

PageNet: Towards End-to-End Weakly Supervised Page-Level Handwritten Chinese Text Recognition

Dezhi Peng¹ · Lianwen Jin^{1,4,5*} · Yuliang Liu² · Canjie Luo¹ · Songxuan Lai³

Received: date / Accepted: date

Abstract Handwritten Chinese text recognition (HCTR) has been an active research topic for decades. However, most previous studies solely focus on the recognition of cropped text line images, ignoring the error caused by text line detection in real-world applications. Although some approaches aimed at page-level text recognition have been proposed in recent years, they either are limited to simple layouts or require very detailed annotations including expensive line-level and even character-level bounding boxes. To this end, we propose PageNet for end-to-end weakly supervised page-level HCTR. PageNet detects and recognizes characters and predicts the reading order between them, which is more robust and flexible when dealing with complex layouts including multi-directional and curved text lines. Utilizing the proposed weakly supervised learning framework, PageNet requires only transcripts to be annotated for real data; however, it can still output detection and recognition results at both the character and line levels, avoiding the labor and cost of labeling bounding boxes of characters and text lines. Extensive experiments conducted on five datasets demonstrate the superiority of PageNet over existing weakly supervised and fully supervised page-level methods. These experimental results may spark further research beyond the realms of existing methods based on connectionist temporal classification or attention. The source code is available at <https://github.com/shannanyinxiang/PageNet>.

Keywords Handwritten Chinese Text Recognition · Page-Level Handwritten Text Recognition · Weakly Supervised Learning · Reading Order

¹South China University of Technology, Guangzhou, China

²Huazhong University of Science and Technology, Wuhan, China

³Huawei Cloud Computing Technologies, Shenzhen, China

⁴Pazhou Laboratory (Huangpu), Guangzhou, China

⁵Peng Cheng Laboratory, Shenzhen, China

*Corresponding author

1 Introduction

Handwritten Chinese text recognition (HCTR) has been studied for decades (Graves et al., 2009; Wang et al., 2012; Zhou et al., 2013; Keysers et al., 2017; Zhang et al., 2018). However, most previous studies (Yin et al., 2013; Wang et al., 2012, 2016; Wu et al., 2017; Peng et al., 2019; Su et al., 2009; Du et al., 2016; Wang et al., 2018, 2020a; Messina and Louradour, 2015; Wu et al., 2017; Xie et al., 2020; Xiu et al., 2019; Xie et al., 2019b; Wang et al., 2020b; Zhu et al., 2020; Luo et al., 2021; Rodriguez-Serrano et al., 2015; Jaderberg et al., 2016) assume that text line detection is provided by annotations and only focus on the recognition of cropped text line images. Although the accuracy of these line-level methods seems to be sufficient when combined with language models, they are limited to the one-dimensional distribution of characters and are significantly affected by the accuracy of text line detection in real-world applications. Therefore, handwritten text recognition at page level has important industrial value and has recently attracted remarkable research interest. One category of page-level methods (Huang et al., 2019; Chung and Delteil, 2019; Carbonell et al., 2019; Yang et al., 2018; Xie et al., 2019a; Ma et al., 2020; Moysset et al., 2017; Wigington et al., 2018; Tensmeyer and Wigington, 2019; Yang et al., 2018; Liu et al., 2021; Feng et al., 2021; Liu et al., 2020) segments text regions from the full page and recognizes the text regions, while the others (Yousef and Bishop, 2020; Bluche, 2016; Bluche et al., 2017) address page-level text recognition in a segmentation-free or implicit-segmentation fashion, utilizing connectionist temporal classification (CTC) (Graves et al., 2006) or attention mechanism combined with multi-dimensional long short-term memory.

However, existing page-level methods have several limitations. First, most of them (Huang et al., 2019; Chung

and Delteil, 2019; Carbonell et al., 2019; Ma et al., 2020; Yang et al., 2018; Moysset et al., 2017) cannot be trained in a weakly supervised manner, i.e., using only line-level or page-level transcripts. Extra annotations, such as bounding boxes of text lines or characters, are necessary, but it is costly to annotate them. The methods proposed by (Xing et al., 2019; Baek et al., 2019) can produce character bounding boxes without using corresponding annotations, but still require bounding box annotations of text lines. Some studies (Wigington et al., 2018; Tensmeyer and Wigington, 2019) allow a part of the training data to be annotated with only transcripts; however, expensive detection annotations are still required in the remaining training data. Although the method proposed by (Xie et al., 2019a) can be trained under weak supervision, it is limited to a specific layout and segments pages into text lines through vertical projection. Moreover, the methods proposed by (Yousef and Bishop, 2020; Bluche, 2016; Bluche et al., 2017) are trained with transcripts but cannot explicitly output bounding boxes of characters or text lines. Second, the reading order problem, which should be a very important issue for precisely understanding sentences, has rarely been discussed in previous literature. Retrospectively, most line-level and page-level methods output recognition results from left to right. There are also page-level methods (Chung and Delteil, 2019; Ma et al., 2020) that simply cluster detected words or characters into text lines based on specifically designed rules which cannot be generalized to other layouts. However, the reading order in the real world is significantly more complex, such as traditional Chinese texts that are read from top to bottom and curved text lines that are difficult to detect and recognize. Third, most previous approaches are not end-to-end trainable, which somewhat undermines accuracy and efficiency. Some studies (Chung and Delteil, 2019; Moysset et al., 2017) separately train two models to localize and recognize text lines; however, it may cause localization errors to propagate to the recognition part. The Start-Follow-Read model (Wigington et al., 2018) consists of three sequentially executed sub-networks, resulting in the inefficiency of the entire process. Finally, most previous methods are not designed for Chinese texts and thus do not perform well. Although a few approaches have been proposed for Chinese documents, they are limited to specific layouts (Yang et al., 2018; Xie et al., 2019a; Yang et al., 2018) or require detailed annotations (Ma et al., 2020).

To address the limitations mentioned above, we propose a novel method named PageNet for end-to-end weakly supervised page-level HCTR. PageNet performs page-level text recognition from a new perspective, i.e., detecting and recognizing characters and predicting the reading order between them. Three novel components are proposed for PageNet. The detection and recognition module detects and recognizes each character on a page. The reading

Table 1 Comparison of the required annotations versus the model output of existing page-level methods (L: line-level; W: word-level; C: character-level)

| Method | Annotations | | | Outputs | | |
|--------------------------|-------------|--------|------------|-----------|--------|------------|
| | Detection | | Transcript | Detection | | Transcript |
| | L | W or C | | L | W or C | |
| Bluche (2016) | | | ✓ | | | ✓ |
| Yousef and Bishop (2020) | | | ✓ | | | ✓ |
| Wigington et al. (2018) | | | ✓ | | ✓ | ✓ |
| Huang et al. (2019) | ✓ | | ✓ | ✓ | | ✓ |
| Ma et al. (2020) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ours | | | ✓ | ✓ | ✓ | ✓ |

order module determines the linking relationship between characters and whether a character is the start/end of a line. Finally, the graph-based decoding algorithm outputs detection and recognition results at both the character and line levels. Each component is seamlessly integrated into a unified network, which makes it end-to-end trainable with high efficiency. Generally, expensive bounding box annotations of characters and text lines are required to train such a network. To this end, a novel weakly supervised learning framework, consisting of matching, updating, and optimization, is proposed to make PageNet trainable under weak supervision. Bounding box annotations are no longer required, and only line-level transcripts need to be annotated for real data.

To the best of our knowledge, PageNet is the first method to solve page-level HCTR under weak supervision. Although no bounding box annotation is provided for real data, our model can still produce rich information that contains detection and recognition results at both the character and line levels. Therefore, our method can avoid the high cost of annotating the bounding boxes. Moreover, weakly annotated data is easy to obtain from the Internet, which makes data collection almost free. A comparison of the required annotations versus the model output is presented in Table 1. Compared with existing page-level methods, our method requires fewer annotations but outputs more information. To the best of our knowledge, PageNet is also the first method to solve the reading order problem in page-level HCTR. The reading order problem involves determining the order in which characters are read. By utilizing the proposed reading order module and graph-based decoding algorithm, our model can handle arbitrarily curved and multi-directional texts. In addition, although the model is designed for Chinese texts, it can also process multilingual texts including Chinese and English.

