Neurocomputing 350 (2019) 271–281

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Weakly supervised precise segmentation for historical document images

Zecheng Xie[a,1], Yaoxiong Huang[a,1], Lianwen Jin[a,*], Yuliang Liu[a], Yuanzhi Zhu[a], Liangcai Gao[b], Xiaode Zhang[b]

[a] College of Electronic and Information Engineering, South China University of Technology, Guangzhou, China
[b] ICST, Peking University, Beijing, China

ARTICLE INFO

Article history:
Received 20 October 2018
Revised 26 February 2019
Accepted 2 April 2019
Available online 25 April 2019

Communicated by Dr. Z. Wang

Keywords:
Weakly supervised learning
Recognition-guided
Historical document images segmentation

ABSTRACT

With the passing of history, precious cultural heritage was left behind to tell ancient stories, especially those in the form of written documents. In this paper, a weakly supervised segmentation system with recognition-guided information on attention area, is proposed for high-precision historical document segmentation under strict intersection-over-union (IoU) requirements. We formulate the character segmentation problem from Bayesian decision theory perspective and propose boundary box segmentation (BBS), recognition-guided BBS (Rg-BBS), and recognition-guided attention BBS (Rg-ABBS), progressively, to search for the segmentation path. Furthermore, a novel judgment gate mechanism is proposed to train a high-performance character recognizer in an incremental weakly supervised learning manner. The proposed Rg-ABBS method is shown to substantially reduce time consumption while maintaining sufficiently high precision of the segmentation result by incorporating both character recognition knowledge and line-level annotation. Experiments show that the proposed Rg-ABBS system significantly outperforms traditional segmentation methods as well as deep-learning-based instance segmentation and detection methods under strict IoU requirements.

1. Introduction

In the course of thousands of years of history, our predecessors left behind a large number of historical documents that contain valuable information on historical knowledge and literary arts. However, after years of storage, historical document collections encounter serious degradation [1] via staining, tearing, ink seepage, etc. The problem of how to preserve this priceless culture heritage for the next generation has received intense interest from numerous researchers [2–4]. Historical document digitization can protect printed paper documents from the effect of direct manipulation for consulting, exchanging and remote access purposes. Typically, historical documents are digitized through photographing, followed by document segmentation, recognition, preservation, management, and research. Among all the above-mentioned stages, document segmentation is conducted as a first step and the overall digitization performance of the system heavily depends on the segmentation quality. Therefore, for historical document analysis, highly precise bounding boxes, i.e., with high intersection-over-union (IoU) values, for the characters, are required to facilitate the execution of subsequent research. In general, document segmentation consists of three principal stages: document layout analysis, text line segmentation and character segmentation, as illustrated in Fig. 1.

Document layout analysis, i.e., page segmentation, the prerequisite step for document image analysis and understanding, is to separate a document image into regions of interest [5]. Traditional document layout analysis methods mainly rely on hand-crafted features [6,7], prior knowledge [7,8], or their hybrid information [9,10]. Although these methods are efficient and useful for some specific document styles, most of them cannot easily be generalized to other layout situations. Recently, deep-learning-based methods have demonstrated excellent capability in semantic segmentation [11]. Pixel-wise segmentation with fully convolutional networks [5,12,13] or hybrid convolutional multidimensional

* Corresponding author.
E-mail addresses: zcheng.xie@gmail.com (Z. Xie), hwang.yaoxiong@gmail.com (Y. Huang), lianwen.jin@gmail.com (L. Jin), liu.yuliang@mail.scut.edu.cn (Y. Liu), z.yuanzhi@foxmail.com (Y. Zhu), glc@pku.edu.cn (L. Gao), zhangxiaode@pku.edu.cn (X. Zhang).
[1] These authors contributed equally.

https://doi.org/10.1016/j.neucom.2019.04.001

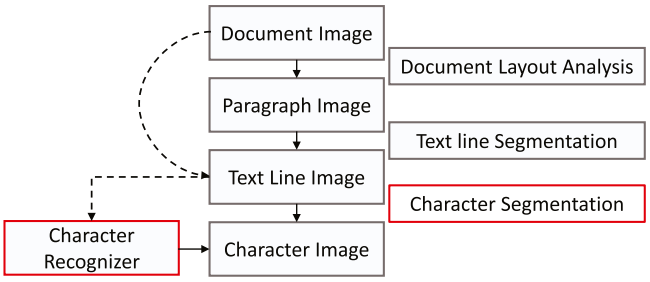


Fig. 1. A high level flow diagram depicting document segmentation. The stages in the red boxes are the main focus of this study.

long short-term memory (MD-LSTM) networks [14] have been introduced in document layout analysis, significantly improving segmentation performance.

Text line segmentation is the division of document images or paragraph images into individual text line images for subsequent character segmentation or text line recognition. However, a text line is hard to define, especially for historical handwritten documents. Previous researchers provided three types of approaches, namely baseline [15], bounding box [16], and X-Height [17] methods. A survey of existing methods developed during the last decades for text line segmentation of historical documents has been presented in [18]. Recently, a new scheme [19] for historical manuscripts, consisting of minimum filtering, average block projecting, segmentation path selection and nonlinear segmentation path construction, was proposed for binarization-free text line segmentation. However, previous methods usually rely on heuristic knowledge and handcraft features, making them less widely applicable and scalable. By contrast, deep-learning-based methods show much more robust and effective performance on complex and noisy page images. Renton et al. [20] proposed a dilated convolutions-based variant of deep fully convolutional network (FCN) for handwritten text line segmentation. Without manual parameter tuning or heuristics, Breuel and Robust [14] combined convolutional and MDLSTM networks to achieve fast yet reliable text line segmentation.

Character segmentation is an operation that seeks to decompose text images into subimages of individual symbols [21]. The precision of character segmentation has significant influence on the subsequent processes, such as historical document preservation, backtracking, research and discovery. After precise character segmentation, people can easily backtrack the image samples from different dynasties for a specific character. In this way, historians can easily study the historical evolution of a specific character by analyzing character difference in structure. Furthermore, calligrapher, or even ordinary people, can take inspiration from the character font structure, thanks to the convenient access to the historical character image. However, except in the cases of boxed or clearly spaced characters, segmenting characters independently from the recognition process yields poor recognition performance [22]. Therefore, character recognizer is usually incorporated in the character segmentation step to ensure high-precision segmentation results [22,23]. Note, however, that character recognition and character segmentation rely on each other, and a circular dependency is created between them, referred to as Sayre's paradox [24]. Sayre's paradox yields a problem for our consideration:

When we are interested in a new kind of historical document with only text line images and its text-level annotation available, how can we train a character recognizer from scratch?

Previous methods [22,23,25,26] simply neglect this problem and directly use an existing character dataset for training, which can barely be satisfied in practice. However, character segmentation for text line images is generally unsatisfactory without the aid of recognition knowledge. If line-level annotations are directly assigned to such segments, the resulting character dataset would have a large proportion of erroneous samples, including mis-segmentation and mis-labeled characters. Therefore, how to effectively train a character recognizer with inaccurately labeled character samples is a problem worthy of study.

In this paper, we propose a weakly supervised precise segmentation system for historical document images, as detailed in Fig. 2. The proposed system mainly consists of four stages, including preprocessing, boundary box segmentation (BBS), incremental weakly supervised learning and recognition-guided attention boundary box segmentation (Rg-ABBS), with the following distinctive contributions:

- The character segmentation problem is formulated from the perspective of Bayesian decision theory. Through maximizing the posterior probability of class sequence given text line image, we derive three new algorithms to search for the segmentation path, i.e., BBS, Rg-BBS, and Rg-ABBS, progressively.
- We proposed a judgment gate (JG) mechanism that enables incremental weakly supervised learning on character recognition network (character recognizer) that can provide reliable character recognition score to improve character segmentation results.
- The proposed Rg-ABBS significantly reduces time consumption by performing recognition-guided segmentation only on 'attention' area and achieves comparable performance in comparison to performing recognition-guided segmentation on the whole text line image, i.e., Rg-BBS.
- The system is comprehensive, including line and character segmentation for historical images, that provides sufficiently precise bounding boxes for characters, even under high IoU requirements.

The reminder of this paper is organized as follows. Section 2 reviews the related works. Section 3 formally defines the problem of historical document segmentation. Section 4 details the proposed segmentation methods: BBS, Rg-BBS, and Rg-ABBS, progressively, and presents a novel weakly supervised learning method for character recognizer. Section 5 presents the experimental results. Finally, Section 6 concludes the paper.

2. Related work

In this section, we describe work in the literature related to three aspects of our own, specifically, segmentation methods, approximate string matching approaches, and weakly supervised learning strategy. For segmentation, we review works on projection-based [19,27–29], grouping [30,31], recognition-based [22,23,32] and deep-learning-based [33–37] methods.

