ASAHmap: An Adaptive Chinese Handwritten Character Segmentation Algorithm for Large-Scale Ancient Handwritten Document Based on Histogram Projection and Gaussian Kernel Convolution Map

Ruiyang Song[1], Fuhao Guo[1], Yunchang Wang[1], Hongqi Han[2], Jishou Ruan[1], and Cihan Ruan[3]
[1]School of Mathematical Sciences, Nankai University, Tianjin 300071 China
[2]Institute of Science and Technical Information of China, Beijing 100038 China
[3]Department of Computer Science Engineering, Santa Clara University, Santa Clara, CA 95050 USA
ww_sry@163.com, fuhao.guo.nku@gmail.com, 949085952@qq.com, hanhq@istic.ac.cn, jsruan@nankai.edu.cn, cruan@scu.edu
Corresponding Author: Cihan Ruan Email: cruan@scu.edu

2023 8th International Conference on Image, Vision and Computing (ICIVC) | 979-8-3503-3523-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIVC58118.2023.10269964

***Abstract*—This paper presents a method for recognizing handwritten Chinese characters on ancient documents using optical character recognition (OCR). The method employs a segmentation algorithm based on histogram projection and Gaussian kernel convolution map, which accurately divides images of characters into sub-images of a single character. The algorithm was tested on a dataset of over one million Chinese handwritten characters and achieved a segmentation accuracy of over 97.75% in the best-case scenario for four categories of ancient documents. The proposed method provides a lightweight preprocessing technique for subsequent work aimed at recognizing ancient Chinese handwritten character documents using a neural network.**

***Index Terms*—ancient documents, Chinese character segmentation, Chinese character recognition, histogram projection, Gaussian kernel convolution map**

I. Introduction

Chinese boasts one of the largest character systems among all the world's languages. According to the *Dictionary of Chinese Character Variants*, published by the Taiwan Ministry of Education in 2004, the number of recorded Chinese characters reached a staggering 106,230 at one point. One fundamental difference between Chinese and other popular languages is that Chinese characters are logograms rather than letters. Over the course of history, many Chinese characters have undergone various graphical transformations (e.g., Fig. 1), and the actual number of characters that have existed is much larger. Consequently, in the field of pattern recognition, Chinese character recognition (CCR) is a critical application of optical character recognition (OCR).

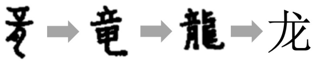


Fig. 1. The revolution of Chinese character "*Dragon*"

Research on CCR has a long history, dating back to 1966 [1]. Today, there are multiple CCR algorithms, such as JieSuOCR [2] and PaddleOCR [3], which can recognize modern Chinese characters with high accuracy and are widely used in commercial applications. However, recognizing ancient Chinese characters, especially those written by hand, is more challenging due to the variety and vast amount of data. Implementing an ancient Chinese handwritten character recognition system requires two crucial components: a large amount of pre-processed data and accurate segmentation of Chinese characters from optical images of ancient books. Character segmentation is a fundamental process of OCR systems that partitions an image of characters into sub-images of a single character. Consequently, CCR on ancient Chinese documents remains a highly challenging subject to tackle.

The segmentation algorithms for Chinese characters can be divided into three categories: traditional methods, machine learning algorithms, and deep learning-based methods. Traditional methods include techniques such as histogram projection-based [4], morphological operation-based [5], stroke-enclosing box-based [6], and clustering-based methods [7]. Machine learning algorithms, such as K-means clustering [8] [9], have also been used. Deep learning-based methods, such as CTPN [10], SegLink [11], and EAST [12], have become increasingly popular in recent years. These algorithms can be evaluated based on four aspects: the desired size of the segmentation unit, cold start stage performance, accuracy, and labor cost.

In general, traditional segmentation methods treat a single character as a unit, while deep learning networks take a whole sentence as input. Therefore, traditional methods perform better in the cold start stage. On the other hand, machine learning algorithms are more adept at handling fuzzy details like blurry strokes or ligatures. However, it's worth noting that deep learning-based methods require a much larger, clean dataset for further training compared to the other two approaches. Ultimately, the choice of algorithm will depend on the specific needs of the application and the available resources.