To verify the effectiveness of our method, extensive experiments are conducted on five datasets, namely CASIA-HWDB (Liu et al., 2011), ICDAR2013 (Yin et al., 2013), MTHv2 (Ma et al., 2020), SCUT-HCCDoc (Zhang et al., 2020), and JS-SCUT PrintCC. Because our model is weakly supervised, we further propose two evaluation metrics, termed accurate rate* (AR*) and correct rate* (CR*), for the situation in which only line-level transcripts are given.

The experimental results show that PageNet outperforms other weakly supervised page-level methods. Compared with fully supervised approaches, PageNet can also achieve competitive or better performance. Moreover, PageNet performs better than existing line-level methods for HCTR that directly recognize cropped text line images.

In summary, the main contributions of this paper are:

- We propose a novel method named PageNet for end-to-end weakly supervised page-level HCTR. PageNet solves page-level text recognition from a new perspective, namely, detecting and recognizing characters and predicting the reading order.
- A novel weakly supervised learning framework, consisting of matching, updating, and optimization, is proposed to make PageNet trainable with only line-level transcripts annotated for real data. Nevertheless, it can output detection and recognition results at both the character and line levels. Therefore, the cost of manual annotation can be significantly reduced.
- To the best of our knowledge, PageNet is the first method to address the reading order problem in page-level HCTR. The model can handle pages with multi-directional reading order and arbitrarily curved text lines.
- Extensive experiments on five benchmarks demonstrate the superiority of PageNet, indicating that it may be a remarkable step towards a new effective approach to the page-level HCTR problem.

2 Related Work

2.1 Line-level Handwritten Chinese Text Recognition

The methods for line-level HCTR aim to recognize text line images, which can be divided into two categories: segmentation-based and segmentation-free methods.

Segmentation-based methods address this problem based on oversegmentation or deep detection networks. The strategy using oversegmentation first obtains consecutive oversegments and then searches for the optimal segmentation-recognition path by integrating classifier outputs, geometric context, and linguistic context (Wang et al., 2012). Wang et al. (2016) improved the oversegmentation method using deep knowledge training and heterogeneous convolutional neural networks. Furthermore, based on oversegmentation methods, Wu et al. (2017) explored neural network language models and Wang et al. (2020b) proposed a weakly supervised learning method. However, it is difficult for these methods to recognize touching and overlapping characters. Therefore, with the prevalence of deep detection networks, Peng et al. (2019) proposed a segmentation and recognition module to detect and recognize characters in an end-to-end manner.

In addition, there are methods that solve line-level HCTR from a segmentation-free perspective. The methods proposed by (Su et al., 2009; Du et al., 2016; Wang et al., 2018) adopted systems based on hidden Markov model. Wang et al. (2020a) further introduced writer adaptation to this type of approach. Combining long short-term memory recurrent neural network (LSTM-RNN) and CTC (Graves et al., 2006) is another framework. Messina and Louradour (2015) used multi-dimensional LSTM-RNN to resolve line-level HCTR. Wu et al. (2017) proposed a separable multi-dimensional LSTM-RNN and achieved a significant improvement compared with previous LSTM-RNN-based methods. In addition to the methods focusing on the network architecture, Xie et al. (2020) explored data preprocessing and augmentation pipelines and achieved state-of-the-art results. The attention mechanism can also be used for line-level HCTR. Xiu et al. (2019) explored the attention-based decoder and proposed a multi-level multimodal fusion network to incorporate both the visual and linguistic semantic information.

Furthermore, to utilize both segmentation-based and segmentation-free methods, Zhu et al. (2020) proposed to combine these two kinds of approaches using a convolutional combination strategy.

In contrast to these line-level methods, the proposed PageNet model recognizes the text directly from the full page in an end-to-end fashion.

2.2 Page-level Handwritten Text Recognition

The goal of page-level handwritten text recognition is to recognize handwritten text from the full page. One category of methods detects text regions and then recognizes them. Chung and Delteil (2019) developed two separate components for text localization and recognition. Carbonell et al. (2019) proposed an end-to-end text detection and transcription framework wherein the two components are jointly trained. Huang et al. (2019) further improved the end-to-end framework using an adversarial feature enhancing network. Moysset et al. (2017) proposed to regress the left-side triplets rather than the coordinates of bounding boxes and determine the end of a line using a recognizer with an extra end-of-line label. Some methods for scene text spotting, such as Mask TextSpotter (Lyu et al., 2018; Liao et al., 2021) and FOTS (Liu et al., 2018), can also be applied to page-level handwritten text recognition. For Chinese text, Ma et al. (2020) presented a historical document processing system that simultaneously performs layout analysis, character detection, and character recognition. Yang et al. (2018) proposed a recognition-guided detector for tight Chinese character detection in historical documents.

However, detection annotations, such as bounding boxes

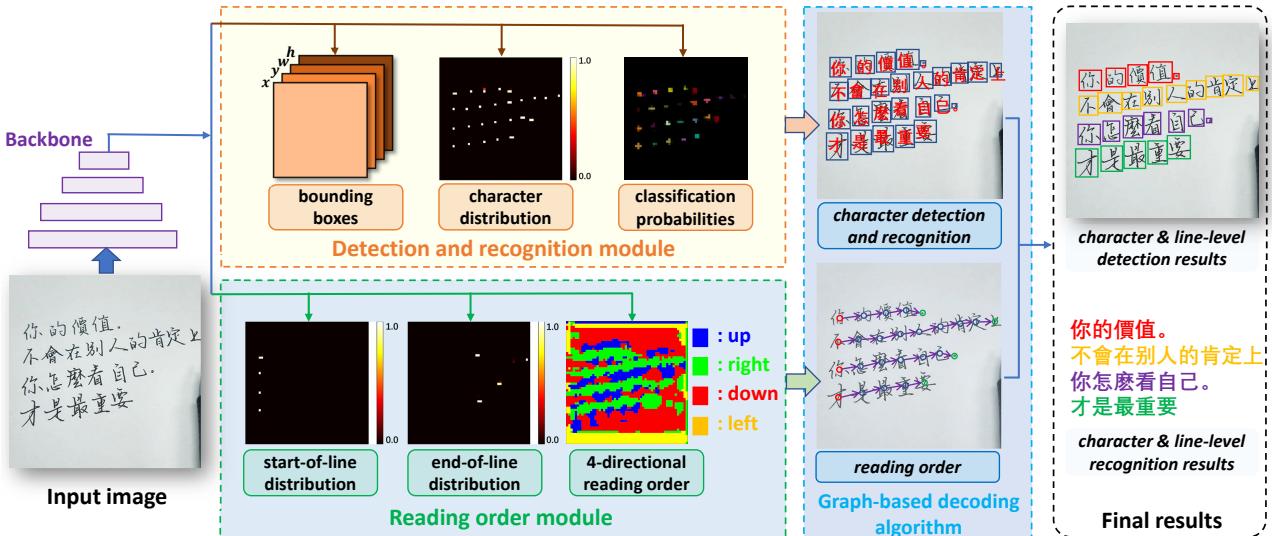


Fig. 1 Overall architecture of PageNet. Based on the features extracted by the backbone network, the detection and recognition module predicts the character detection and recognition results, while the reading order module predicts the reading order between characters. Combining these two predictions, the graph-based decoding algorithm outputs the final results containing detection and recognition results at both the character and line levels.

of text lines or characters, must be provided for the aforementioned methods. Therefore, some studies have focused on weakly supervised page-level handwritten text recognition that requires only transcripts for model training. Xie et al. (2019a) proposed a method for weakly supervised character detection in historical documents. However, this method is limited to a specific layout and cannot be generalized to unconstrained situations. Wigington et al. (2018) proposed the Start-Follow-Read model that requires only a small proportion of data to be fully annotated and the remaining data to be weakly annotated. Tensmeyer and Wigington (2019) further designed a novel alignment algorithm and enabled methods such as Start-Follow-Read to be trained using transcripts without line breaks. Combining multi-dimensional LSTM-RNN with the attention mechanism is another way to solve weakly supervised page-level handwritten text recognition. Following this idea, Bluche et al. (2017) and Bluche (2016) proposed methods for transcribing paragraphs. Furthermore, OrigamiNet (Yousef and Bishop, 2020) demonstrates that CTC can also be used for page-level text recognition by implicitly unfolding the 2-dimensional input signal to 1-dimensional.