For projection-based segmentation methods, projection profiles are obtained by accumulating pixel values along a particular axis. Variants for obtaining a profile curve include connected components and projecting black/white transitions [27], rather than pixels. Furthermore, a smoothed profile curve via Gaussian or median filtering was also applied to eliminate local maxima [28]. Considering the efficiency of projection-based methods, we apply vertical projection to slice the document image into text line images for text line segmentation.

Grouping methods build alignments by aggregating units using a bottom-up strategy, where the units may be pixels or higher-level units, such as units from boundary detection [38] or connected components [31]. For handwritten pages and historical documents, Likforman-Sulem and Faure [30] proposed an iterative method based on perceptual grouping for forming alignments. In [31], Le et al. proposed connected-component-based segmentation (CCS) by introducing a classifier to further evaluate whether the connected component is text or not. In our implementation,

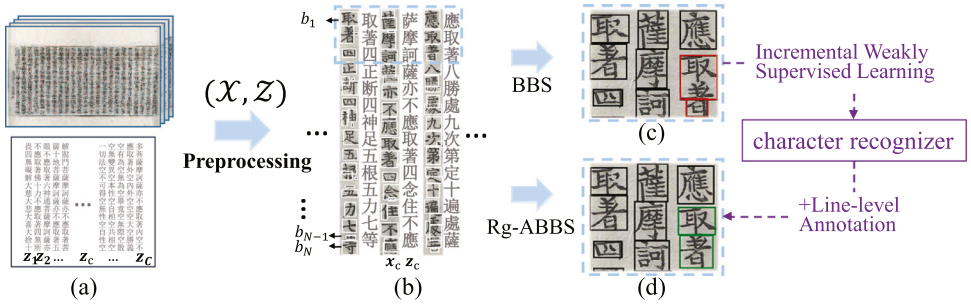


Fig. 2. Overview of the proposed architecture. The input document image $\mathcal{X}$ is accompanied with label $\mathcal{Z}$ that is organized line by line $\mathcal{Z} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_C})$, as shown in (a). During the preprocessing stage, document image $\mathcal{X}$ is first split into text line images $\mathcal{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_C})$ and each text line image $\boldsymbol{x_c}$ is assigned its corresponding label $\boldsymbol{z_c}$. Then the text line image will be further decomposed into segments $\mathfrak{B} = (b_1, b_2, \ldots, b_N)$, as shown by the white bounding box in (b). After that, boundary box segmentation (BBS) is applied for coarse character segmentation, then a weakly labeled data pool (see Fig. 7(a)) is created for character recognizer that is trained in an incremental weakly supervised learning manner. Finally, the character recognizer is incorporated with line-level annotation during recognition-guided attention boundary box segmentation (Rg-ABBS) to rectify the mis-segmented characters (compare (c) with (d)) and output the final segmentation results.

boundary detection [38] is adopted to over-segment the line images into strokes or radicals, because this algorithm is classic and easy to reproduce.

For character segmentation, recognition-based methods [22,23,32] usually provide more precise segmentation results. In [22], Schenkel et al. reported the segmentation performance of on-line hand-printed capital Latin character, with a neural network recognition engine and a graph-algorithmic post-processor. Another method proposed by Daifallah et al. [23] is based on arbitrary segmentation, followed by segmentation enhancement, consecutive joints connection and finally segmentation point locating. However, these methods did not formulate recognition-guided segmentation problems as an objective function; thus, they cannot easily provide systematic analysis of the problem. In our paper, we formulate recognition-guided segmentation problem from Bayesian decision theory perspective, and provide different types of algorithm to search for the segmentation paths, from coarse to finer.

For deep-learning-based segmentation methods, we investigate instance segmentation methods [11,33,34] and object-detection-based methods that can be divided into region proposal methods [39–41] and regression-based methods [35–37,42,43]. These methods leverage the outstanding capability of deep learning networks for feature representation, but are heavily dependent on large labeled training datasets, which is barely satisfied for historical image segmentation.

Approximate string matching [44], in its most general form, is an algorithm finding strings that match a pattern approximately, allowing a limited number of 'errors' in the matches. Typical solutions include edit distance [45], Hamming distance [46], block distance [47] methods. In our problem, we compare our character prediction on segments with text-line annotation through edit distance to identify the mismatched parts, namely the 'attention' areas. As detailed in Section 4.4, we perform recognition-guided segmentation only on the attention area to save time.

Unfortunately, there is only a weakly labeled data pool (see Figs. 2 and 7(a)) to train the character recognizer, in which the supervision information is not always ground-truth, i.e., some label may suffer from error. This is typically called *inaccurate supervision* [48]. By assuming that labels are subject to random noise, researchers [49] have directly optimized the classifier with noise label. Another approach [50] is to identify and eliminate or correct mislabeled training instances for supervised learning. Typically, *data editing* method [51] removes or relabels a suspected instance, when the proportion of examples of the same class in a geometrical graph is not significantly large enough. However, such techniques heavily rely on neighborhood information to make decision. Therefore, they are less reliable in high-dimensional feature space, especially when data are sparse [48]. Partially inspired by the above-mentioned methods, we develop an incremental weakly supervised learning strategy based on the prediction probabilities to identify and relabel mis-labeled data.

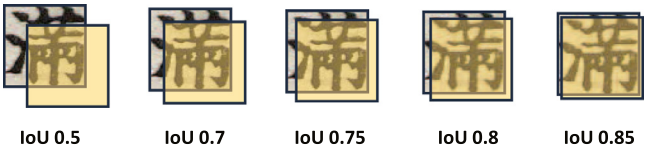


Fig. 3. Illustration of overlap area between the ground truth and predicted bounding box with respect to IoU ranging from 0.5 to 0.85.

3. Problem definition

Given a historical page image and its label $(\mathcal{X}, \mathcal{Z})$, where $\mathcal{Z} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_C})$ as shown in Fig. 2(a), the problem is how to locate every character inside the image by providing each character a bounding box. Note that our label $\mathcal{Z}$ is page-level annotation in which line-level annotations are organized line by line $(\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_C})$, where C is the line number.

Traditional detection methods [36,37,43,52] consider an object/text to be found when the overlap area between the bounding box and the object/text, i.e., IoU, is larger than 0.5. However, this criterion is not friendly for historical document segmentation, because character segmentation precision has a profound effect on subsequent processes, such as historical image preservation, management, research, and discovery. For example, with character segmentation results from historical documents from different dynasties, historians can easily compare different samples for a particular character and study its evolution by analyzing differences in structure. Specifically, in Fig. 3, we demonstrate an example of overlap area between the ground truth and predicted bounding boxes, illustrating IoU ranging from 0.5 to 0.85. It can be observed that an IoU of 0.5 is not sufficient to meet the strict requirements of historical image segmentation problem, because key radicals may be lost and the remainder can barely be recognized. IoUs of 0.7 and 0.75 also inadequately meet the requirement while IoUs equal to or greater than 0.8 are approximately adequate. Therefore, in this paper, we mainly focus on bounding box predictions with IoU values larger than 0.7, especially those equal to or larger than 0.8.

3.1. Problem formulation

The objective function of the historical document image segmentation problem can be formulated from Bayesian decision view

[25]. In Fig. 2(b), text line image $\boldsymbol{x}$ is segmented into *bounding boxes* of segments $\mathfrak{B} = (b_1, b_2, \ldots, b_N)$ after preprocessing. Denote $\boldsymbol{y}$ as a segmentation path indicating how to merge $\mathfrak{B}$ into bounding boxes of characters $\mathfrak{B}' = (c_1, c_2, \ldots, c_T), T \le N$. The posterior probability of being recognized as the class sequence $\boldsymbol{l} = (l_1, l_2, \ldots, l_T)$ given text line image $\boldsymbol{x}$ can be formulated as follow:

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{y}} P(\boldsymbol{l}, \boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}) P(\boldsymbol{l}|\boldsymbol{y}, \boldsymbol{x}), \tag{1}$$

where $P(\boldsymbol{y}|\boldsymbol{x})$ represents the posterior probability of the $\boldsymbol{y}$th path given the text line image, and $P(\boldsymbol{l}|\boldsymbol{y}, \boldsymbol{x})$ denotes the posterior probability of the class sequence $\boldsymbol{l}$ given the $\boldsymbol{y}$th path and the text line image. The objective function is to search for the optimal segmentation path:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y},\boldsymbol{l}} P(\boldsymbol{l}|\boldsymbol{x}) = \arg\max_{\boldsymbol{y},\boldsymbol{l}} P(\boldsymbol{y}|\boldsymbol{x}) P(\boldsymbol{l}|\boldsymbol{y}, \boldsymbol{x}) \tag{2}$$

Furthermore, the posterior probability of the class sequence $\boldsymbol{l}$ can be decomposed as:

$$P(\boldsymbol{l}|\boldsymbol{y}, \boldsymbol{x}) = \prod_{t=1}^{T} p(l_t|c_t^u) p(l_{t-1} l_t|c_t^b) p(l_t|c_t), \tag{3}$$

where c_t^u and c_t^b represent unary and binary outline geometric features of bounding boxes, respectively (see Fig. 5 for more detail). Since $P(\boldsymbol{y}|\boldsymbol{x})$ in objective function Eq. (2) follows uniform distribution in our implementation, it has no influence on the choice of $\boldsymbol{y}$; thus, can be removed from the objective function. Then, substituting Eq. (3) to objective function Eq. (2) gives the general objective function of text line image segmentation:

$$\begin{aligned} \boldsymbol{y}^* &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \prod_{t=1}^{T} p(l_t|c_t^u) p(l_{t-1} l_t|c_t^b) p(l_t|c_t) \\ &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \log \left(\prod_{t=1}^{T} p(l_t|c_t^u) p(l_{t-1} l_t|c_t^b) p(l_t|c_t) \right). \end{aligned} \tag{4}$$

In Section 4, we extend the general objective function Eq. (4) to some specific cases according to practical requirements, and will explain each item in the objective function in detail.