In recent years, neural networks have demonstrated high effectiveness in CCR due to their ability to handle a vast number of characters with different shapes. Applying centralization and normalization techniques to datasets can further enhance the efficiency of network training, resulting in improved results. Character segmentation is a crucial step in the OCR system for converting handwriting into digital text, particularly for ancient Chinese documents, which pose significant challenges due to their fuzzy layout, inconsistent fonts, and glyphs. Unfortunately, the lack of a reliable large-scale database of ancient Chinese character segmentation is a significant hurdle. Without high-quality labeled datasets, accurately segmenting characters from ancient documents becomes increasingly difficult, despite the effectiveness of neural networks in processing various shapes of characters. However, a possible solution to the labeling problem is the use of semi-supervised learning networks. Nonetheless, high-quality character image segmentation remains a critical issue that cannot be avoided. Therefore, it is crucial to build a qualified dataset for training neural networks and achieving accurate character segmentation in CCR.

Our system is based on the ACCR [13], a semi-supervised auto-labeling neural network we designed for ancient Chinese characters that reduces human effort while maintaining high accuracy. However, to save computing resources and reduce workload, we need to design a lightweight algorithm as a pre-processing step for the semi-supervised neural network. This algorithm is necessary to achieve adaptive Chinese character image segmentation for large-scale ancient documents.

Due to the computational complexity of neural networks, choosing one for segmentation is challenging, and further complicated by the existing conditions and cold start problem. Therefore, we explored the method proposed by Baek *et al.* [14], which uses the Gaussian kernel convolution map to handle single Chinese character segmentation by columns. While this method works well for documents, it is computationally complex when applied to splitting characters from columns into individual character units.

Overall, our decision to avoid using a neural network for segmentation is driven by the desire to save computing resources and improve efficiency. We aim to design a lightweight algorithm that can handle the adaptive Chinese character image segmentation necessary for large-scale ancient documents.

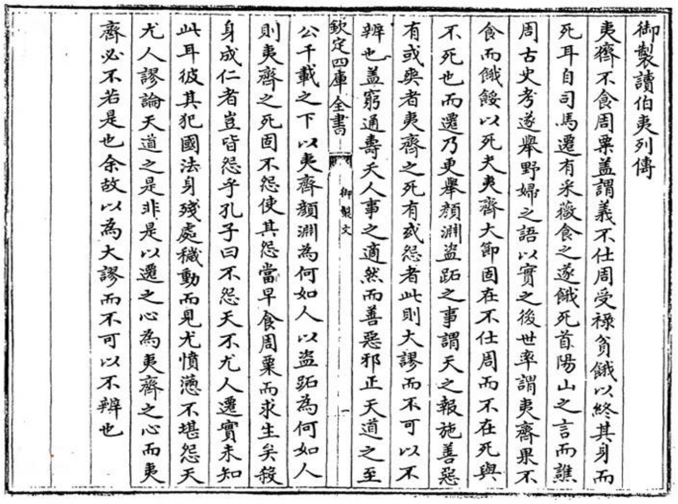
We propose an adaptive algorithm called ASAHmap (Adaptive Chinese Handwritten Character Segmentation Algorithm) to address the challenge of segmenting large-scale ancient handwritten documents. The algorithm is based on histogram projection and Gaussian kernel convolution map. The algorithm employs histogram projection and Gaussian kernel convolution map. Firstly, it applies histogram projection to divide the document into columns. Then, a Gaussian kernel convolution map is used to break each column into smaller blocks, generating a convolution kernel. The convolution is iteratively executed until the kernel width reaches 1 pixel. Subsequently, the algorithm uses the histogram projection method again to convert the convolutional map into a curve. By utilizing the curve vertex, the center of each character is determined, and the entire page of optical document images is segmented into single characters.

The proposed method significantly reduces labor and computing resources while achieving high accuracy in character segmentation for large-scale ancient documents. The remaining content is structured as follows: In the second section, we provide an introduction to the source of the original data and the problems that need to be addressed. Section III provides a detailed explanation of the algorithm based on histogram projection and Gaussian kernel convolution map. In Section IV, we present the results and analysis of our experiments. Finally, we summarize our findings and conclusions in Section V.