Compared with existing methods, the proposed PageNet is trained without bounding box annotations for real data but outputs detection and recognition results at both the character and line levels. PageNet is also the first method to solve the reading order problem in page-level HCTR, which makes the model more robust and flexible.

3 Methodology

Most existing methods solve page-level text recognition following a top-down pipeline, i.e., text line detection and recognition. However, curved text lines have become a major challenge for such methods, and the reading order problem has rarely been investigated. Moreover, unlike other languages, Chinese characters are the basic elements that directly form sentences. Therefore, following a bottom-up pipeline, we propose PageNet for end-to-end weakly supervised page-level HCTR.

PageNet performs page-level text recognition from a new perspective, i.e., detecting and recognizing characters and predicting the reading order between them, which enables it to handle pages with multi-directional reading order and arbitrarily curved text lines. As shown in Fig. 1, PageNet consists of four parts: (1) the backbone network for feature extraction, (2) the detection and recognition module for character detection and recognition, (3) the reading order module for predicting the reading order between characters, and (4) the graph-based decoding algorithm that outputs the final results containing detection and recognition results at both the character and line levels. The detailed network architecture of PageNet is shown in Fig. 7. The components for character detection, character recognition, and reading order are integrated into a single network that is end-to-end optimized.

Manual annotations, including expensive line-level and character-level bounding boxes, are required by most previous methods. To this end, a novel weakly supervised learning framework (Fig. 5) is proposed to make PageNet trainable with only line-level transcripts annotated for real

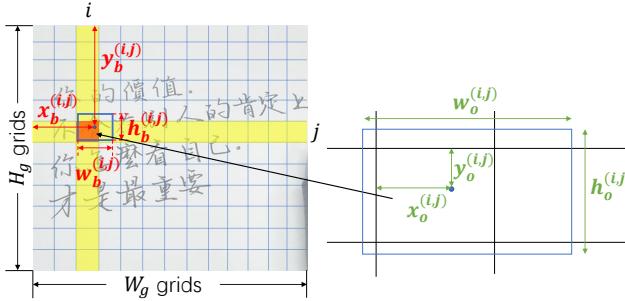


Fig. 2 Relationship between the output O_{box} of CharBox branch and the coordinates B_{box} of bounding boxes.

data, thereby avoiding the labor and cost of labeling bounding boxes of characters and text lines.

3.1 Backbone Network

Given an image with height H and width W , the backbone network extracts high-level feature maps of shape $\frac{W}{16} \times \frac{H}{16} \times 512$. In the following, we denote $\frac{W}{16}$ as W_g and $\frac{H}{16}$ as H_g for convenience.

3.2 Detection and Recognition Module

Following the successful decoupled three-branch design of our previous work (Peng et al., 2019), the detection and recognition module is proposed for character detection and recognition, which consists of character bounding box (CharBox), character distribution (CharDis), and character classification (CharCls) branches. We first apply $W_g \times H_g$ grids to the input image, as shown in the left part of Fig. 2, and denote the grid at the i -th column and j -th row as $G^{(i,j)}$. Then, the function of each branch is as follows:

CharBox Branch outputs O_{box} of shape $W_g \times H_g \times 4$. Fig. 2 and Eq. (1) show the relationship between $O_{box}^{(i,j)} = (x_o^{(i,j)}, y_o^{(i,j)}, w_o^{(i,j)}, h_o^{(i,j)})$ and the coordinate $B_{box}^{(i,j)} = (x_b^{(i,j)}, y_b^{(i,j)}, w_b^{(i,j)}, h_b^{(i,j)})$ of the bounding box for grid $G^{(i,j)}$.

$$\begin{aligned} x_b^{(i,j)} &= (i - 1 + x_o^{(i,j)}) / W_g \times W, \\ y_b^{(i,j)} &= (j - 1 + y_o^{(i,j)}) / H_g \times H, \\ w_b^{(i,j)} &= w_o^{(i,j)}, \\ h_b^{(i,j)} &= h_o^{(i,j)}. \end{aligned} \quad (1)$$

CharDis Branch produces character distribution O_{dis} of shape $W_g \times H_g$, where $O_{dis}^{(i,j)}$ is the confidence that grid $G^{(i,j)}$ contains characters.

CharCls Branch generates O_{cls} of shape $W_g \times H_g \times N_{cls}$, where $O_{cls}^{(i,j)}$ contains the classification probabilities of N_{cls} categories for grid $G^{(i,j)}$.

3.3 Reading Order Module

For a line-level recognizer, it is natural to arrange the recognized characters from left to right. However, the situation becomes significantly more complicated when characters can be arbitrarily distributed along two dimensions. The reading order problem has rarely been studied in previous literature, especially on the task of HCTR. However, this problem is important for building a flexible and robust page-level recognizer. Therefore, we propose the reading order module to solve this problem.

3.3.1 Problem Definition

Given an unordered set of characters, the reading order problem is to determine the order in which characters are read. We only investigate the reading order at line level rather than page level by rearranging the characters into multiple line-level transcripts. This means that the characters in one line-level transcript are sorted according to the reading order, but we do not consider the page-level reading order between different line-level transcripts. When a page contains simple layouts, such as only one paragraph, the page-level reading order can be easily determined using location information. However, when a page contains complex layouts, it is usually difficult to determine the page-level reading order. Different people may read the text lines in different orders. Moreover, most datasets only provide the transcript of each line separately.

3.3.2 Our Solution

Most previous methods simply solve the reading order problem by detecting text lines and recognizing them from left to right (Moysset et al., 2017; Huang et al., 2019; Lyu et al., 2018; Liu et al., 2018). However, these methods have difficulty handling multi-directional and curved text lines. To solve these issues, as illustrated in Fig. 3, we

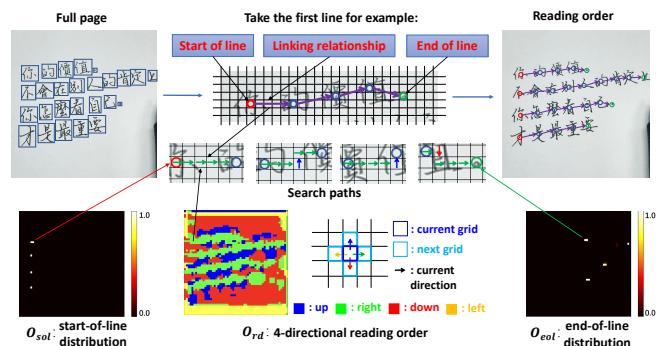


Fig. 3 Reading order problem is solved by making three predictions: (1) O_{sol} : start-of-line distribution, (2) O_{rd} : 4-directional reading order prediction, and (3) O_{eol} : end-of-line distribution.

decompose the reading order problem into three steps: (1) starting at the start-of-line, (2) finding the next character according to the linking relationship between characters, and (3) stopping at the end-of-line. The linking relationship is further decomposed into the movements from the current grid to its neighbor step by step. This pipeline makes it possible to deal with text lines with arbitrary directions and curves.

Specifically, we solve the reading order problem by making three predictions, namely the start-of-line distribution O_{sol} , the 4-directional reading order prediction O_{rd} , and the end-of-line distribution O_{eol} . The detailed network architecture of the reading order module is shown in Fig. 7(c). Both O_{sol} and O_{eol} are of shape $W_g \times H_g$, where $O_{sol}^{(i,j)}$ and $O_{eol}^{(i,j)}$ are the confidence that the character in grid $G^{(i,j)}$ is the start-of-line and the end-of-line, respectively. The 4-directional reading order prediction O_{rd} is of shape $W_g \times H_g \times 4$, where $O_{rd}^{(i,j)}$ are the probabilities of the four directions for grid $G^{(i,j)}$. The four predefined directions are up, right, down, and left, respectively. If the direction of grid $G^{(i,j)}$ is right, then the next grid is on its right, i.e., the next grid is grid $G^{(i+1,j)}$. The other three directions can be defined similarly. Thus, from a character, we can find the next character in the reading order by iteratively moving from a grid to the next according to the direction with maximum probability until arriving at a new character, as illustrated in the visualization of search paths in Fig. 3.