4. Approach

Towards high-precision segmentation of historical documents, we propose a comprehensive segmentation system, as illustrated in Fig. 2, consisting of four parts: preprocessing stage to segment the page image into radical components, boundary box segmentation (BBS) to provide an approximate segmentation result, incremental weakly supervised learning to train a high-performance character recognizer, and finally recognition-guided attention boundary box segmentation (Rg-ABBS) to provide extremely precise segmentation result.

4.1. Preprocessing

In this stage, vertical projection is applied to slice the page image $\mathcal{X}$ into line images $\mathcal{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_C})$, so that we can obtain the line image-annotation pair $(\boldsymbol{x}, \boldsymbol{z})$. As shown in Fig. 4, we first vertically project the image page onto the x-axis to derive the projection profile, then follow the way proposed in [53] to extract line images. Next, boundary detection [38] is adopted to over-segment the line images into strokes or radicals, as shown in Fig. 2(b). Note that our main framework can be effectively integrated with other over-segmentation methods, but we choose traditional boundary detection method to make our approach easier to reproduce. After boundary detection, text line image $\boldsymbol{x}$ is over-segmented into *bounding boxes* $\mathfrak{B} = (b_1, b_2, \ldots, b_N)$ with N indicating box number.

4.2. Boundary box segmentation (BBS)

Before incremental weakly supervised learning, character recognizer has not yet been trained; thus, the recognition confidence score of the character recognizer, i.e., $p(l_t|c_t)$ in Eq. (4), is not provided. In BBS, $p(l_t|c_t)$ is assumed to be uniform distribution, and that may explain why the segmentation result of BBS is relatively coarse. Formally, the objective function of BBS can be transformed from the general objective function Eq. (4) by setting $p(l_t|c_t) = 1$ as follows:

$$\begin{aligned} \boldsymbol{y}^* &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \log \left(\prod_{t=1}^{T} p(l_t|c_t^u) p(l_{t-1} l_t|c_t^b) p(l_t|c_t) \right). \\ &\overset{p(l_t|c_t)=1}{=} \arg\max_{\boldsymbol{y},\boldsymbol{l}} \log \left(\prod_{t=1}^{T} p(l_t|c_t^u) p(l_{t-1} l_t|c_t^b) \right) \\ &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \sum_{t=1}^{T} \sum_{i=1}^{2} \lambda_i \omega_{ti} \end{aligned} \tag{5}$$

where $\omega_{t1} = \log p(l_t|c_t^u)$ estimates the unary outline geometric features of bounding boxes (refer to A_{height} and A_{width} in Fig. 5), $\omega_{t2} = \log p(l_{t-1} l_t|c_t^b)$ estimates the binary outline geometric features of bounding boxes (refer to A_{pad} in Fig. 5), λ_i are weights to balance the effects of these components in practical applications.

Next, we briefly introduce how we estimate the outline geometric features, i.e., ω_{t1} and ω_{t2}, of bounding boxes in BBS. For example, as illustrated in Fig. 5, (T_x^A, T_y^A) and (P_x^A, P_y^A) are the coordinates of the upper-left and lower-right corner of bounding box A, respectively, while (T_x^B, T_y^B) and (P_x^B, P_y^B) are those of box B. The *Self-Score* of bounding box A is formulated as follows:

$$S_{ss}(A) = 4 \cdot \left(\left(\mathbb{I}\{A_{height} < \mathcal{H}\} - \frac{1}{2} \right) + \left(\mathbb{I}\{A_{pad} > \mathcal{P}\} - \frac{1}{2} \right) \right)$$

where $\mathbb{I}\{condition\}$ equals to 1 when *condition* is true and 0 otherwise, $A_{height} = P_y^A - T_y^A$ is the height of bounding box A, $A_{pad} = T_y^B - P_y^A$ is the distance between A and its next adjacent bounding box B; $\mathcal{H}$ and $\mathcal{P}$ represent the maximum merging height threshold and minimum distance threshold, e.g., 90 and 8 pixels, respectively. Note that we assume $A_{pad} > \mathcal{P}$ for the last bounding box. Let $\hat{\boldsymbol{y}}$ represent the prefix of path $\boldsymbol{y}$ with the last element removed and $\boldsymbol{y}_e$ denote the last element of $\boldsymbol{y}$. The *Accumulative-Score* of path $\boldsymbol{y}$ in step t is thus defined as:

$$S_{acc}(\boldsymbol{y}, t) = \begin{cases} S_{acc}(\hat{\boldsymbol{y}}, t-1) + S_{ss}(\boldsymbol{y}_e), & t > 1 \\ S_{ss}(\boldsymbol{y}_1), & t = 1. \end{cases} \tag{6}$$

In our implementation, we have $S_{ss}(c_t) = \lambda_1 \omega_{t1} + \lambda_2 \omega_{t2}$ (ω_{ti} is the same in Eq. (5)) and $S_{acc}(\boldsymbol{y}, T) = \sum_{t=1}^{T} S_{ss}(c_t)$, hence the objective function for BBS can be transformed as:

$$\begin{aligned} \boldsymbol{y}^* &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \sum_{t=1}^{T} \sum_{i=1}^{2} \lambda_i \omega_{ti} \\ &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \sum_{t=1}^{T} S_{ss}(c_t) \\ &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} S_{acc}(\boldsymbol{y}, T). \end{aligned} \tag{7}$$

Therefore, the problem is to search for the optimal path $\boldsymbol{y}^*$ with maximum *Accumulative-Score* $S_{acc}(\boldsymbol{y}^*, T)$. Although the optimal path can be found by dynamic programming, it would consume considerable time and effort. In practice, to strike a trade-off between accuracy and efficiency, we apply beam search [54] to find an approximately optimal solution for the objective function

Fig. 4. Some examples of text line segmentation. Page images are vertically projected onto x-axis to construct the projection profile. After that, line images are extracted based on the projection profile, following the method proposed in [53].

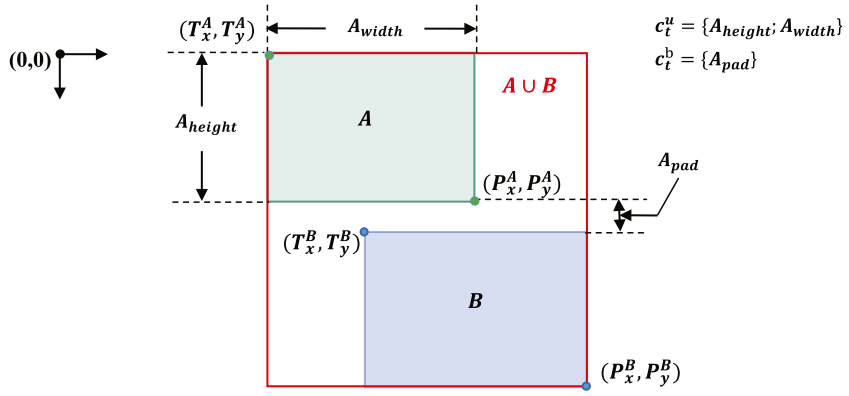


Fig. 5. Illustration of the notation for bounding box A and the subsequent adjacent bounding box B. c_t^u and c_t^b represent unary and binary outline geometric features of bounding boxes, respectively (see Eq. (3) for more detail).

