A denoising approach, which removes unrelated pixels surrounding the text pixels, is always needed in the preprocessing step of OCR to improve the accuracy of recognition.

There is some fixed-pattern of noise in the OCR text image. For example, named entities are usually underlined in Chinese ancient documents and it is in great need of making vast amount of Chinese ancient documents digital by OCR. These underlines are the fixed noise and experiments have shown that most of the underlined named entities are wrongly recognized by OCR without denoising. An efficient algorithm is needed to remove the underline from the text image and then to improve the accuracy.

There are several denoising algorithms for text image, like mean filtering [2], median filtering [3], [4], [5], wavelet denoising [6], morphological processing [7], [8], least-squares method [9], high order cumulant method [10], marginal noise removal [11], characters stroke characteristic-based method [12] and diffusion method [13]. However, most of these algorithms aim to remove non-fixed-pattern of noise, such as impulse noise. They are not suitable for the text images which have good scan quality, but have fixed pattern of noise.

A convolution-based fixed noise removal approach is proposed in this paper. The convolution is a powerful tool in feature recognition tasks and it can convert the arrangement of pixels in a unit area into a specific value, thus transforming the complicated text feature judgments into simple numerical judgments. In our method, we use convolution to identify the right position of the text and remove the fixed noise and other noises surrounding which affect the accuracy of recognition. The algorithm proposed can not only be used to remove underline, but it is also applicable of removing other type of fixed-pattern noise.

This paper is organized as follows. Section 2 gives detail of convolution-based fixed noise removal approach. Section 3 presents experiments conducted on real document images. Concluding remarks and future works are discussed in Section 4.

II. CONVOLUTION-BASED FIXED NOISE REMOVAL APPROACH

As mentioned above, the convolution is a powerful tool in feature recognition tasks and we propose a convolution-based approach for fixed noise removal problem. The approach has following three steps:

Firstly, identify the center point of each character and mark it by using convolution operations and numerical judgments. In order to cope with the situation of "repeated mark of same-character center point" and "misjudgment of the non-character area as the character" which may appear in the recognition process, this step also includes "reduplicate center points removal" and "fake center point removal" operations.

Secondly, generate a reservation matrix. Initialize an all-zero matrix with the same specifications as the original image and then use the center point marked in the first step as a standard, set the value in each rectangle centered on the point to be 1. After the "Set 1" operation is completed, all pixels with a value of 1 in the reservation matrix represent that the corresponding position of the original image will be kept, and all pixels with a value of 0 represent that the corresponding position of the original image will be removed.

At last, take the reserve operation. Make a dot product of the pre-processed original image matrix and the reservation matrix generated in step 2, the resulting matrix image will be an image in which noises surrounding are all removed.

A. Convolution

In this paper, we stipulate all the sizes of the convolution kernel to be odd and all the elements index's initial value to be 0.

978-1-5386-6987-7/18/$31.00 ©2018 IEEE

Suppose we have the matrix $A \in \Re^{n\times n}$ and the convolution kernel $K \in \Re^{n\times n}$. The i th row and j th column element of A is denoted $a_{i,j}$ and The i th row and j th column element of K is denoted $k_{i,j}$. The convolution result of A and K is the variable y where:

$$y = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_{i,j} \cdot k_{i,j}$$

Suppose we have the matrix $B \in \Re^{m\times n}$ and the convolution kernel $K \in \Re^{d\times d}$. The i th row and j th column element of B is denoted $b_{i,j}$ and the i th row and j th column element of K is denoted $k_{i,j}$. The convolution result of B and K is the matrix $R \in \Re^{m\times n}$. The i th row and j th column element of R is denoted $r_{i,j}$ where:

$$r_{i,j} = \sum_{s=0}^{d-1}\sum_{t=0}^{d-1} k_{s,t} \cdot b'_{s+i-\frac{d-1}{2},t+j-\frac{d-1}{2}}$$

and

$$b'_{p,q} = \begin{cases} b_{p,q} & (p \geq 0 \,\&\, q \geq 0) \\ 0 & (p < 0 \,|\, q < 0) \end{cases}$$