3.4 Graph-based Decoding Algorithm

Based on the predictions from the detection and recognition module and the reading order module, we propose a novel

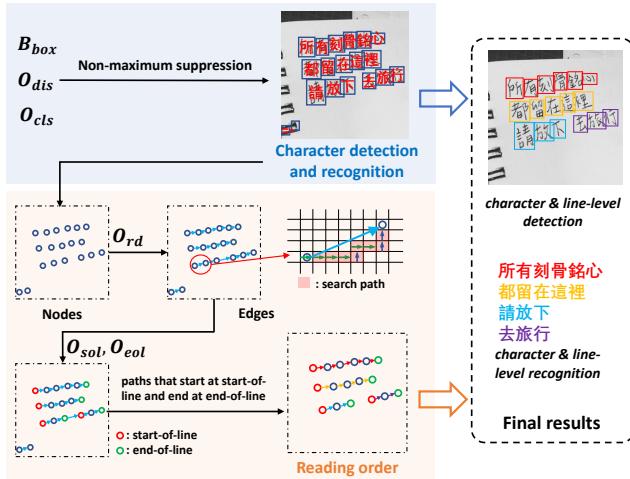


Fig. 4 Pipeline of the graph-based decoding algorithm. Based on the outputs from the detection and recognition module and the reading order module, the graph-based decoding algorithm produces the final results.

graph-based decoding algorithm to produce the final output containing detection and recognition results at both the character and line levels by viewing characters and reading order as a graph.

As shown in Fig. 4, the graph-based decoding algorithm consists of the following three steps: (1) the character detection and recognition results are derived from the outputs of the detection and recognition module, (2) the reading order is generated based on the outputs of the reading order module, and (3) the final results, which contain detection and recognition results at both the character and line levels, are obtained by combining the reading order and the character detection and recognition results.

3.4.1 Character Detection and Recognition

The detection and recognition module predicts the coordinates B_{box} of bounding boxes, the character distribution O_{dis} , and the classification probabilities O_{cls} . We use non-maximum suppression (NMS) (Neubeck and Van Gool, 2006) to remove redundant bounding boxes and obtain the character detection and recognition results, as shown in the blue part of Fig. 4. The character detection and recognition results contain multiple characters with their bounding boxes and categories.

3.4.2 Reading Order

The pipeline for generating the reading order is illustrated in the orange part of Fig. 4. The three steps are as follows.

Nodes. Each character detection and recognition result is viewed as a node. Therefore, each node corresponds to a grid in which the bounding box and category of the related character are predicted.

Edges. Based on the 4-directional reading order prediction O_{rd} , we find the next node of every node. Starting at the corresponding grid of one node, we move into the neighboring grid step by step according to the direction with the maximum probability in O_{rd} . If a grid with a corresponding node is reached, the next node is successfully found. However, if the search path exceeds the boundary of the grids or is stuck in a cycle, the next node does not exist.

Reading Order. We distinguish whether a node is the start-of-line or the end-of-line according to the start-of-line distribution O_{sol} and the end-of-line distribution O_{eol} . Then, the reading order is represented by the paths that start at the start-of-line and end at the end-of-line.

3.4.3 Final Results

In the reading order, each path represents a text line, and each node corresponds to a character. After reorganizing the

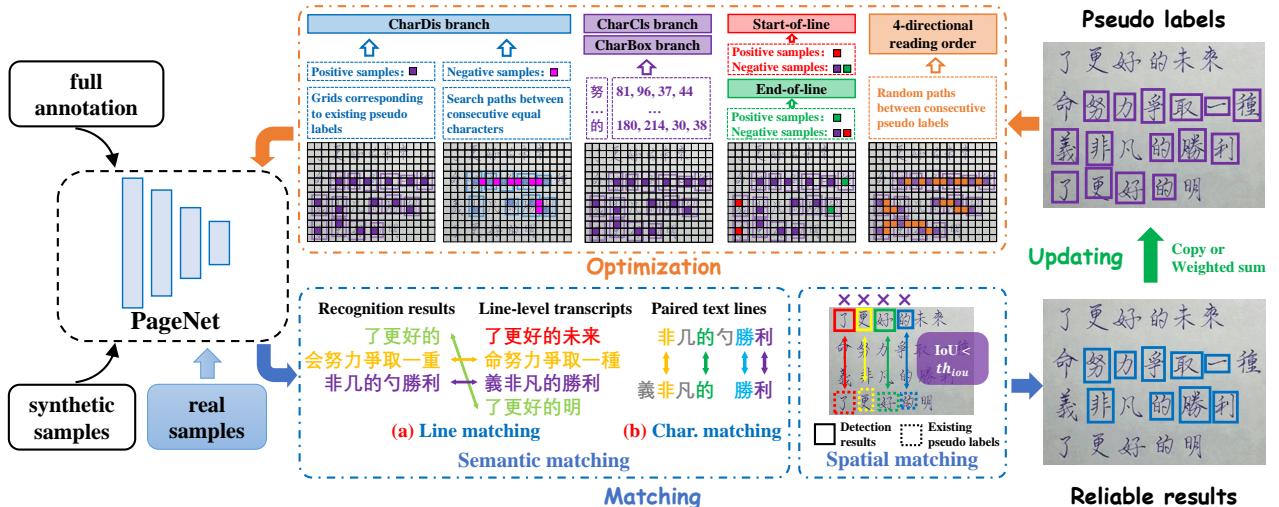


Fig. 5 Overall framework of weakly supervised learning.

character detection and recognition results according to the reading order, the final results are obtained as:

$$R = \{R^{(p)} | 1 \leq p \leq N_{ln}\}, \quad (2)$$

where N_{ln} is the number of text lines. $R^{(p)}$ contains the categories and bounding boxes of the characters in the p -th line, as shown in Eq. (3).

$$R^{(p)} = \{(c^{(p,m)}, x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)}) | 1 \leq m \leq N_{ch}^{(p)}\}, \quad (3)$$

where $N_{ch}^{(p)}$ is the number of characters in the p -th line. $c^{(p,m)}$ and $(x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)})$ are the category and bounding box of the m -th character in the p -th line, respectively, which are defined as:

$$c^{(p,m)} = \arg \max_{1 \leq c \leq N_{cls}} O_{cls}^{(\alpha^{(p,m)}, \beta^{(p,m)}, c)}, \quad (4)$$

$$(x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)}) = B_{box}^{(\alpha^{(p,m)}, \beta^{(p,m)})}, \quad (5)$$

where $(\alpha^{(p,m)}, \beta^{(p,m)})$ is the coordinate of the grid corresponding to the m -th node of the p -th path.

3.5 Weakly Supervised Learning

Normally, training PageNet requires full annotations, including the bounding boxes and categories of characters. However, the cost of annotating the bounding box and category of each character on a page is significantly higher than only annotating the transcript of each line. Moreover, line-level transcripts can be obtained almost without cost if the pages are from published books or historical documents. There is also a large amount of weakly annotated data on the Internet that has not been made full use of. Therefore, in this section, we present a weakly supervised learning framework, which consists of matching, updating, and

optimization, to make PageNet trainable with only line-level transcripts annotated for real data. The character-level and line-level bounding boxes not only no longer need to be labeled, but can even be automatically annotated through the proposed weakly supervised learning framework.