II. Data Source and The Proposed Method

A. Data Source Analysis

Our project is centered around two well-preserved types of ancient Chinese documents: poetry and history books. Specifically, we used the *Ri Zang Han Ji* collection, which comprises Han dynasty books held by Japanese private collectors and dates from 202 B.C. to 9 A.D. and 25 to 220 A.D., as well as *Emperor's Four Treasuries*, the largest official collection of Chinese historical books. The poetry dataset contains over 5,000 optical images with more than one million characters, while the history book dataset contains approximately 40,000 images with more than 10 million characters. We selected these documents due to their relatively high-quality, organized layouts, and common fonts. An example of an ancient Chinese book's layout can be seen in Fig. 2.



御製讀伯夷列傳
夷齊不食周粟蓋謂義不仕周受祿貧餓以終其身而
死耳自司馬遷有采薇食之遂餓死首陽山之言而譙
周古史考遂舉野婦之語以實之後世率謂夷齊果不
食而餓餒以死夫夷齊大節固在不仕周而不在死與
不死也而遷乃更舉顏淵盜跖之事謂天之報施善惡
有或與者夷齊之死有或怨者此則大謬而不可以不
辨也蓋窮通壽夭人事之適然而善惡邪正天道之至
欽定四庫全書　御製文　一
公千載之下以夷齊顏淵為何如人以盜跖為何如人
則夷齊之死固不怨使其怨當早食周粟而求生矣殺
身成仁者豈皆怨乎孔子曰不怨天不尤人遷實未知
此耳彼其犯國法身殘處穢動而見尤憤懣不堪怨天
尤人謬論天道之是非是以遷之心為夷齊之心而夷
齊必不若是也余故以為大謬而不可以不辨也

Fig. 2. Example of an optical image of ancient Chinese document

The selected ancient handwritten documents we are working with are valuable historical artifacts. However, they present some quality issues that need to be addressed to accurately segment and extract individual characters from them.

One of the common issues is inconsistent typesetting, where the document's layout changes frequently, making it challenging to determine each character's boundaries. Additionally, mixed font sizes and ligatures can complicate the segmentation process. These issues are further compounded by blurry strokes, atypical punctuation, and the random appearance of stamps and notes. Traditional segmentation algorithms struggle to accurately identify and separate individual characters due to these issues.

To address these challenges, we have designed a targeted algorithm that combines the strengths of histogram projection and Gaussian kernel convolution map techniques. This algorithm is specifically tailored to handle the unique challenges presented by these ancient handwritten documents. It allows us to accurately segment and extract individual characters despite the quality issues.

B. The Proposed Method

The workflow of ASAHmap is shown in Fig. 3, and it consists of three main parts: text enhancement, text detection, and text localization, followed by segmentation.

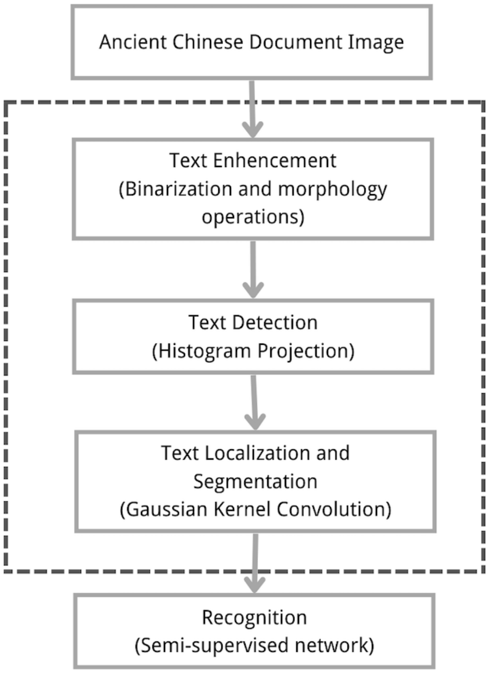
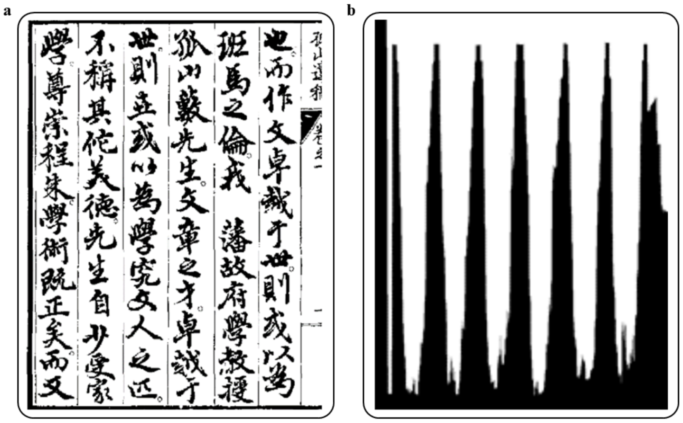