Eq. (7), i.e., the proposed BBS in Algorithm 1. Through the proposed BBS, we are able to efficiently merge the bounding boxes

Algorithm 1: Boundary box segmentation (BBS)

Initialise: $P \leftarrow \{\varnothing\}$;
for $n = 1 \dots N$ **do**
  $\widehat{P} \leftarrow$ the W highest scored path in P;
  $P \leftarrow \{\}$;
  for $\boldsymbol{y} \in \widehat{P}$ **do**
    $S_{acc}(\boldsymbol{y} + b_n, n) \leftarrow S_{acc}(\boldsymbol{y}, n-1) + S_{ss}(b_n)$;
    Add $\boldsymbol{y} + b_n$ to P;
    if $\boldsymbol{y} \neq \varnothing$ **then**
      $\boldsymbol{y}'_e \leftarrow \boldsymbol{y}_e \cup b_n$;
      $S_{acc}(\hat{\boldsymbol{y}} + \boldsymbol{y}'_e, n) \leftarrow S_{acc}(\hat{\boldsymbol{y}}, n-1) + S_{ss}(\boldsymbol{y}'_e)$;
      Add $\hat{\boldsymbol{y}} + \boldsymbol{y}'_e$ to P.
Return: $\max_{\boldsymbol{y} \in P} S_{acc}(\boldsymbol{y})$

$\mathfrak{B} = (b_1, b_2, \dots, b_N)$ into characters $\mathfrak{B}' = (c_1, c_2, \dots, c_T)$ for subsequent research.

Although the recognition result of BBS is not as precise as those recognition-guided, it performs fast segmentation without consulting character recognizer, and constitute a key component of Rg-ABBS. Furthermore, after assigning labels to the segmentation result of BBS, $\mathfrak{B}' = (c_1, c_2, \dots, c_T)$, in order using text line annotation, they can be used directly as input of incremental weakly supervised learning, i.e., weakly labeled data pool as shown in Fig. 7(a).

4.3. Recognition-guided boundary box segmentation (Rg-BBS)

Character recognition score is very important in the process of character segmentation, because it can provide information indicating whether the character is appropriately segmented [22,23,32]. However, in practice, we found that deep-learning-based character recognizer sometimes provides high confidence predictions for unrecognizable wrongly-segmented characters [55]. To overcome this problem, we propose recognition-guided boundary box segmentation (Rg-BBS) to incorporate text line annotation as well as recognition score of character recognizer to facilitate character segmentation. Specifically, Rg-BBS is to find the optimal segmentation path $\boldsymbol{y}$ that has the optimal rationality in geometric structure and the highest posterior probability of being recognized as text line annotation $\boldsymbol{z} = (z_1, z_2, \dots, z_T)$ given the path $\boldsymbol{y}$ and the text line image. Therefore, the objective function of Rg-BBS can be transformed from the general objective function Eq. (4) as follows:

$$\begin{aligned} \boldsymbol{y}^* &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \log\left(\prod_{t=1}^{T} p(l_t|c_t^u)\, p(l_{t-1}l_t|c_t^b)\, p(l_t|c_t)\right) \\ &\overset{\boldsymbol{l}=\boldsymbol{z}}{=} \arg\max_{\boldsymbol{y}} \log\left(\prod_{t=1}^{T} p(z_t|c_t^u)\, p(z_{t-1}z_t|c_t^b)\, p(z_t|c_t)\right) \\ &= \arg\max_{\boldsymbol{y}} \sum_{t=1}^{T}\sum_{i=1}^{3} \lambda_i \omega_{ti}, \end{aligned} \tag{8}$$

where ω_{t1} and ω_{t2} are the same as those of BBS, $\omega_{t3} = \log p(z_t|c_t)$ indicates the recognition confidence score of the character recognizer. Unlike the general objective function Eq. (4) in which class sequence $\boldsymbol{l}$ is unknown before segmentation, Rg-BBS sets class sequence $\boldsymbol{l}$ as text line annotation $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$, so that it only considers the characters of text line annotation $\boldsymbol{z}$ when consulting character recognizer. For example, for the tth bounding box in Eq. (8), we require it to have the highest probability of being recognized as the tth character in text line annotation $\boldsymbol{z}$. In Algorithm 2 , we provide detail algorithm based on beam

Algorithm 2: Recognition-guided BBS (Rg-BBS)

Input : $P \leftarrow \{\varnothing\}$;
Text line annotation $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$;
Output: Predicted boundary boxes
for $n = 1 \ldots N$ **do**
  $\widehat{P} \leftarrow$ the W highest scored path in P;
  $P \leftarrow \{\}$;
  for $\boldsymbol{y} \in \widehat{P}$ **do**
    t = $\|\boldsymbol{y}\|$;
    $S_{acc}(\boldsymbol{y} + b_n, n) \leftarrow$
    $S_{acc}(\boldsymbol{y}, n-1) + S_{ss}(b_n) + \lambda_3 Pr(z_{t+1}, b_n)$;
    Add $\boldsymbol{y} + b_n$ to P;
    if $\boldsymbol{y} \neq \varnothing$ **then**
      $\boldsymbol{y}'_e \leftarrow \boldsymbol{y}_e \cup b_n$;
      $S_{acc}(\hat{\boldsymbol{y}} + \boldsymbol{y}'_e, n) \leftarrow$
      $S_{acc}(\hat{\boldsymbol{y}}, n-1) + S_{ss}(\boldsymbol{y}'_e) + \lambda_3 Pr(z_t, \boldsymbol{y}'_e)$;
    Add $\hat{\boldsymbol{y}} + \boldsymbol{y}_e$ to P.
Return: $\max_{\boldsymbol{y} \in P} S_{acc}(\boldsymbol{y})$

$\|\boldsymbol{y}\|$ return merged bounding box number in path $\boldsymbol{y}$.

search to search for the approximately optimal segmentation path. Algorithm 2 is adapted from Algorithm 1 by introducing text line annotation and character recognizer. Note that $Pr(z_t, c_t = b_n/\boldsymbol{y}'_e)$ in Algorithm 2 denote the recognition confidence score of c_t being recognized as tth character in text line annotation $\boldsymbol{z}$, and c_t is tth element of segmentation result of Rg-BBS, $\mathfrak{B}' = (c_1, c_2, \ldots, c_T)$.

4.4. Recognition-guided attention boundary box segmentation (Rg-ABBS)

Although BBS exhibits high efficiency in line image segmentation, its segmentation precision is not sufficient for application. Besides, it completely neglect the character recognition information as well as text line annotation during decoding. On the other hand, Rg-BBS (see Eq (8)) can incorporate character recognition information to promote segmentation result, but waste much time on consulting the character recognizer.

In order to utilize the advantages of BBS and Rg-BBS while discard their drawbacks, we develop recognition-guided attention boundary box segmentation (Rg-ABBS) to help system focus only on the confusing parts. The idea behind Rg-ABBS is to integrate character recognition precisely on the 'attention' area of the text line image, where mis-segmentation problems usually occurs. Therefore, how to identify an attention area becomes the key issue in the character segmentation problem. Fortunately, BBS provide us a coarse segmentation result $\mathfrak{B}' = (c_1, c_2, \ldots, c_T)$. By applying our high-performance character recognizer (see Section 4.5) on $\mathfrak{B}'$, we can derive their corresponding recognition result $\{R_{max}(c_t), t = 1, 2, \ldots, T\}$, where $R_{max}(c_t)$ denotes the highest prediction result. Next, we can perform approximate string matching [44] to identify the attention area by comparing $R_{max}(\mathfrak{B}')$ with text line annotation $\boldsymbol{z}$. Edit distance [45] is an ideal implementation of approximate string matching [44] and is adopted to explore the attention area $\mathfrak{B}_{miss}$ in this paper. The remainder area is denoted as matched area $\mathfrak{B}_{match}$. Formally, the objective function for Rg-ABBS is transformed from Eq. (4) as follow:

$$
\begin{aligned}
\boldsymbol{y}^* &= \arg\max_{\boldsymbol{y},\boldsymbol{l}} \prod_{t=1}^{T} p(l_t|c_t^u)\, p(l_{t-1}l_t|c_t^b)\, p(l_t|c_t) \\
&\overset{\boldsymbol{l}=\boldsymbol{z}}{=} \arg\max_{\boldsymbol{y}} \log\left(\prod_{t=1}^{T} p(z_t|c_t^u)\, p(z_{t-1}z_t|c_t^b)\, p(z_t|c_t)\right) \\
&\overset{ED}{=} \arg\max_{\boldsymbol{y}} \left\{ \prod_{c_t \in \mathfrak{B}_{match}} p(z_t|c_t^u)\, p(z_{t-1}z_t|c_t^b) \right. \\
&\quad \left. + \prod_{c_t \in \mathfrak{B}_{miss}} p(z_t|c_t^u)\, p(z_{t-1}z_t|c_t^b)\, p(z_t|c_t) \right\} \\
&= \arg\max_{\boldsymbol{y}} \{ \sum_{c_t \in \mathfrak{B}_{match}} \sum_{i=1}^{2} \lambda_i \omega_{ti} + \sum_{c_t \in \mathfrak{B}_{miss}} \sum_{i=1}^{3} \lambda_i \omega_{ti} \} \\
&= \arg\max_{\boldsymbol{y}} \{ \underbrace{\sum_{c_t \in \mathfrak{B}_{match}} S_{ss}(c_t)}_{\text{BBS}} + \underbrace{\sum_{c_t \in \mathfrak{B}_{miss}} \{S_{ss}(c_t) + \lambda_3 \omega_{t3}\}}_{\text{Attention Area}} \},
\end{aligned} \tag{9}
$$

where the left-side item $\sum_{c_t \in \mathfrak{B}_{match}} S_{ss}(c_t)$ in Eq. (9) comes directly from the segmentation result of BBS and the right-side item $\sum_{c_t \in \mathfrak{B}_{miss}} \{S_{ss}(c_t) + \lambda_3 \omega_{t3}\}$ reveals how we incorporate recognition information with geometry features. Next, Rg-ABBS is developed to find an approximately optimal solution for objective function Eq. (9), as detailed in Algorithm 3. Note that the mismatched boxes and mismatched labels are represented with $\mathfrak{B}_{miss} = (c_M, c_{M+1}, \ldots, c_{M+P})$ and $\boldsymbol{z}_{miss} = (z_K, z_{K+1}, \ldots, z_{K+Q})$, respectively.