3.5.1 Overview

The overall framework of weakly supervised learning is shown in Fig. 5. The training data consists of synthetic and real samples. As described in Sec. 4.1, the synthetic samples have full annotations; therefore, the model can be normally optimized. However, the annotations of the real samples only contain line-level transcripts. Thus, three steps are designed for real samples. (1) *Matching*: match the results of PageNet with the line-level transcripts in the annotations to find reliable results. (2) *Updating*: Use the reliable results to update pseudo-labels. The pseudo-labels are the bounding boxes of the characters in the transcript annotations. (3) *Optimization*: Calculate the losses using the updated pseudo-labels to optimize the parameters. Because not all the characters have corresponding pseudo-labels, it is challenging to effectively train the model in such a situation.

3.5.2 Definition of Symbols

For convenience, the symbols are defined as follows.

- Given a real image, PageNet predicts the results R (Eq. (2)), where $c^{(p,m)}$ and $(x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)})$ are the category and bounding box of the m -th character in the p -th line, respectively, as specified in Sec. 3.4.3. Based on the predicted results, we further define the recognition result of the p -th line as $L^{(p)} = \{c^{(p,m)} | 1 \leq m \leq N_{ch}^{(p)}\}$.
- $B_{sco}^{(p,m)}$ is defined as the score of the bounding box of the m -th character in the p -th line .

Algorithm 1: Line Matching

Input: recognition results L , transcripts A
Output: line matches M_l

- 1 **for** $p = 1$ to N_{ln} and $q = 1$ to \hat{N}_{ln} **do**
- 2 $M_{AR}^{(p,q)} \leftarrow \text{AR between } L^{(p)} \text{ and } A^{(q)}$;
- 3 Sort M_{AR} from large to small, yielding the sorted indices
 $S_{AR} = \{(i^{(k)}, j^{(k)}) | M_{AR}^{(i^{(k+1)}, j^{(k+1)})} \leq M_{AR}^{(i^{(k)}, j^{(k)})}\}$;
- 4 **for** $k = 1$ to $N_{ln} \times \hat{N}_{ln}$ **do**
- 5 **if** $M_{AR}^{(i^{(k)}, j^{(k)})} \geq th_{AR}$ **then**
- 6 **if** both $L^{(i^{(k)})}$ and $A^{(j^{(k)})}$ are not matched **then**
- 7 add $(i^{(k)}, j^{(k)})$ to M_l ;

Algorithm 2: Character Matching

Input: line matches M_l , recognition results L , transcripts A
Output: character matches M_c , consecutive equals M_{ce}

- 1 **foreach** $(p, q) \in M_l$ **do**
- 2 compute edit distance between $L^{(p)}$ and $A^{(q)}$;
- 3 $\xi = \{\xi^{(1)}, \xi^{(2)}, \dots | \xi^{(j)} \in \{"E", "S", "I", "D"\}\}$ are the matching states of $L^{(p)}$;
- 4 **for** $j = 1$ to $|\xi|$ **do**
- 5 **if** $\xi^{(j)}$ is “E” **then**
- 6 $\xi^{(j)}$ corresponds to $c^{(p,m)} = c_{gt}^{(q,n)}$;
- 7 add (p, m, q, n) to M_c ;
- 8 **if** $\xi^{(j+1)}$ is “E” or $j = |\xi|$ **then**
- 9 add (p, m) to M_{ce} ;

- $P_{sch}^{(p,m)}$ is defined as the coordinates of the grids in the search path that starts from the m -th character of the p -th line.
- The line-level transcript annotations are denoted as $A = \{A^{(q)} | 1 \leq q \leq \hat{N}_{ln}\}$, where $A^{(q)} = \{c_{gt}^{(q,n)} | 1 \leq n \leq \hat{N}_{ch}^{(q)}\}$ is the transcript of the q -th line and \hat{N}_{ln} is the total number of lines. Moreover, $c_{gt}^{(q,n)}$ is the category of the n -th character in the q -th line, and $\hat{N}_{ch}^{(q)}$ is the number of characters in the q -th line.
- The pseudo-label of character $c_{gt}^{(q,n)}$ is the coordinate of its bounding box $A_{ps}^{(q,n)} = (x_{ps}^{(q,n)}, y_{ps}^{(q,n)}, w_{ps}^{(q,n)}, h_{ps}^{(q,n)})$. We further denote the score of $A_{ps}^{(q,n)}$ as $\gamma^{(q,n)}$.

3.5.3 Matching

Given the results R and the line-level transcripts A , the aim of matching is to find reliable character-level results and their corresponding ground-truth characters. Specifically, the matching algorithm consists of semantic matching and spatial matching.

Semantic Matching. In general, correctly recognized characters also have accurate bounding boxes. Based on this observation, semantic matching aims to identify correctly recognized characters in the results. As shown in Fig. 5, semantic matching is composed of two steps as follows.

(1) *Line matching*: We match the line-level transcripts A and the recognition results L by the accurate rate (AR) (Wang et al., 2012) using Algorithm 1. The algorithm first calculates the AR between every pair of line-level transcripts and recognition results, and then obtains matching pairs M_l in the descending order of all the calculated ARs, where the threshold th_{AR} is used to filter out poor recognition results. Specifically, $(p, q) \in M_l$ indicates that the recognition result $L^{(p)}$ is matched to the line-level transcript $A^{(q)}$.

(2) *Character matching*: The characters in each pair of lines are matched according to the edit distance using Algorithm 2, where “E”, “S”, “I”, and “D” denote “equal”, “substitution”, “insertion”, and “deletion”, respectively. The

algorithm outputs the character matches M_c and consecutive equals M_{ce} . Specifically, $(p, m, q, n) \in M_c$ indicates that the character $c^{(p,m)}$ in the results is matched to the character $c_{gt}^{(q,n)}$ in the annotations, and M_{ce} contains the indices of the character in the results where two consecutive states of computing edit distance are “equal.”

Spatial Matching. If several same or similar sentences occur in the line-level transcripts A , it is possible that one line in the recognition results L is matched to any one of these sentences. This type of matching ambiguity cannot be solved using semantic matching. Therefore, spatial matching is proposed to address this issue. If the character $c^{(p,m)}$ in the result R is matched to the character $c_{gt}^{(q,n)}$ in the annotations A , we calculate the intersection over union (IoU) between bounding boxes $(x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)})$ and $A_{ps}^{(q,n)}$. If the IoU is lower than the threshold th_{IoU} , the matching pair (p, m, q, n) is removed from the character matches M_c .

3.5.4 Updating

After matching, the pseudo-labels A_{ps} are updated using Algorithm 3, where a pseudo-label is either copied from the predicted bounding box of the matched character or updated as the weighted sum of the existing pseudo-label and the predicted bounding box of the matched character. Specifically, the weight λ is calculated based on the scores of the predicted bounding box and existing pseudo-label. Because the predicted bounding boxes or pseudo-labels with low scores are usually inaccurate and thus should have a much lower influence on the updated pseudo-labels, the exponential function and scale factor ε are used to enlarge the gap between $B_{sco}^{(p,m)}$ and $\gamma^{(q,n)}$ when calculating the weight λ .

Algorithm 3: Pseudo-label Update

Input: character matches M_c , predicted results R ,
pseudo-labels A_{ps} , scores of bounding boxes B_{sco} ,
scores of pseudo-labels γ

Output: updated pseudo-label A_{ps}

```

1 foreach  $(p, m, q, n) \in M_c$  do
2   if  $A_{ps}^{(q,n)}$  does not exist then
3      $A_{ps}^{(q,n)} = (x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)})$ ;
4      $\gamma^{(q,n)} = B_{sco}^{(p,m)}$ ;
5   else
6      $\lambda = e^{\varepsilon \times \gamma^{(q,n)}} / (e^{\varepsilon \times \gamma^{(q,n)}} + e^{\varepsilon \times B_{sco}^{(p,m)}})$ ;
7      $A_{ps}^{(q,n)} = \lambda \times A_{ps}^{(q,n)} + (1 - \lambda) \times (x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)})$ ;
8      $\gamma^{(q,n)} = \lambda \times \gamma^{(q,n)} + (1 - \lambda) \times B_{sco}^{(p,m)}$ ;

```

3.5.5 Optimization

Because the pseudo-labels A_{ps} may not contain the bounding box of every character in the line-level transcripts A , it becomes challenging to effectively optimize the network. In the following, we introduce how the losses are calculated for each part of the model when the pseudo-labels are incomplete.