Ancient Chinese Document Image
Text Enhencement (Binarization and morphology operations)
Text Detection (Histogram Projection)
Text Localization and Segmentation (Gaussian Kernel Convolution)
Recognition (Semi-supervised network)

Fig. 3. Workflow of our proposed segmentation algorithm

1) Text Enhancement: Our approach involves enhancing text through binarization and morphological operations. To determine the binarization threshold, we utilize Otsu's method for automatic image thresholding. We then apply the morphological opening operation to filter out the noise and improve the quality of ancient documents.

2) Text Detection: We detect text using histogram projection. This technique is particularly effective for ancient documents written on ruled paper, as it can successfully identify bold line separators. As shown in Fig. 4, there is a positive correlation between the resolution of the bold line separators and the segmentation results. By detecting these separators, we are able to partition the entire page of content into columns.



孤山遺稿　卷之一　一

也。而作文卓識于世。則或以為

班馬之倫哉。　藩故府學教授

孤山藪先生。文章之才。卓識于

世。則並或以為學究文人之匹。

不稱其佗美德。先生自少受家

學。尊崇程朱。學術既正矣。而文

Fig. 4. Example of projection histogram (a) The original input documents' layout (b) The projection histogram result

The histogram projection describes the accumulation of pixels in the current direction. For a binary image $A_{M\times N}$, we represent the histogram projection along the vertical direction of A with $\{p_j\}_{j=1,\ldots,N}$ where p_j denotes the number of pixels with a value of 0 in column j. Fig. 4(b) provides an example of the histogram projection of Fig. 4(a) along the vertical direction.

Although the projection diagram in Fig. 4(b) may appear chaotic at the bottom, it can be easily resolved by applying a mean filter. However, some chaos may remain unresolved due to the random length of the mapping of bold line separators in the projection diagram, or because some of the separators' mappings disappear after filtering. To address this issue, we traced back to the original ancient documents and discovered that the column spacing is basically the same. Based on this, we can mark all the missing bold lines. For documents that have multiple layouts mixed together, as shown in Fig. 5(a), we can analyze the projection diagram of each column of characters, as shown in Fig. 5(b). The number of peaks in a certain column can effectively identify if there are multiple sub-columns. Using the width of each waveform, we can split each column of characters from the documents.

3) Text localization and segmentation: Although histogram projection is useful for identifying columns, it cannot separate each character in a column since it cannot determine the distance between two characters or the size of one character. Therefore, we use a Gaussian kernel convolution map as text localization and segmentation to determine the center of each character, as shown in Fig. 6(a).

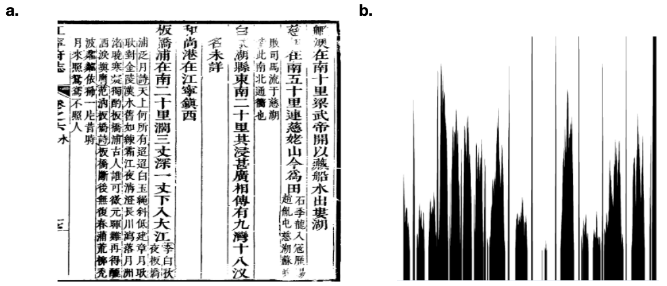


Fig. 5. Different situations of separator identification (a) A case of multiple mixed layouts. (b) The relative projection histogram result