To better illustrated the idea of Rg-ABBS, we provide a typical example in Fig. 6. Given the approximate segmentation result $\mathfrak{B}' = (c_1, c_2, \ldots, c_T)$ from BBS, we start by performing character recognition $R_{max}(c_t)$ on each bounding box c_t, with its confidence score of class q denoted by $Pr(q, c_t)$. Next, we compare the recognition result $\{R_{max}(c_t), t = 1, 2, \ldots, T\}$ with the line-level annotation $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$ of the historical line image using edit distance [45]. Finally, we search for the new segmentation path based on recognition score of character recognizer and text line annotation $\boldsymbol{z}_{miss} = (z_K, z_{K+1}, \ldots, z_{K+Q})$.

Algorithm 3: Rg-ABBS

Input : Boundary boxes $\mathfrak{B}_{miss} = (c_M, \ldots, c_{M+P})$;
mismatched labels $\boldsymbol{z}_{miss} = (z_K, \ldots, z_{K+Q})$;
$P \leftarrow \{\varnothing\}$;
Output: Predicted boundary boxes
for $m = M \ldots M + P$ **do**
  $\widehat{P} \leftarrow$ the W highest scored path in P;
  $P \leftarrow \{\}$;
  for $\boldsymbol{y} \in \widehat{P}$ **do**
    $k = \|\boldsymbol{y}\|$;
    $S_{acc}(\boldsymbol{y} + c_m, m) \leftarrow$
    $S_{acc}(\boldsymbol{y}, m-1) + S_{ss}(c_m) + \lambda_3 Pr(z_{(K+k+1)}, c_m)$;
    Add $\boldsymbol{y} + c_m$ to P;
    if $\boldsymbol{y} \neq \varnothing$ **then**
      $\boldsymbol{y}'_e \leftarrow \boldsymbol{y}_e \cup c_m$;
      $S_{acc}(\hat{\boldsymbol{y}} + \boldsymbol{y}'_e, m) \leftarrow S_{acc}(\hat{\boldsymbol{y}}, m-1) + S_{ss}(\boldsymbol{y}'_e) + \lambda_3 Pr(z_{(K+k)}, \boldsymbol{y}'_e)$;
    Add $\hat{\boldsymbol{y}} + \boldsymbol{y}_e$ to P;
Return: $\mathfrak{B}_{match} \cup \max_{\boldsymbol{y} \in P} S_{acc}(\boldsymbol{y})$

$\|\boldsymbol{y}\|$ return merged bounding box number in path $\boldsymbol{y}$.

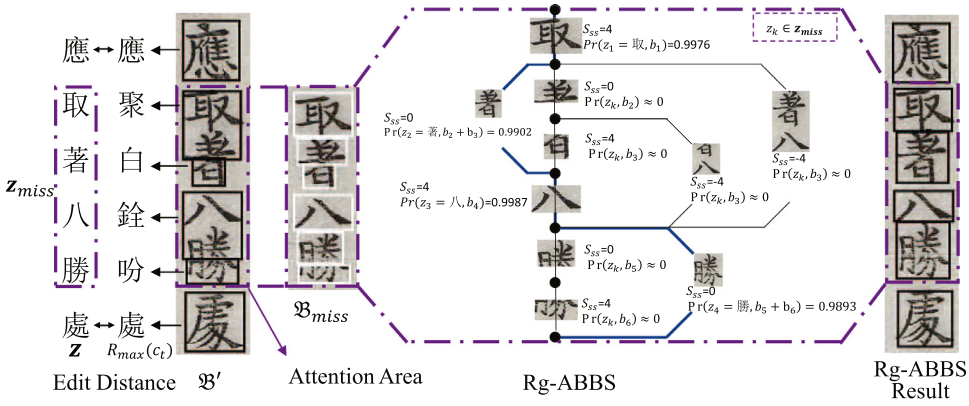


Fig. 6. Illustration of the proposed Rg-ABBS. Given the rough segmentation result $\mathfrak{B}' = (c_1, c_2, \ldots, c_T)$ from BBS, character recognition $R_{max}(c_t)$ is applied to obtain the recognition result for each bounding box c_t. Next, the recognition result $\{R_{max}(c_t), t = 1, 2, \ldots, T\}$ and the line-level annotation $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$ is compared through edit distance [45]. The mismatched boxes and mismatched labels are represented with $\mathfrak{B}_{miss} = (c_M, c_{M+1}, \ldots, c_{M+P})$ and $\boldsymbol{z}_{miss} = (z_K, z_{K+1}, \ldots, z_{K+Q})$, respectively. Finally, we perform beam search to search for better segmentation path. Please refer to objective function Eq. (9) and Algorithm 3 for more details.

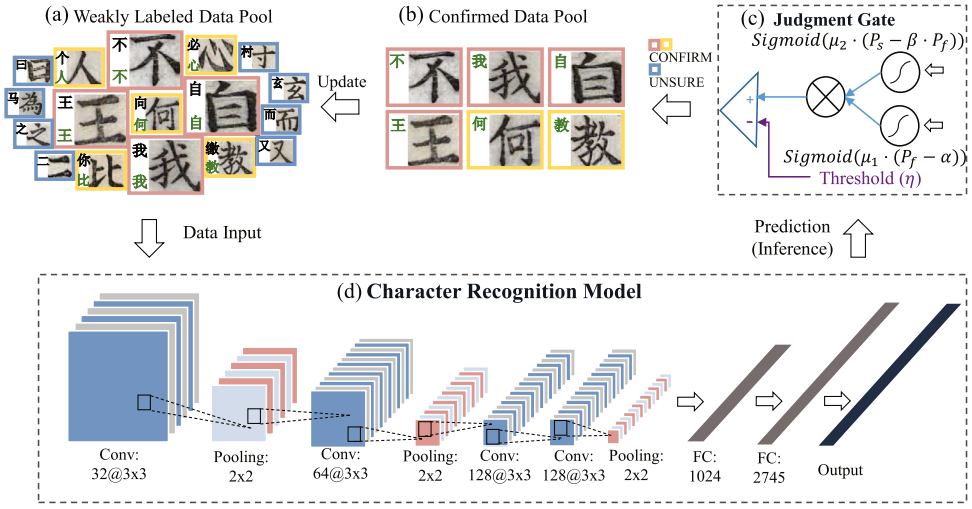


Fig. 7. Illustration of the judgment gate (JG) mechanism and incremental weakly unsupervised learning for the character recognizer.

4.5. Incremental weakly supervised learning

To improve the accuracy of character segmentation, integrating character recognition information during segmentation is a straightforward yet useful approach. In the BBS stage, line images can be efficiently divided into character images $\mathfrak{B}' = (c_1, c_2, \ldots, c_T)$, but the precision of this segmentation is not sufficiently high. This leads to the problem that when we allocate labels to character images based on line-level annotation $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$ in order from top to bottom, i.e., $z_t \rightarrow c_t$, if one position is mis-aligned, the remaining character annotations would be wrong. Therefore, we can only construct a weakly labeled data pool, as shown in Fig. 7(a). We mush note that, in this character-level dataset, we do not know whether a character is correctly annotated, and mislabeled characters occupy a large proportion, preventing us from training a sufficiently high-performance character recognizer.

To solve this *inaccurate supervision* problem [48], one solution is to simply treat the mislabeled sample as noise label [49]. In our situation, mis-labeled samples are too large to be regarded as noise label (nearly half of the characters are mis-labeled). Other researchers proposed data editing method [51] to remove or relabel suspected instances of mislabeling. However, these methods are unstable in high-dimensional sparse-feature space. On the other hand, softmax confidence score has been verified to be a very useful metric for determining the reliability of recognition results [56]. Li and Sethi [57] demonstrated that recognition confidence can be used to determine and correct wrongly labeled samples.