First, we define S_c as the mapping relationship between the grids and existing pseudo-labels, which is given by

$$S_c = \{(i, j, q, n) | \exists A_{ps}^{(q,n)}, (\lceil \frac{x_{ps}^{(q,n)} \times W_g}{W} \rceil, \lceil \frac{y_{ps}^{(q,n)} \times H_g}{H} \rceil) = (i, j)\}, \quad (6)$$

where $(i, j, q, n) \in S_c$ means pseudo-label $A_{ps}^{(q,n)}$ exists and corresponds to grid $G^{(i,j)}$.

CharDis Branch. For the CharDis branch, it is easy to find positive samples (S_c) from the existing pseudo-labels. However, determining negative samples becomes a difficult problem if not all pseudo-labels exist. In Eq. (7), we view the grids in the search paths that begin at consecutive equal characters as negative samples. Because the characters at two ends of these search paths are matched as “equal” consecutively, there is no character in these grids. Therefore, the loss of the CharDis branch is calculated using Eq. (8).

$$S_d^n = \{(i, j) | \exists (p, m) \in M_{ce}, (i, j) \in P_{sch}^{(p,m)}\}, \quad (7)$$

$$L_{dis} = -\frac{1}{2|S_c|} \sum_{(i,j,q,n) \in S_c} \log(O_{dis}^{(i,j)}) - \frac{1}{2|S_d^n|} \sum_{(i,j) \in S_d^n} \log(1 - O_{dis}^{(i,j)}). \quad (8)$$

CharBox Branch. First, each existing pseudo-label $A_{ps}^{(q,n)}$ is transformed back to $O_{ps}^{(q,n)}$ using Eq. (1) inversely. The loss of the CharBox branch is then calculated as the mean square error between every $O_{ps}^{(q,n)}$ generated from the existing pseudo-labels and its corresponding output $O_{box}^{(i,j)}$ of the

CharBox branch:

$$L_{box} = \frac{1}{|S_c|} \sum_{(i,j,q,n) \in S_c} (O_{box}^{(i,j)} - O_{ps}^{(q,n)}) W_{box} (O_{box}^{(i,j)} - O_{ps}^{(q,n)})^T, \quad (9)$$

where W_{box} is a diagonal matrix. The elements on the diagonal are the weights $(\delta_x, \delta_y, \delta_w, \delta_h)$ that are set to $(1, 1, 0.1, 0.1)$.

CharCls Branch. For each existing pseudo label $A_{ps}^{(q,n)}$, we can obtain the character classification probabilities $O_{cls}^{(i,j)}$ at its corresponding grid $G^{(i,j)}$ (Eq. 6) and the ground-truth character category $c_{gt}^{(q,n)}$. Thus, the loss of the CharCls branch is calculated as the cross entropy loss between them:

$$L_{cls} = -\frac{1}{|S_c|} \sum_{(i,j,q,n) \in S_c} \log(O_{cls}^{(i,j, c_{gt}^{(q,n)})}). \quad (10)$$

Start-of-Line. For the start-of-line distribution O_{sol} , if the pseudo-label of the first character in a line exists, its corresponding grid is selected as a positive sample. The grids corresponding to other existing pseudo-labels are regarded as negative samples. Therefore, the loss is calculated as

$$S_s^p = \{(i, j) | \exists (i, j, q, n) \in S_c, n = 1\}, \quad (11)$$

$$S_s^n = \{(i, j) | \exists (q, n), (i, j, q, n) \in S_c\} - S_s^p, \quad (12)$$

$$L_{sol} = -\frac{1}{2|S_s^p|} \sum_{(i,j) \in S_s^p} \log(O_{sol}^{(i,j)}) - \frac{1}{2|S_s^n|} \sum_{(i,j) \in S_s^n} \log(1 - O_{sol}^{(i,j)}), \quad (13)$$

where the subtraction in Eq. (12) means removing the elements in S_s^p from $\{(i, j) | \exists (q, n), (i, j, q, n) \in S_c\}$.

End-of-Line. The loss L_{eol} of the end-of-line distribution O_{eol} can be obtained similarly to L_{sol} , by viewing the grids corresponding to the pseudo-labels of the last characters of lines as positive samples and those corresponding to other pseudo-labels as negative samples.

4-Directional Reading Order. For the 4-directional reading order predictions O_{rd} , we randomly generate paths S_{rd} between the grids corresponding to consecutive pseudo-labels using Algorithm 4, where $(i, j, d) \in S_{rd}$ indicates that grid $G^{(i,j)}$ with d direction is in the paths. Then the loss is given by

$$L_{rd} = -\frac{1}{|S_{rd}|} \sum_{(i,j,d) \in S_{rd}} \log(O_{rd}^{(i,j,d)}). \quad (14)$$

Total Loss. The total loss L_{total} is given by Eq. (15). The model parameters are optimized to minimize the loss.

$$L_{total} = L_{box} + L_{cls} + L_{dis} + L_{sol} + L_{rd}, \quad (15)$$

where the total loss is the simple sum of all the loss terms. Although the performance may be improved when the weighting factors for the loss terms are carefully tuned on the target dataset, we formulate our method in a more generic manner.

Algorithm 4: Paths Generating

```

Input: pseudo-labels  $A_{ps}$ 
Output: paths  $S_{rd}$ 
1  $D_{rd} = \{(0, -1), (1, 0), (0, 1), (-1, 0)\};$ 
2 for  $q = 1$  to  $\hat{N}_{ln}$  and  $n = 1$  to  $\hat{N}_{ch}^{(q)}$  do
3   if  $A_{ps}^{(q,n)}$  exists and  $A_{ps}^{(q,n+1)}$  exists then
4      $(i, j) = (\lceil \frac{x_{ps}^{(q,n)} \times W_g}{W} \rceil, \lceil \frac{y_{ps}^{(q,n)} \times H_g}{H} \rceil);$ 
5      $(s, t) = (\lceil \frac{x_{ps}^{(q,n+1)} \times W_g}{W} \rceil, \lceil \frac{y_{ps}^{(q,n+1)} \times H_g}{H} \rceil);$ 
6      $\zeta = |s - i| + |t - j|;$ 
7     initialize a  $\zeta \times 1$  vector  $\vec{x}$  with  $sgn(s - i)$ ;
8     initialize a  $\zeta \times 1$  vector  $\vec{y}$  with 0;
9     randomly set  $|t - j|$  elements in  $\vec{y}$  to  $sgn(t - j)$ , and
       set the elements at the same indices in  $\vec{x}$  to 0;
10    for  $k = 1$  to  $\zeta$  do
11       $d \leftarrow$  the index of  $(\vec{x}^{(k)}, \vec{y}^{(k)})$  in  $D_{rd}$ ;
12      add  $(i, j, d)$  to  $S_{rd}$ ;
13       $(i, j) = (i, j) + (\vec{x}^{(k)}, \vec{y}^{(k)});$ 

```

4 Experiments

4.1 Dataset

CASIA-HWDB (Liu et al., 2011) is a large-scale Chinese handwriting database. We use two offline databases, namely **CASIA-HWDB1.0-1.2** and **CASIA-HWDB2.0-2.2**. CASIA-HWDB1.0-1.2 contains 3,895,135 isolated character samples. CASIA-HWDB2.0-2.2 contains 5,091 pages.

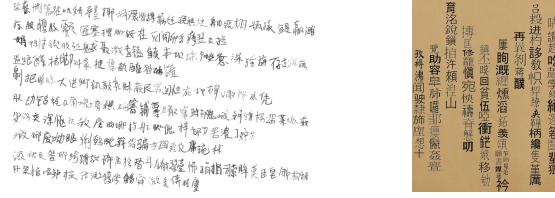
ICDAR2013 (Yin et al., 2013) includes a page-level dataset (**ICDAR13**) and a single character dataset (**ICDAR13-SC**). There are 300 pages in ICDAR13 and 224,419 character samples in ICDAR13-SC. The number of character categories is 7,356 when conducting experiments on **CASIA-HWDB** and ICDAR2013.