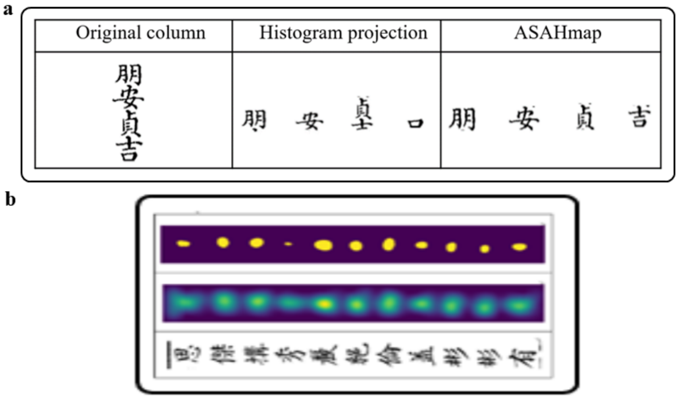


Fig. 6. Gaussian Kernel Convolution Map Explanation (a) Comparison of the two methods' results (b) Sample of Gaussian kernel's result

However, histogram projection cannot achieve a precise split. By using the convolution with a Gaussian kernel, we can accurately determine the center of each character's area, as shown in Fig. 6(b). To simplify the description, we define the n-neighborhood of the current pixel (i, j) mathematically as follows:

Definition 1 For any given positive integer n, a sub-matrix centered with position (i_0, j_0) is described as:

$$i_0 - n \leq i \leq i_0 + n,\ j_0 - n \leq j \leq j_0 + n \tag{1}$$

The $(2n + 1)^2$ pixels are called the *n*-neighborhood of the current pixel (i_0, j_0). With the definition of the *n*-neighborhood, we can analyze the content of documents' square image matrix A consisting of multiple n-neighborhoods.

Definition 2 For any given integer n, we call the typical pixel (i, j) the inner point if the *n*-neighborhood of (i, j) is fully contained in A. Otherwise, we call it a boundary point.

Definition 3 The center of *n*-neighborhood is $(0, 0)$ if a non-negative function h defined on that neighborhood:

$$h\,(i, j) = h_{i,j},\ -n \leq i \leq n,\ -n \leq j \leq n \tag{2}$$

As $\{h_{i,j}\}_{i,j=-n}^{n}$ follows the normalized Gaussian distribution, h is called a Gaussian kernel function. The given image A is equal to an $M \times N$ image matrix. When $1 \leq i \leq n - 1$ or $M - n - 1 \leq i \leq M$ or $1 \leq j \leq n - 1$ or $N - n - 1 \leq i \leq N$, current pixel (i, j) is a boundary point. Because those boundary pixels haven't been assigned any values, we assume they are equal to 1. This also means that we expand the image matrix A by doing one padding to generate a new image of size $(M + 2n) \times (N + 2n)$.

Definition 4 We assume h is the Gaussian convolution kernel, and we name the convolution as padding convolution if the convolutional map $\widehat{A}$ has the same shape as the original picture A. The formula for calculating padding convolution is:

$$a(i, j) = \sum_{k,l} f(i - k, j - l) \cdot h(k, l) \tag{3}$$

where $k, l = -n, ..., n,\ 1 \leq i \leq N,\ 1 \leq j \leq M$.

Definition 5 We assume h is the Gaussian convolution kernel, and we name the convolution that only operates on the inner points of the original picture A as *inner-point convolution*. The formula for calculating *inner-point convolution* is:

$$a(i, j) = \begin{cases} \sum_{k,l} f(i - k, j - l) \cdot h(k, l),\ k, \\ l = -n, \cdots, n;\ n + 1 \leq i \leq N - n; \\ n + 1 \leq j \leq M - n \\ 0,\ i \leq n \textit{ or } M - n + 1 \leq i;\ j \leq n \\ \textit{or } M - n + 1 \leq j \end{cases} \tag{4}$$

For character column segmentation, the *inner-point convolution* offers its own advantages. Let F be a column of characters with size $M \times N$, where N is the estimated character width, and M is the height of the column. Based on the character width N, we can choose the appropriate integer *n* such that $(2n + 1) < N/2$. Then, the shape of the convolutional map $\widetilde{F}$ obtained by inner-point convolution will become smaller, resulting in a column height of $M - 2n$ and character width of $N - 2n$. By iteratively repeating this *inner-point convolution* process, the center part $\{(x_i, y_i)\}_{i=1:K}$ of each character will appear gradually. For each loop, we update the kernel size *n* by $n \leftarrow \left\lceil \frac{N-2n-1}{2} \right\rceil$. The loop will stop while the condition of width $\widetilde{F} \leq 2$ is not met.