Inspired by the aforementioned works, we aimed to explore the predicted probability distribution of our character recognizer to help identify and relabel suspected instances of mislabeling. During our experiments, we found that the first and second candidate of probability distribution carry most of the information: (1) when a character sample is correctly classified, the character recognition system always assigns not only the highest but also the vast majority of the probability to the ground-truth class, while all the remaining classes are all similarly small or negligible. (2) For wrongly classified samples, the second highest probability of the predicted class is likely to be significantly larger than other classes. Therefore, we propose an incremental weakly supervised learning strategy with judgment gate to train a character recognizer progressively, meanwhile rectifying the mislabeled samples. As shown in Fig. 7(d), we design a character recognizer with 2745 classes as follows:

$$76 * 76 Input - 32C3 - MP2 - 64C3 - MP2 - 128C3 - 128C3 - MP2 - 256C3 - 256C3 - MP2 - 384C3 - 384C3 - FC1024 - FC2745 - Output,$$

where xCy represent convolutional layer with kernel number of x and kernel size of $y * y$, MPx denote max

pooling layer with kernel size of x, and FCx is fully connected layer of kernel number of x.

Judgment gate (JG). As illustrated in Fig. 7(c), the proposed judgment gate consists of three components, including $sigmoid(\mu_1(p_f-\alpha)) = \frac{1}{1-e^{\mu_1(p_f-\alpha)}}$ for evaluating the highest probability p_f of the softmax output of character recognizer, $sigmoid(\mu_2(p_s-\beta\cdot p_f)) = \frac{1}{1-e^{\mu_2(p_s-\beta\cdot p_f)}}$ for evaluating the second highest probability p_s, and judgment evaluation obtained by multiplying these two components together and comparing with a given threshold η:

$$f_{JG} = \begin{cases} \textbf{CONFIRM}, & \Phi \geq \eta \\ \textbf{UNSURE}, & \Phi < \eta \end{cases} \tag{10}$$

where $\Phi = sigmoid(\mu_1(p_f-\alpha)) \cdot sigmoid(\mu_2(p_s-\beta\cdot p_f))$, μ_1 and μ_2 denote adapting parameters; α, β, and η are threshold parameters to guide the flow of the evaluation procedure. Specifically, higher α values make the judgment gate more stringent in consideration of the highest probability while higher β makes the judgment gate put greater weight on the second probability. Together, these values determine the judgment gate, confirming whether or not the label of a sample is correct. The judgment gate guides the training of a character recognizer in a dynamical manner by relabeling the wrongly labeled sample and confirming the correctly labeled data incrementally.

Next, we explain how we apply incremental weakly supervised learning to the training of the character recognizer, as shown in Fig. 7. First, the weakly labeled data pool in Fig. 7(a) is applied to optimize the character recognizer, usually taking tens of epochs. After the recognizer is sufficiently trained, the judgment gate mechanism in Fig. 7(c) classifies the samples into two categories, including **CONFIRM** where character samples are appended to the confirmed data pool in Fig. 7(b), as shown by yellow (relabeled) and orange (not re-labeled) outlined boxes and **UNSURE** where character sample labels remain unchanged, as shown by blue boxes. Lastly, we apply the confirmed data pool in Fig. 7(b) to update the weakly labeled data pool. Formally, we define the abovementioned processes as a *weakly supervised learning iteration*. As the iteration progresses, the weakly labeled data pool is updated incrementally and the character recognizer is optimized accordingly. In practice, the learning process will continue until the weakly labeled data pool is no longer changed or the re-labeled samples number is within tolerance. Note that it is not easy to decide whether a sample should be distinguished as **CONFIRM** or **UNSURE**, because lower η in Eq. (10) will facilitate interfusion of some wrong labeled samples into the confirmed data pool, while higher η will make the optimization longer and convergence more difficult. With the proper value of η, we can maintain the balance between accuracy and efficiency, incrementally adding samples to the confirmed data pool.

Finally, after incremental weakly supervised learning, we manage to learn a high-performance character recognizer from the weakly labeled data pool for Rg-BBS and Rg-ABBS. Experiments show that system segmentation result highly relies on the performance of character recognizer, which in return validates the effectiveness of incremental weakly supervised learning.

5. Experiments

5.1. Dataset

In the following experiments, a historical document dataset, consisting of approximately 160,000 page-labeled images from the Tripitaka Koreana in Han [58], was downloaded from the Internet to evaluate the proposed historical image segmentation system. After preprocessing, we applied the proposed BBS algorithm to construct the weakly labeled data pool consisting of 2745 classes. We randomly selected 10,000 character images and made manual annotation to create a standard testing set for incremental weakly supervised learning of the character recognizer. Furthermore, for character segmentation evaluation, we manually annotated 1000 Tripitaka images, namely TKH Dataset [59],[2] with 700 images for training and 300 for testing the segmentation methods. It is worthy noting that these 700 training images is specially annotated for deep-learning-based detection and instance segmentation methods, and not used for BBS, Rg-BBS, or Rg-ABBS.

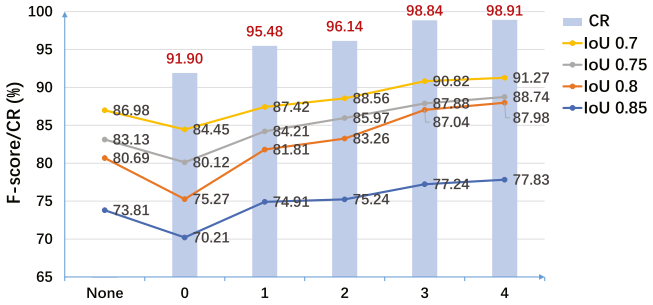


Fig. 8. Character recognizer *F*-score and CR (character recognition rate) of IoU values ranging from 0.7 to 0.85 with respect to *incremental weakly supervised learning iteration*. The iteration represents the times we perform judgment gate mechanism, 0 indicates that the judgment gate mechanism has not yet been executed while *None* means the character recognizer has not been applied.

Table 1
Character recognition performance with respect to judgment gate parameter α and β, and incremental weakly supervised learning iteration number.

α	0.95			0.975		
β	0.05	0.025	0.0125	0.025	0.0125	0.00625
1	95.55	95.48	95.21	95.21	95.21	95.19
2	96.14	96.14	96.21	96.24	96.26	96.11
3	98.71	98.84	98.77	98.81	98.80	98.69
4	98.88	98.91	98.89	98.90	98.86	98.84

Note that 1, 2, 3, and 4 represent incremental weakly supervised learning iteration number.

5.2. Experimental results

5.2.1. Incremental weakly supervised learning

In Fig. 8, the x-axis represents the iteration number of the incremental weakly supervised learning of the character recognizer while the y-axis denotes the *F*-score or character recognition rate (CR). *F*-score is the harmonic mean of precision and recall. As illustrated in Fig. 8, without the proposed incremental weakly supervised learning method, the character recognizer has a poor recognition accuracy of 91.90%. When we applied incremental weakly supervised learning, the performance of the character recognizer gradually improves as the weakly supervised learning iteration increases, demonstrating the effectiveness of our incremental weakly supervised learning mechanism. Furthermore, the performance of the overall system has the same trend of character recognition precision under all IoU settings, which reflects the significance and importance of the recognition confidence information. Note that when the performance of the character model is relatively low, i.e., 91.90% at 0th iteration, our segmentation performs even worse than the situation in which recognition confidence information is not introduced. This is because a low-performance character recognizer can easily assign a high confidence score to a wrongly

[2] https://github.com/HCIILAB/TKH_MTH_Datasets_Release.

Table 2
Segmentation results using different kinds of recognition information. (R, P, and F represent Recall, Precision, and *F*-score, respectively. *F*-score is the harmonic mean of precision and recall.)

Method	Runtime (ms/line)	IoU 0.7			IoU 0.75			IoU 0.8			IoU 0.85		
		R	P	F	R	P	F	R	P	F	R	P	F
baseline (BBS)	142 ± 36	88.24	85.76	86.98	84.21	82.07	83.13	81.21	80.18	80.69	74.21	73.41	73.81
Rg-BBS	2257 ± 204	**92.68**	89.42	91.02	**90.17**	86.93	88.52	**86.66**	82.97	84.77	78.76	75.39	77.04
Rg-ABBS w.o. text line annotation	538 ± 82	88.31	85.89	87.08	84.33	82.11	83.21	81.98	81.03	81.50	74.85	73.62	74.23
Rg-ABBS	626 ± 124	92.63	**90.56**	**91.58**	90.15	**88.13**	**89.13**	86.54	**84.61**	**85.56**	**78.80**	**77.04**	**77.91**

Table 3
Comparison with previous methods.