MTHv2 (Ma et al., 2020) contains 3,199 pages of historical documents, including 2,399 pages for training and 800 pages for testing. There are 6,762 categories of characters in MTHv2.

SCUT-HCCDoc ([Zhang et al., 2020](#)) contains 12,253 camera-captured documents with 6,109 categories of characters. The training and testing sets contain 9,801 images and 2,452 images, respectively.

JS-SCUT PrintCC is an in-house dataset that consists of 398 scanned images of printed documents. The images are divided into 348 for training and 50 for testing. There are 2,652 character classes in the dataset.

Synthetic Dataset. As shown in Fig. 6, we synthesize four datasets, namely CASIA-SR, MTH-SFB, HCCDoc-SFB, and JS-SF. “SR” and “SF” denote synthesizing using character samples from real isolated character databases and font files, respectively. The datasets whose names end with “B” are synthesized using background images rather

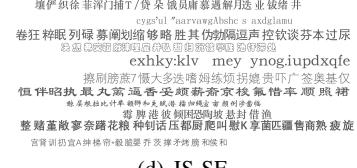


(a) CASIA-SR

(b) MTH-SFB



(c) HCCDoc-SFB



(d) JS-SF

Fig. 6 Example images from CASIA-SR, MTH-SFB, HCCDoc-SFB, and JS-SF.

than a white background. We adopt 101 font files and 32 background images that are downloaded from the Internet without using knowledge about real datasets. Specifically, the font files are randomly selected from the free fonts of the FounderType website¹. The background images are chosen from the pictures obtained by searching for “paper” on the Internet because our work is aimed at document recognition. For the CASIA-SR dataset, single character samples from CASIA-HWDB1.0-1.2 are used. All synthetic datasets have full annotations, i.e., line-level transcripts and bounding boxes of characters. Despite the different layouts of the real datasets, all four synthetic datasets follow a simple synthesis procedure. First, we synthesize text lines with randomly selected characters and obliquities. Note that no corpus is used when synthesizing these text lines. Afterwards, multiple text lines are combined to form a page. There is also no perspective transformation or illumination applied to the synthetic image.

4.2 Training Strategy

Directly training the randomly initialized PageNet using the proposed weakly supervised learning framework will lead to unsatisfactory performance. This is because of the low accuracy of the model and the lack of pseudo-labels during early iterations. Therefore, an improved training strategy is proposed, which consists of pretraining, initializing, and training stages. First, in the pretraining stage, the model is pretrained using synthetic samples. Then, in the initializing stage, the procedure is the same as Fig. 5 but without the optimization, which means that the pseudo-labels are not used to train the model. However, a part of the pseudo-labels is initialized and updated during the initializing stage.

¹ <http://www.foundertype.com/>

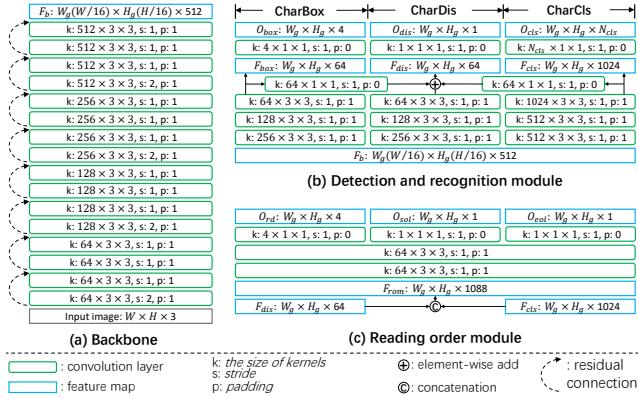


Fig. 7 Detailed network architecture.

Finally, the training stage is exactly the procedure shown in Fig. 5.

4.3 Implementation Details

4.3.1 Network Architecture

The detailed network architecture is illustrated in Fig. 7. The architecture of the backbone follows the previous work (Peng et al., 2019), which is verified to be effective for HCTR. It can also be easily changed to a standard backbone, such as ResNet (He et al., 2016). In the detection and recognition module, the interaction between the three branches is the same as that in the segmentation and recognition module (Peng et al., 2019). First, the feature F_{box} from the CharBox branch and the feature F_{cls} from the CharCls branch each go through a 1×1 convolution layer. Then, the two output features and the feature from the last convolution layer of the CharDis branch are element-wise added, yielding the feature F_{dis} .

4.3.2 Graph-based Decoding Algorithm

Score of Bounding Boxes. In Sec. 3.4.1, NMS is used to remove redundant bounding boxes. The confidence in the character distribution O_{dis} can be used as the score of bounding boxes. However, following the segmentation and recognition module (Peng et al., 2019), semantic information is integrated into the score of bounding boxes. Specifically, the score is the weighted sum of the character distribution confidence and maximum classification probability. The weight of the character distribution confidence is set to 0.8 following Peng et al. (2019).

Edges. As shown in the search path of Fig. 4, the next node should be at the next grid of the final grid in the search path. However, this is too strict and the 4-directional reading order prediction O_{rd} must be very accurate. Therefore, the next node is only required to be in the 4-neighborhoods of

the final grid in the search path. Furthermore, we limit the maximum number of steps in a search path for acceleration.

Start-of-Line and End-of-Line. In Sec. 3.4.2, a node is identified as the start-of-line or the end-of-line if the corresponding confidence in O_{sol} or O_{eol} is greater than 0.9.

Special Property of Graph. In page-level documents, a character has at most one previous character and one next character. Therefore, for a node in the graph, we must ensure that there is at most one edge in and one edge out. If there are multiple nodes in the 4-neighborhoods of the final grid in the search path and the direction of the final grid does not point to any one of them, the node whose corresponding bounding box has the highest score is selected. If there are multiple edges ending at the same node, the edge whose slope is closest to the slopes of the previous edges in the path is maintained.

4.3.3 Weakly Supervised Learning

The threshold th_{AR} in Algorithm 1, th_{IoU} in spatial matching, and the scale factor ε in Algorithm 3 are set to 0.3, 0.5 and 10, respectively.

4.3.4 Experiment Settings

We implement our method with PyTorch and conduct experiments using an NVIDIA RTX 2080ti GPU with 11GB of memory. Stochastic gradient descent with a batch size of 1 is used to optimize the network. Both the pretraining and training stages contain 300,000 iterations, and the learning rate is initialized to 0.01 and multiplied by 0.1 after 100,000, 200,000, and 275,000 iterations. The initializing stage contains 75,000 iterations and the learning rate is set to 0.0001. During the initializing and training stages, the probabilities of loading real samples and synthetic samples are 0.7 and 0.3, respectively. In the training stage, following existing methods (Xie et al., 2020; Baek et al., 2019), we use synthetic samples in addition to real samples to increase the diversity of training data and improve the stability of training. No validation set is adopted. All the training and testing images are resized to normalize their widths while maintaining their aspect ratios. The pixel value of the input image is normalized to the range of [0, 1]. Gaussian noise with a mean of 0 and a variance of 0.01 is applied to the synthetic images. Other settings for specific experiments are listed as follows, where the image width is estimated based on the number of characters on a page. Because Chinese characters are composed of complicated strokes, we should ensure that the characters are recognizable with the given input size, as well as consider training efficiency.

ICDAR13. The model is trained using 5,091 real samples from CASIA-HWDB2.0-2.2 and 20,000 synthetic samples from CASIA-SR. The testing is conducted on 300 samples

from ICDAR13. The width of the input image is normalized to 1,920 pixels.

MTHv2. We use the train set of MTHv2 and 10,000 samples from MTH-SFB to train the model and test it on the test set of MTHv2. The width of the input image is normalized to 2,960 pixels.

SCUT-HCCDoc. The model is trained using the training images from SCUT-HCCDoc and 20,000 synthetic images from HCCDoc-SFB. The test set of SCUT-HCCDoc is used to evaluate the model. The width of the input image is normalized to 1,600 pixels. Note that we adopt $4 \times$ iterations in the training stage of this experiment owing to the more complex scenarios and larger scale of SCUT-HCCDoc.