Once the final convolution map $\widetilde{F}$ is obtained, we apply histogram projection to convert it to a curve of pixel values as shown in Fig. 7. Apart from two missing characters, all the identified lines pass through the center of the remaining characters. To improve the results, we can use the assumption that the spacing between characters is similar to that of officially published books and store the space distance as prior knowledge for calibration.

III. Results

We created a large dataset of ancient Chinese handwritten documents consisting of four parts selected from book categories to evaluate the performance of ASAHmap. These four parts were taken from the poetry collections *"Bai Shi Yi Gao"* and *"Shan Yang Yi Gao"*, as well as the *"Liang Shu"* and *"Sui Shu"* (sub-volumes of *"Emperor's Four Treasuries"*) separately. In total, the dataset contains over 1.4 million characters for our segmentation work. With a dataset of this size,

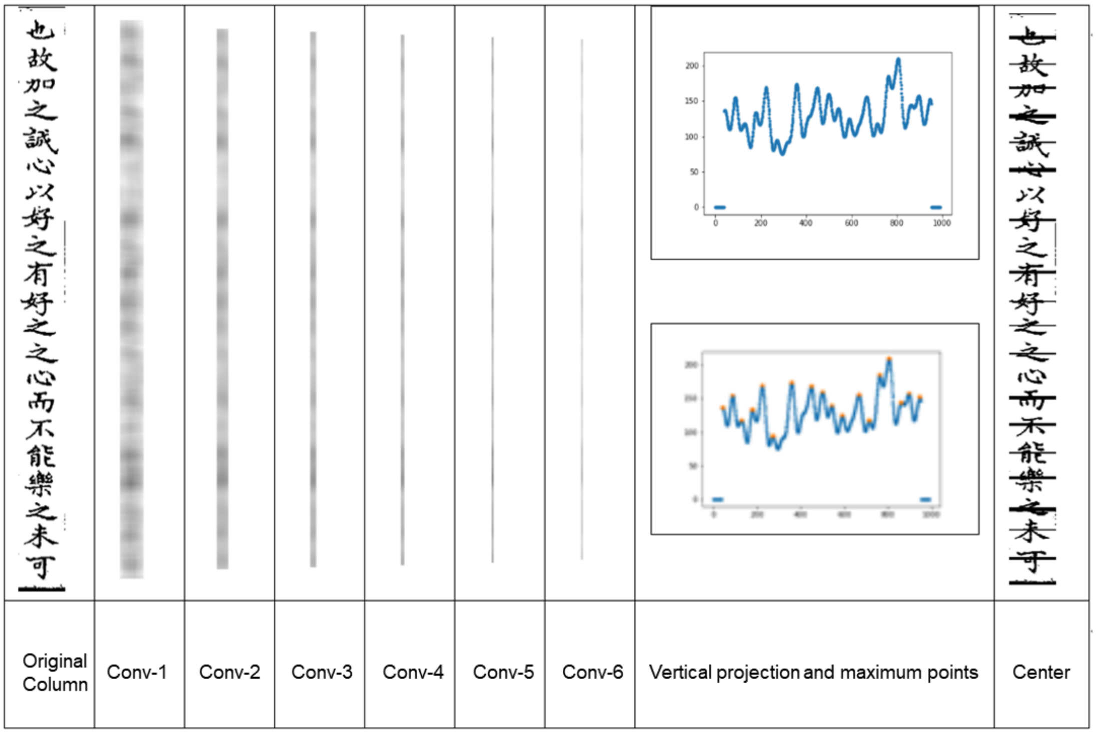


Fig. 7. Schematic diagram of the effect of using multiple iterations of Gaussian kernel convolution map to find the center of each Chinese character in the column.

manual verification is not very feasible. To test the results, we divided the segmented images into six batches and passed them into our semi-supervised auto-labeling network, ACCR, for training. During this process, the neural network automatically picked out unrecognizable images, and 1,189,786 of them could be successfully labeled. We then manually reviewed the cases in which segmentation failed to measure the effect of segmentation. The result is shown in Table I.