B. Identify the Center Point of the character

Suppose we have an original text image matrix whose black pixel value is 1 and the white pixel value is 0 and the font size is s_0 *px*. According to the definition of the convolution, when the kernel is aligned with an image area whose center point is in the i th row and j th column, the value of the i th row and j th column element in the resulting matrix is going to be the sum of the dot product of the kernel and the image area. Let's just call this value the "single convolution result". Obviously, when the kernel is aligned with a whole white area, the "single convolution result" is 0.We need to construct a special kernel $K \in \Re^{(s_0+2)\times(s_0+2)}$ so that it knows whether it center a character.

To verify whether the kernel center the character, we set the values of the kernel's edge to be $\left(-1 \cdot s_0^2\right)$ and call it the "punishment edge". As shown in Figure 1, suppose the kernel values within the edge are all 1. If the kernel doesn't center the character, there must be many black pixels of the character that "touch" the "punishment edge". By the definition of the convolution, the "single convolution result" is absolutely non-positive. Therefore, only when the kernel centers the character, the "single convolution result" is positive.

We have assumed they are all 1. When the kernel center the character, the "single convolution result" is absolutely positive. However, there is an exception that the "single convolution result" is positive but the kernel actually doesn't align with a character.

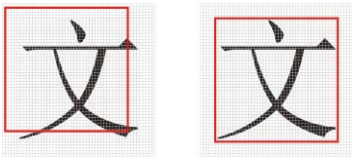

Figure 1. The sample of the "punishment edge".

To solve this problem, we set the top quarter of the kernel values within the edge to 0. We call this area the "ignored area". There is a basic fact that all of the Chinese characters have black pixels out of the "ignored area" if the kernel center them, thus the "single convolution results" are going to be positive, but for the exception case, the pixels of the single underline are all contained in the "ignored area", so the "single convolution result" is 0.

With the use of "punishment edge" and "ignored area", the kernel finally get the ability to verify whether it center the character. Obviously, all positive elements in the convolution result of the text image matrix and the kernel are the center point of the characters.

C. Construct the Center Point Identification Kernel

We manually measure the font size S_0 and the max interval between the character and the underline m_0. Their units are *px*. We stipulate S_0 to be odd.

Suppose we have a center point identification kernel $K_1 \in \Re^{(s_0+2)\times(s_0+2)}$, the i th row and j th column element of K_1 is denoted $k_{1_{i,j}}$, where:

$$k_{1_{i,j}} = \begin{cases} 1 & \left(i \in \left[\left\lfloor \frac{s_0+2}{4} \right\rfloor, s_0\right] \,\&\, j \in [1, s_0]\right) \\ -1 \cdot s_0^2 & \left[(i = 0 \,|\, i = s_0+1) \,\&\, (j = 0 \,|\, j = s_0+1)\right] \\ 0 & \left(i \in \left[1, \left\lfloor \frac{s_0+2}{4} \right\rfloor\right] \,\&\, j \in [1, s_0]\right) \end{cases}$$

D. Mark the Center Point by Convolution

In the gray image matrix, the black pixel's value is 0 while the white pixel's value is 255. We need to convert them to 1 and 0. Suppose the original pixel's value is denoted a_1 and the converted value is denoted a_2, then we have the following convert formula:

$$a_2 = \mathrm{mod}(a_0 + 1, 256)$$

We call the new matrix "zero-one matrix".

Suppose we have the "zero-one matrix" $B \in \Re^{m \times n}$ and the convolution kernel K_1. The convolution result of B and K_1 is the matrix $R_1 \in \Re^{m \times n}$. The i th row and j th column element of R_1 is denoted $r_{1_{i,j}}$. We make a new matrix $R_2 \in \Re^{m \times n}$. The i th row and j th column element of R_2 is denoted $r_{2_{i,j}}$ where:

$$r_{2_{i,j}} = \begin{cases} 1 & \left(r_{1_{i,j}} > 0\right) \\ 0 & \left(r_{1_{i,j}} \le 0\right) \end{cases}$$

Here we say all of the pixels in R_2 whose value are 1 are the center points of the characters in the text image.