Method	IoU 0.7			IoU 0.75			IoU 0.8			IoU 0.85		
	R	P	F	R	P	F	R	P	F	R	P	F
Projection [19]	32.61	34.34	33.45	22.23	23.16	22.69	15.38	15.79	15.58	12.16	12.41	12.28
Grouping [30]	26.60	21.07	23.51	15.73	10.93	12.90	15.02	10.81	10.87	14.08	9.97	11.67
CCS [31]	75.51	76.42	75.96	70.63	71.48	71.05	62.21	62.96	62.58	45.37	45.92	45.64
SS [60]	27.41	21.07	23.83	24.18	20.87	22.40	20.35	16.73	18.36	15.21	11.43	13.05
ACF [42]	25.48	26.17	25.81	23.41	24.35	23.87	21.48	22.27	21.87	20.53	21.42	20.97
R-fcn [43]	88.54	97.32	92.72	83.85	92.17	87.81	71.15	78.22	74.52	47.57	52.29	49.82
SSD [37]	59.56	**98.54**	74.24	57.46	**95.19**	71.66	52.31	**86.60**	65.23	42.69	70.56	53.20
YOLOv2 [36]	**93.92**	97.11	**95.49**	**90.42**	93.50	**91.93**	81.32	84.09	82.68	60.02	62.06	61.02
TextBox [52]	56.46	98.51	71.78	53.72	91.02	67.56	48.58	84.77	61.77	42.29	79.26	55.15
MNC [34]	90.70	91.22	90.96	86.17	84.66	85.41	76.29	76.73	76.51	56.14	56.45	56.29
FCIS [33]	74.89	75.78	75.33	55.17	55.82	55.49	30.82	31.19	31.01	21.94	22.08	22.01
BBS (ours)	88.24	85.76	86.98	84.21	82.07	83.13	81.21	80.18	80.69	74.21	73.41	73.81
Rg-ABBS (ours)	92.63	90.56	91.58	90.15	88.13	89.13	**86.54**	84.61	**85.56**	**78.80**	**77.04**	**77.91**

segmented bounding box while assigning low confidence scores to correctly segmented bounding boxes. This counter-example can also validate the importance of an excellent recognizer and the weakly supervised learning strategy for our Rg-ABBS system.

Judgment gate. In our experiment, we set parameters μ_1 and μ_2 respectively to 1 and -1, with the parameter η as 0.25. As shown in Table 1, we investigate the parameter α and β under different settings with incremental weakly supervised learning over 4 iterations (each iteration consists of 30 epochs). It can be observed that as the judgment gate becomes stricter (higher α and smaller β), the final performance of the character recognizer rises in the beginning (from $\beta = 0.05$ to 0.025 when $\alpha = 0.95$) and then decreases afterward (from $\beta = 0.0125$, $\alpha = 0.95$ to $\beta = 0.00625$, $\alpha = 0.975$). This is because when the judgment gate becomes stricter, we can obtain more accurately labeled samples; thus, the performance of the trained character recognizer is better. However, when the constraint becomes extremely strong, e.g., $\beta = 0.00625$ with $\alpha = 0.975$, there are not enough samples for the confirmed data pool, especially for the rarely-used Chinese characters, and therefore the character recognizer will perform worse in this situation.

5.2.2. Recognition information evaluation

In this section, we report an evaluation of recognition information, including both character and text confidence scores, to estimate their roles in the proposed Rg-BBS system. Note that for the following experiments, we set hyper-parameter λ_1, λ_2, and λ_3 as 1, 1, and 10, respectively.

BBS. In Table 2, it is observed that the proposed BBS has the poorest performance but the fastest segmentation speed among all the listed methods. This is because BBS do not have to consult the character recognizer. Considering the advantage of BBS, it is applied to provide the coarse segmentation result for incremental weakly supervised learning and constitute the basic component Rg-ABBS.

Rg-BBS. As illustrated in Algorithm 2 and Eq (9), the proposed Rg-BBS is constructed based on BBS by introducing character recognition information and text line annotation. As shown in Table 2, Rg-BBS enjoys an absolute recall/precision/*F*-score improvement of 3–5% for IoU values ranging from 0.7 to 0.85. However, Rg-BBS consumes much more time, nearly 16 times slower, than that of BBS in the decoding process.

Rg-ABBS. To utilize both the fast decoding speed of BBS and the high-precision decoding result of Rg-BBS, we proposed Rg-ABBS which use BBS as the basic decoding strategy and Rg-BBS for the confusing parts (attention area). As shown in Table 2, Rg-ABBS not only substantially reduces the decoding time, but also keeps sufficiently high segmentation performance as compared to Rg-BBS.

Rg-ABBS w.o. text line annotation. To evaluate the effectiveness of text line annotation, we remove text line annotation information from Rg-ABBS, and denote this situation as *Rg-ABBS w.o. text line annotation*. As shown in Table 2, Rg-ABBS w.o. text line annotation only shows slightly better the segmentation result than BBS. This is because the character recognizer can easily assign a high recognition confidence score to a wrongly-merged bounding box [55], e.g., some character-like radicals, while sometimes assigning low recognition confidence scores to correct bounding boxes.

5.2.3. Comparison with previous methods

In Table 3, we compare our approach with some well-established existing methods, including traditional document segmentation methods such as Projection [19], Grouping [30], SS [60] and ACF [42], deep-learning-based instance segmentation methods such as MNC [34] and FCIS [33], and deep-learning-based detection methods such as R-fcn [43], SSD [37], YOLOv2 [36], and TextBoxes [52]. Note that for fair comparison, we use the bounding box of instance segmentation methods as their prediction in the experiments. As demonstrated in Table 3, all the traditional methods obtain poor results for character segmentation, even when the IoU threshold is 0.7. This is due to the complex and complicated layout and background of the Tripitaka document. When the IoU threshold is set as 0.7 or 0.75, detection-based and segmentation-based methods dominate the best results. However, with the increase of the IoU threshold, we can observe that segmentation results for all the methods are substantially reduced, and our methods exhibit superior results than all the others. Specifically, our method can still provide an *F*-score of 77.91% with the rigorous

requirement of an IoU threshold of 0.85, which reveals the fact that the proposed Rg-ABBS method is likely to provide extremely precise segmentation. As illustrated in Fig. 3, for historical image segmentation purposes, only predicted bounding boxes with IoU equal or greater than 0.8 are good enough, indicating that the adoption of the proposed Rg-ABBS method is advisable for the segmentation of historical documents, e.g., Tripitaka images.

Considering the impressive performance of YOLOv2 on IoU of 0.7 and 0.75, we also combine the segmentation results of YOLOv2 and the proposed Rg-ABBS method by first aligning their predicted bounding boxes, and then averaging the center positions, heights, and widths of the matched boxes (IoU $>$ 0.5), while leaving the unmatched boxes remain in the final result. The ensemble model strikes a trade-off between YOLOv2 and the proposed Rg-ABBS, with *F*-score of 94.36, 90.20, 83.21, and 65.71 on IoU of 0.7, 0.75, 0.8 and 0.85, respectively.

6. Conclusion

In this paper, we formulate the challenging problem of historical document image segmentation from a Bayesian decision theory perspective. Towards high-precision segmentation, we proposed three novel algorithms, including boundary box segmentation (BBS), recognition-guided BBS (Rg-BBS), and recognition-guided attention BBS (Rg-ABBS), progressively. Furthermore, we propose an incremental weakly supervised learning strategy with judgment gate (JG) mechanism for character recognizer training. Experiments show that the proposed incremental weakly supervised learning strategy can substantially improve the performance of the character recognizer as well as the final segmentation result. We also observe that the proposed Rg-ABBS successfully integrates the recognition information of character and line-level annotation to facilitate the segmentation result while consuming much less time and effort than Rg-ABBS. Compared with traditional and deep learning based methods, the proposed recognition-guided segmentation system exhibits superior performance for higher IoU thresholds, which is crucial for reliable historical image segmentation.

Acknowledgment

This research is supported in part by the National Key Research and Development Program of China (No. 2016YFB1001405), GD-NSF (no. 2017A030312006), NSFC (Grant Nos. 61673182 and 61771199), and GDSTP (Grant No. 2017A010101027).