TABLE I
COMPARISON OF ASAHMAP'S SEGMENTATION ACCURACY WITH OTHER METHODS

Dataset	ASAHmap	Histogram projection	paddleOCR
Bai Shi Yi Gao	0.8647	0.4384	0.6665
Shan Yang Yi Gao	0.9385	0.3142	0.7569
Liang Shu	0.9071	0.7201	0.8370
Sui Shu	0.9775	0.2921	0.8723

Our evaluation results show that ASAHmap is superior to the traditional histogram projection method, particularly on the textit“Sui Shu” dataset where the segmentation accuracy is 3.35 times higher. Additionally, when combined with ACCR, ASAHmap achieves even higher recognition accuracy on the same dataset. In comparison to PaddleOCR, which is a well-known commercial software with excellent performance in modern Chinese, our algorithm outperforms it on all four ancient Chinese datasets. It is important to note that ASAHmap has a simpler structure than other methods, which makes it more efficient in terms of computing resources and human effort.

IV. CONCLUSIONS

Character segmentation is a crucial pre-processing step in developing a Chinese character recognition system. The accuracy of recognition results can be significantly affected by the quality of the segmentation. Moreover, a lightweight algorithm can save labor and energy.

In this paper, we propose ASAHmap, a segmentation algorithm designed for large-scale ancient Chinese handwritten documents. By combining histogram projection and Gaussian kernel convolution map, ASAHmap achieves segmentation by columns and characters in succession. The algorithm is applied to construct a dataset of over one million ancient Chinese handwritten characters from optical pictures of ancient documents. ASAHmap achieves a success rate of over 97.75% in the best-case scenario for four categories of ancient documents. The combination of ASAHmap and ACCR resulted in a recognition rate of 81.46% for the characters in the four categories of ancient documents. This was achieved after validation by our auto-labeling neural network ACCR.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Xin Yang, Li Wang, and Dr. Jun Xu for their contributions to this paper, as well as to Tianyu Du, Lin Wang, and Haoyu Wu for their assistance. This research was supported by the Institute of Science and Technical Information of China under the project ”OCR image detection and recognition of ancient characters based on deep learning mode” (Grant No. ZD2020/09-03).

REFERENCES

[1] R. Casey and G. Nagy, "Recognition of printed chinese characters," *IEEE Transactions on Electronic Computers*, no. 1, pp. 91–101, 1966.

[2] "JieSuOCR official website," http://https://www.jsocr.com/.

[3] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang *et al.*, "Pp-ocr: A practical ultra lightweight ocr system," *arXiv preprint arXiv:2009.09941*, 2020.

[4] M. J. Swain and D. H. Ballard, "Indexing via color histograms," in *Active perception and robot vision*. Springer, 1992, pp. 261–273.

[5] R. P. dos Santos, G. S. Clemente, T. I. Ren, and G. D. Cavalcanti, "Text line segmentation based on morphology and histogram projection," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 651–655.

[6] H. Wang and Y. Jiang, "Offline handwritten chinese character segmentation algorithm based on stroke bounding box," *Computer engineering and design*, vol. 026, no. 003, pp. 803–806, 2005.

[7] R. Ma and Y. Jiang, "An effective multi-step segmentation method for handwritten chinese characters," *Chinese journal of image and graphics*, vol. 12, no. 11, p. 6, 2007.

[8] Y. Zhu, Y. Lu, and H. Li, "Research and implementation of text segmentation algorithm based on k-means clustering," *Computer Applications and Software*, vol. 2, p. 5, 2013.

[9] X. Tian, T. Sun, and Y. Qi, "Ancient chinese character image segmentation based on interval-valued hesitant fuzzy set," *IEEE Access*, vol. 8, pp. 146 577–146 587, 2020.

[10] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European conference on computer vision*. Springer, 2016, pp. 56–72.

[11] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2550–2558.

[12] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[13] P. Wu, X. Yang, F. Guo, L. Wang, and C. Ruan, "Accr: Auto-labeling for ancient chinese handwritten characters recognition on cnn," in *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2022, pp. 1–5.

[14] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365–9374.