E. Reduplicate Center Points Removal

As the font size we measured could have some error, during part A, there might be more than one pixel recognized as the center point of the same character. Besides, the set of their position is a rectangle.

We just want to keep one center point for each character and we stipulate that only keep the lower right corner one or so-called "the last one". Obviously, if a character has only one center point being recognized, the center point is also the "last one". We need to construct another special kernel which can verify whether the center point is the "last one". We call it the last point identification kernel. Like the center point identification kernel, when the last point identification kernel center is the "last one", the "single convolution result" is positive, otherwise, it is non-positive.

Suppose we have a last point identification kernel K_2, and it looks like:

$$K_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

The convolution result of R_2 and K_2 is the matrix $R_3 \in \Re^{m \times n}$. The i th row and j th column element of R_3 is denoted $r_{3_{i,j}}$. We make a new matrix $R_4 \in \Re^{m \times n}$. The i th row and j th column element of R_4 is denoted $r_{4_{i,j}}$ where:

$$r_{4_{i,j}} = \begin{cases} 1 & \left(r_{3_{i,j}} > 0\right) \\ 0 & \left(r_{3_{i,j}} \le 0\right) \end{cases}$$

Here we say all of the pixels in R_4 whose values are 1 are the "last" center points of the characters in the text image.

F. Fake Center Point Removal

Now we have found the center points of all the characters. Most of the characters have only one center point after the reduplicate center points removal. However, for those characters who have a single underline, they might have two center points.

It is easy to see, the upper point is the real one, so we need to construct another special kernel which can verify it. We call it the real center point identification kernel. Like the center point identification kernel, when the real center point identification kernel center the real one, the "single convolution result" is positive, otherwise, non-positive.

Before we construct the kernel, there is another fact we have to know: the distance between the real center point and the fake one is lower than the max interval between the character and the underline.

Now we construct a real center point identification kernel K_3, with a shape of $(2m_0+1)\times(2m_0+1)$. The i th row and j th column element of K_3 is denoted $k_{3_{i,j}}$, where:

$$k_{3_{i,j}} = \begin{cases} 1 & \left(i = j = m_0\right) \\ 0 & \left[\left(i = m_0 \;\&\; j \ne m_0\right) \mid \left(i \in \left[m_0+1, 2m_0\right]\right)\right] \\ -1 & \left(i \in \left[0, m_0-1\right]\right) \end{cases}$$

The convolution result of R_4 and K_3 is the matrix $R_5 \in \Re^{m \times n}$. The i th row and j th column element of R_5 is denoted $r_{5_{i,j}}$. We make a new matrix $R_6 \in \Re^{m \times n}$. The i th row and j th column element of R_6 is denoted $r_{6_{i,j}}$ where:

$$r_{6_{i,j}} = \begin{cases} 1 & \left(r_{5_{i,j}} > 0\right) \\ 0 & \left(r_{5_{i,j}} \le 0\right) \end{cases}$$

Here we say all of the pixels in R_6 whose values are 1 are the real center points of the characters in the text image.

G. The Reserve Operation

After the pre-processing, the values of the black pixels' in image matrix are 1 while the whites are 0. Now we want the values of those underline pixels to be 0 while the values of those character pixels keep the same.

To do this, we can construct a zero matrix with a same size of the image matrix. Then, find all the character pixels and let the value of the corresponding pixels in the zero matrix be 1. Finally, make a dot product of these two matrices. In the resulting matrix of the dot product, only the values of those character pixels are same as the corresponding values in the image matrix and the values of those non-character pixels are 0, no matter the corresponding values in the image matrix are 0 or 1. Obviously, this resulting matrix is so-called "underline removed" image that we want. We call that zero matrix the reservation matrix.