References

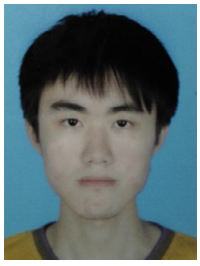
[1] T. Van Phan, B. Zhu, M. Nakagawa, Development of Nom character segmentation for collecting patterns from historical document pages, in: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, 2011, pp. 133–139.
[2] A. Antonacopoulos, R. Ritchings, Flexible page segmentation using the background, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2, 1994, pp. 339–344.
[3] L. O'Gorman, The document spectrum for page layout analysis, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1162–1173.
[4] K.Y. Wong, R.G. Casey, F.M. Wahl, Document analysis system, IBM J. Res. Dev. 26 (6) (1982) 647–656.
[5] K. Chen, M. Seuret, J. Hennebert, R. Ingold, Convolutional neural networks for page segmentation of historical document images, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 1, IEEE, 2017, pp. 965–970.
[6] C. Grana, D. Borghesani, R. Cucchiara, Automatic segmentation of digitalized historical manuscripts, Multimed. Tools Appl. 55 (3) (2011) 483–506.
[7] S.S. Bukhari, T.M. Breuel, A. Asi, J. El-Sana, Layout analysis for arabic historical document images using machine learning, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2012, pp. 639–644.
[8] C. Panichkriangkrai, L. Li, K. Hachimura, Character segmentation and retrieval for learning support system of Japanese historical books, in: Proceedings of the Second International Workshop on Historical Document Imaging and Processing, ACM, 2013, pp. 118–122.
[9] R. Cohen, A. Asi, K. Kedem, J. El-Sana, I. Dinstein, Robust text and drawing segmentation algorithm for historical documents, in: Proceedings of the Second International Workshop on Historical Document Imaging and Processing, ACM, 2013, pp. 110–117.
[10] A. Asi, R. Cohen, K. Kedem, J. El-Sana, I. Dinstein, A coarse-to-fine approach for layout analysis of ancient manuscripts, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2014, pp. 140–145.
[11] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.
[12] Y. Xu, W. He, F. Yin, C.-L. Liu, Page segmentation for historical handwritten documents using fully convolutional networks, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 1, IEEE, 2017, pp. 541–546.
[13] D. He, S. Cohen, B. Price, D. Kifer, C.L. Giles, Multi-scale multi-task FCN for semantic page segmentation and table detection, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 1, IEEE, 2017, pp. 254–261.
[14] T.M. Breuel, Robust, simple page segmentation using hybrid convolutional MDLSTM networks, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 1, IEEE, 2017, pp. 733–740.
[15] T. Grüning, R. Labahn, M. Diem, F. Kleber, S. Fiel, Read-bad: a new dataset and evaluation scheme for baseline detection in archival documents, in: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 2018, pp. 351–356.
[16] B. Moysset, C. Kermorvant, C. Wolf, J. Louradour, Paragraph text segmentation into lines with recurrent neural networks, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2015, pp. 456–460.
[17] Q.N. Vo, G. Lee, Dense prediction for text line segmentation in handwritten document images, in: Proceedings of the International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3264–3268.
[18] L. Likforman-Sulem, A. Zahour, B. Taconet, Text line segmentation of historical documents: a survey, Int. J. Doc. Anal. Recognit. (IJDAR) 9 (2–4) (2007) 123–138.
[19] M.W.A. Kesiman, J.-C. Burie, J.-M. Ogier, A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 325–330.
[20] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, T. Paquet, Handwritten text line segmentation using fully convolutional network, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 5, IEEE, 2017, pp. 5–9.
[21] R.G. Casey, E. Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 18 (7) (1996) 690–706.
[22] M. Schenkel, H. Weissman, I. Guyon, C. Nohl, D. Henderson, Recognition-based segmentation of on-line hand-printed words, in: Proceedings of the Advances in Neural Information Processing Systems, 1993, pp. 723–730.
[23] K. Daifallah, N. Zarka, H. Jamous, Recognition-based segmentation algorithm for on-line arabic handwriting, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2009, pp. 886–890.
[24] K.M. Sayre, Machine recognition of handwritten words: a project report, Pattern Recognit. 5 (3) (1973) 213–228.
[25] Q.-F. Wang, F. Yin, C.-L. Liu, Handwritten Chinese text recognition by integrating multiple contexts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1469–1481.
[26] Y.-C. Wu, F. Yin, C.-L. Liu, Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models, Pattern Recognit. 65 (2017) 251–264.
[27] U.-V. Marti, H. Bunke, On the influence of vocabulary size and language models in unconstrained handwritten text recognition, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2001, pp. 260–265.
[28] R. Manmatha, N. Srimal, Scale space technique for word segmentation in handwritten documents, Lect. Notes Comput. Sci. 1682 (1999) 22–33.
[29] J. He, A.C. Downton, User-assisted archive document image analysis for digital library construction, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2003, pp. 498–502.
[30] L. Likforman-Sulem, C. Faure, Extracting text lines in handwritten documents by perceptual grouping, in: Advances in Handwriting and Drawing: A Multidisciplinary Approach, 1994, pp. 117–135.
[31] V.P. Le, N. Nayef, M. Visani, J.-M. Ogier, C. De Tran, Text and non-text segmentation based on connected component features, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 1096–1100.
[32] A. Cheung, M. Bennamoun, N.W. Bergmann, An arabic optical character recognition system using recognition-based segmentation, Pattern Recognit. 34 (2) (2001) 215–233.
[33] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[34] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3150–3158.

[35] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[36] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 21–37.
[38] S. Suzuki, et al., Topological structural analysis of digitized binary images by border following, Comput. Vis. Graph. Image Process. 30 (1) (1985) 32–46.
[39] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[40] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1440–1448.
[41] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2980–2988.
[42] P. Dollár, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1532–1545.
[43] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 379–387.
[44] G. Navarro, A guided tour to approximate string matching, ACM Comput. Surv. (CSUR) 33 (1) (2001) 31–88.
[45] K. Kukich, Techniques for automatically correcting words in text, ACM Comput. Surv. (CSUR) 24 (4) (1992) 377–439.
[46] J. Kececioglu, D. Sankoff, Exact and approximation algorithms for the inversion distance between two chromosomes, in: Proceedings of the Annual Symposium on Combinatorial Pattern Matching, Springer, 1993, pp. 87–105.
[47] W.F. Tichy, The string-to-string correction problem with block moves, ACM Trans. Comput. Syst. (TOCS) 2 (4) (1984) 309–321.
[48] Z.-H. Zhou, A brief introduction to weakly supervised learning, Natl. Sci. Rev. 5 (1) (2017) 44–53.
[49] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[50] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. 11 (1999) 131–167.
[51] F. Muhlenbach, S. Lallich, D.A. Zighed, Identifying and handling mislabelled instances, J. Intell. Inf. Syst. 22 (1) (2004) 89–109.
[52] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, TextBoxes: a fast text detector with a single deep neural network, in: Proceedings of the AAAI, 2017.
[53] R.P. dos Santos, G.S. Clemente, T.I. Ren, G.D. Cavalcanti, Text line segmentation based on morphology and histogram projection, in: Proceedings of the Tenth International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 651–655.
[54] D.R. Reddy, et al., in: Speech Understanding Systems: A Summary of Results of the Five-Year Research Effort, Department of Computer Science, Camegie-Mell University, Pittsburgh, PA, 1977.
[55] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 427–436.
[56] M. He, S. Zhang, H. Mao, L. Jin, Recognition confidence analysis of handwritten Chinese character with CNN, in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 61–65.
[57] M. Li, I.K. Sethi, Confidence-based active learning, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1251–1261.
[58] Dazangjing, in: Tripitaka Koreana in Han, 2017 (http://kb.sutra.re.kr/ritk/index.do).
[59] H. Yang, L. Jin, W. Huang, Z. Yang, S. Lai, J. Sun, Dense and tight detection of Chinese characters in historical documents: datasets and a recognition guided detector, IEEE Access 6 (2018) 30174–30183.
[60] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

Zecheng Xie is a Ph.D. student in information and communication engineering at the South China University of Technology. He received a B.S. in electronics and information engineering from South China University of Technology in 2014. His research interests include machine learning, document analysis and recognition, computer vision, and human–computer interaction.

Yaoxiong Huang is a master student in communication and information system at the South China University of Technology. He received a B.S. in electronics and information engineering from South China University of Technology. His research interests include machine learning and computer vision.

Lianwen Jin received the B.S. degree from the University of Science and Technology of China, Anhui, China, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 1991 and 1996, respectively. He is a professor in the College of Electronic and Information Engineering at the South China University of Technology. His research interests include handwriting analysis and recognition, image processing, machine learning, and intelligent systems. He has authored over 100 scientific papers. He received the New Century Excellent Talent Program of MOE Award and the Guangdong Pearl River Distinguished Professor Award, in 2006 and 2011, respectively.



Yuliang Liu is currently a Ph.D. student at the Deep Learning and Vision Computing lab (DLVCLab), South China university of Technology, Guangdong, China. He received the B.S. degree in electronic and information engineering from South China University of Technology in 2016. He works on scene text understanding, handwritten character recognition, document analysis, deep learning-based text detection and recognition. He has published more than 10 papers on reputable international journals and conferences.

Yuanzhi Zhu is a master student in electronic and communication engineering at the South China University of Technology. He received a B.S. in electronics and information engineering from South China University of Technology. His research interests include machine learning, document analysis and recognition and computer vision.

Liangcai Gao received his B.S. degree from Shijiazhuang Railway Institute in 2002, M.S. degree from Beijing Jiaotong University in 2005 and Ph.D. degree from Peking University in 2010. He is currently an associate professor in the Institute of Computer Science and Technology at Peking University. His research interests include document recognition, information retrieval, digital library and intelligent systems. He has received the Science and Technology Nova Program of Beijing in 2015. He is a member of the IEEE Computer Society and ACM.

Xiaode Zhang is a Master student in the Institute of Computer Science and Technology at Peking University. He received a B.S. in the Department of Automation from Tsinghua University in 2016. His research interests include machine learning, document analysis and recognition, computer vision.