However, we can't do this immediately because we haven't marked the black pixels of all the characters precisely, so we can mark a square area with an edge length of the font size that contains a character. That's why we need to find the center points of all characters in the above operations and only mark the square areas that center the center points. Then we can successfully construct the reservation matrix where the values of the pixels in the square areas are 1.

To mark the square areas for the characters, we need to construct an all-one kernel. Let's just call it the reservation kernel. Suppose we have to mark a square area which centers a character. Obviously, when the kernel centers one of the pixels in the square area, the "single convolution result" is positive, otherwise, non-positive. In the end, we change the values of the positive pixels to 1, and then we finish constructing the reservation matrix.

The reservation kernel is denoted K_4 with an edge length of s_0. The convolution result of R_6 and K_4 is the matrix $R_7 \in \Re^{m\times n}$. The i th row and j th column element of R_7 is denoted $r_{7_{i,j}}$. We make a new matrix $R_8 \in \Re^{m\times n}$. The i th row and j th column element of R_8 is denoted $r_{8_{i,j}}$ where:

$$r_{8_{i,j}} = \begin{cases} 1 & \left(r_{7_{i,j}} > 0\right) \\ 0 & \left(r_{7_{i,j}} \le 0\right) \end{cases}$$

Here we say the matrix R_8 is the reservation matrix. Then make a dot product of the reservation matrix and the "zero-one matrix" B. The sample of a dot product is shown in the following equation.

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \otimes \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} = \begin{vmatrix} 1 & 6 \\ 15 & 28 \end{vmatrix}$$

Obviously, the resulting matrix of the dot product is the "underline removed" image matrix. Here we need to change the resulting matrix to the gray image matrix. Suppose the values of the resulting matrix are denoted b_1 and the values of the new gray image are denoted b_2, where:

$$b_2 = \mathrm{mod}(b_1 + 255, 256)$$

Finally, we get the "underline removed" gray image.

III. Experimental Framework

In this part, we report the experimental framework we take and show whether the convolution-based removal approach can increase the accuracy of OCR task.

A. OCR Accuracy

The accuracy is defined as follows. The total number of the characters is denoted as *T*, and the total number of the wrong recognized characters is denoted *E*.

$$accuracy = \frac{T - E}{T}$$

B. BLEU

BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another [14]. In this paper, we use it to examine the text similarity between the actual text and the text recognized from "underline removed" image.

The set of *n* characters is denoted *n-gram*. The actual text is denoted *s* and the text recognized from the "underline removed" image is denoted *c*. The *k* th *n-gram* in the s is denoted w_k. The total number of the w_k in s is denoted $h_k(s)$ and the total number of the w_k in the c is denoted $h_k(c)$. The total number of the characters in s is denoted l_s and the total number of the characters in c is denoted l_c. The BLEU score is denoted $BLEU$. Then we define the $BLEU$ with the following formula:

$$P_n = \frac{\sum_k \min\left(h_k(c), h_k(s)\right)}{\sum_k h_k(c)}$$

$$BP = \begin{cases} 1 & l_c > l_s \\ e^{1-\frac{l_s}{l_c}} & l_c \le l_s \end{cases}$$

$$BLEU = BP \times e^{\sum_{n=1}^{4} \frac{\log P_n}{n}}$$

C. Data set

For OCR accuracy experiments, the data set is taken from the Complete Translation of the Twenty-Four Histories series and randomly two pages are selected as test set from each ten book of this series. For the BLEU experiment, we chose the first 160 pages of volume three of the Book of Later Han as the test image. The total number of the characters in these pages was 73,660.

D. Parameter Setting

The s_0 was set to 43 while the m_0 was set to 5.

E. Baseline Algorithm

To compare the performance, we implement a traditional non-convolution-based method which scans the image vertically and finds the row has the underlines. Once it finds that one row has underline pixels, it changes the whole pixels of this row into white. We call it the cut method.

F. Results

1) OCR accuracy test

TABLE I. The Result of the Accuracy Test. T Stands for the Total Number of the Characters. M Stands for the Removal Approach Type. E Stands for the Total Number of the Wrong Recognized Characters. A Stands for the Accuracy

T	M	E	A
8383	None	599	0.9285
	Cut	83	0.9900
	Conv	64	**0.9923**

2) BLEU test

TABLE II. THE RESULT OF THE BLEU TEST

Method	BLEU
None	0.8449
Cut	0.9071
Conv	**0.9250**

G. Analysis

Table I, Table II show that the convolution-based removal approach can increase the OCR accuracy significantly. Therefore, we came to a conclusion that the convolution-based fixed-pattern noise removal approach is an effective way to remove the underline and increase the accuracy of OCR.

IV. CONCLUSION

In this paper, a convolution-based fixed-pattern noise removal approach in OCR is proposed. The approach identifies the location of text content pixels and preserves them so as to remove noise pixels in disguised form. The result of the experiment shows that the approach is an effective way to remove the underline and increase the accuracy of OCR. The algorithm proposed can not only be used to remove underline, but it is also applicable of removing other type of fixed-pattern noise.

However, as we manually select some parameters, the algorithm may have some mistakes like preserving the noise pixels or missing the text content.

As future work, we will try to develop a new algorithm that can automatically get those parameters to reduce man-made errors.

ACKNOWLEDGMENT

This work is supported by *College Students' Innovation and Entrepreneurship Training Program of Shenzhen University.*

REFERENCES

[1] Sun Yufei.Studies of OCR Technology for Degraded Document Images.Graduate University of Chinese Academy of Sciences.2005

[2] Tang,Cai- hong Cai,Lidong.A Weighted Mean Filtering Algorithm Based on Histogram[J],2006-13

[3] LIU Guo-hong, GUO Wen-ming.Application of improved arithmetic of median filtering denoising.Computer Engineering and Applications, 2010, 46 (10):187-189

[4] LI Yan-jun, SU Hong-qi, YANG Feng, FAN Guo-liang, LIN Peng. Improved algorithm study about removing image noise.Computer Engineering & Design , 2009

[5] XG Yang, F Meng, LI Jun-Shan. Practical image filtering algorithm. Journal of Computer Applications , 2009

[6] WANG Zhong-feng(92493 Army 22 Unit,Huludao 125000,China).Study on Document Image Denoising using Wavelet Transform and Mathematical Morphology[J].Geomatics & Spatial Information Technology;2010-03

[7] YAN Yuchen ZHOU Qianjin DUAN Liuyun CHEN Qinghu, Morphologic Severely Polluted Documental Image Processing;Geomatics and Information Science of Wuhan University; 2012-09

[8] FF Zeng, YY Gao, FU Xiao-Ling. Application of denoising algorithm based on document image. Computer Engineering & Design , 2012 , 33 (7) :2701-2705

[9] LIU Yu, ZHANG Yanduo, LU Tongwei.Method of connected domain denoise based on least absolute deviation in OCR.in Vol.33 No.1 of J. Wuhan Inst. Tech,2011

[10] BI Xiao-Jun, W Zhao. Text image denoising based on higher-order cumulant. Applied Science & Technology , 2007

[11] YANG Bo, QI Feihu, HAO Junsheng. New approach for marginal noise removal of binary document image. Computer Engineering , 2006 , 32 (5) :186-188

[12] TIAN Da-zeng, HAO Yong, HA Ming-hu. New algorithm for removal of salt-pepper noises of visual text images. Computer Engineering & Applications , 2007 , 43 (14) :81-83

[13] ZHANG Yuan,CAI Lidong. A Method of Salt-pepper Denoising for Text Images. Journal of Changchun University of Science and Technology(Natural Science Edition), 2010-02

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. in ACL, 2002.