JS-SCUT PrintCC. The model is trained using the training samples from JS-SCUT PrintCC and 10,000 synthetic samples from JS-SF. The trained model is evaluated on the test samples of JS-SCUT PrintCC. The width of the input image is normalized to 2,080 pixels.

4.4 Evaluation Metrics

Our method requires only line-level transcripts to be annotated for real data. However, there is no metric to evaluate the performance when only line-level transcripts are annotated. To this end, we propose two evaluation metrics termed accurate rate* (AR*) and correct rate* (CR*). First, a matching algorithm, which is the same as Algorithm 1 but without filtering out poor recognition results at line 5, is executed, yielding line matches M_l^* . Moreover, we define S_R^* and S_A^* as the indices of unpaired lines in the results and annotations, respectively. Then AR* and CR* are given by

$$N_{Ie}^* = \sum_{(p,q) \in M_l^*} IE(L^{(p)}, A^{(q)}) + \sum_{i \in S_R^*} N_{ch}^{(i)}, \quad (16)$$

$$N_{De}^* = \sum_{(p,q) \in M_l^*} DE(L^{(p)}, A^{(q)}) + \sum_{i \in S_A^*} \hat{N}_{ch}^{(i)}, \quad (17)$$

$$N_{Se}^* = \sum_{(p,q) \in M_l^*} SE(L^{(p)}, A^{(q)}), \quad (18)$$

$$N_{total}^* = \sum_i \hat{N}_{ch}^{(i)}, \quad (19)$$

$$AR^* = (N_{total}^* - N_{Ie}^* - N_{De}^* - N_{Se}^*) / N_{total}^*, \quad (20)$$

$$CR^* = (N_{total}^* - N_{De}^* - N_{Se}^*) / N_{total}^*, \quad (21)$$

where the functions IE , DE , and SE compute the number of insertion, deletion, and substitution errors between the two input sequences, respectively. The errors between every matching pair are accumulated, and all the characters of the unpaired lines in the results and annotations are viewed as insertion and deletion errors, respectively. Compared with the vanilla accurate rate (AR) and correct rate (CR) (Wang et al., 2012) for line-level text recognition which indicates only text recognition performance, the proposed AR* and CR* for page-level text recognition considers both text detection and recognition.

4.5 Line-level Detection and Recognition

4.5.1 Performance on ICDAR13 Dataset

In Table 2, we compare the line-level detection and recognition results of our approach with existing page-level methods on the ICDAR13 dataset. The page-level methods include fully supervised approaches such as Det + Recog, Mask TextSpotter (Lyu et al., 2018; Liao et al., 2021), and FOTS (Liu et al., 2018), as well as weakly supervised approaches such as Start-Follow-Read (Wigington et al., 2018) and OrigamiNet (Yousef and Bishop, 2020). The method denoted as Det + Recog is the combination of two independently trained models which are a Mask R-CNN (He et al., 2017) (for text line detection) and a recognizer (Xie et al., 2020) (for text line recognition). The recognizer (Xie et al., 2020) achieves state-of-the-art performance on the text line recognition task of ICDAR13, as shown in Table 4. The fully supervised methods are trained using CASIA-SR and fully annotated CASIA-HWDB2.0-2.2, whereas the weakly supervised methods are trained using CASIA-SR and weakly annotated CASIA-HWDB2.0-2.2.

In addition to the proposed AR* and CR*, other evaluation metrics adopted in Table 2 are as follows. (1) Because there is only one paragraph on each page of ICDAR13, page-level transcript annotations and recognition results can be easily obtained. Thus, we calculate the page-level AR and CR in Table 2. (2) The normalized edit distance (NED) is calculated following the evaluation protocol of task 4 in (Zhang et al., 2019), which considers both text line detection and recognition. Because the results of our method and the annotations of ICDAR13 only provide the bounding boxes of characters, the bounding box of a text line is calculated as the rotated rectangle with the minimum area enclosing the characters of this text line. (3) Precision, recall, and f-measure are used to evaluate the performance of text line detection with an IoU threshold of 0.5.

As shown in Table 2, compared with existing page-level methods including three fully supervised methods, the proposed PageNet with weak supervision achieves state-of-the-art performance in terms of both end-to-end recognition and text line detection. For Det + Recog, although text line detection seems to be accurate in terms of f-measure, it is common that the bounding box of one text line contains the noise from other text lines and does not entirely cover the characters at both ends, which affects the accuracy of recognition. For the two end-to-end methods, namely Mask TextSpotter and FOTS, the large number of categories and the diversity of writing styles of Chinese texts make recognition a heavy burden for model optimization and their feature sharing mechanism. OrigamiNet performs page-level text recognition by unfolding 2-dimensional features to 1-dimensional. However, in contrast to English texts,

Table 2 Comparison with existing page-level methods on ICDAR13

| Supervision | Method | End-to-End Recognition | | | | | Text Line Detection | | |
|-------------|---|------------------------|--------------|--------------|--------------|--------------|---------------------|--------------|--------------|
| | | AR* | CR* | AR | CR | NED | Precision | Recall | F-measure |
| Full | Det + Recog (He et al., 2017; Xie et al., 2020) | 88.36 | 89.09 | 88.39 | 89.08 | 88.27 | 99.54 | 99.88 | 99.71 |
| | Mask TextSpotter (Lyu et al., 2018) | 49.48 | 57.95 | 50.60 | 58.29 | 50.40 | 91.21 | 96.77 | 93.91 |
| | FOTS (Liu et al., 2018) | 67.20 | 67.75 | 67.32 | 67.82 | 65.89 | 98.64 | 97.29 | 97.96 |
| Weak | Start-Follow-Read (Wigington et al., 2018) | 82.60 | 83.42 | 82.91 | 83.55 | 82.35 | 99.88 | 98.89 | 99.39 |
| | OrigamiNet (Yousef and Bishop, 2020) | - | - | 5.99 | 5.99 | - | - | - | - |
| | PageNet (Ours) | 92.83 | 93.23 | 92.86 | 93.24 | 92.49 | 99.56 | 99.94 | 99.75 |

Table 3 Comparison with existing page-level methods on MTHv2, SCUT-HCCDoc, and JS-SCUT PrintCC datasets

| Supervision | Method | MTHv2 | | SCUT-HCCDoc | | JS-SCUT PrintCC | |
|-------------|---|--------------|--------------|--------------|--------------|-----------------|--------------|
| | | AR* | CR* | AR* | CR* | AR* | CR* |
| Full | Det + Recog (He et al., 2017; Shi et al., 2017) | 94.50 | 95.29 | 83.44 | 87.97 | 94.68 | 95.03 |
| | FOTS (Liu et al., 2018) | 87.97 | 89.25 | 66.61 | 70.01 | 94.19 | 94.31 |
| Weak | Start-Follow-Read (Wigington et al., 2018) | 69.54 | 73.11 | 56.29 | 61.26 | 81.36 | 82.47 |
| | OrigamiNet (Yousef and Bishop, 2020) | 9.72 | 9.83 | 30.55 | 30.97 | 44.09 | 45.72 |
| | PageNet (Ours) | 93.76 | 95.23 | 77.95 | 82.15 | 97.25 | 98.19 |

each Chinese character itself is a complex 2-dimensional structure, which may make this mechanism difficult to work.

4.5.2 Performance on Other Datasets

The quantitative results of our method and existing page-level methods on MTHv2, SCUT-HCCDoc, and JS-SCUT PrintCC are listed in Table 3, where Det + Recog is the combination of a Mask R-CNN (He et al., 2017) (for text line detection) and a CTC-based recognizer (Shi et al., 2017) (for text line recognition). The fully supervised methods are trained using both synthetic data and real fully annotated data, whereas the weakly supervised methods are trained using both synthetic data and real weakly annotated data.

The proposed metrics AR* and CR* are reported in Table 3, as they are verified to be effective by Table 2 and require fewer annotations compared with other metrics. In particular, for the JS-SCUT PrintCC dataset that only provides line-level transcript annotations, only AR* and CR* can be calculated. Other metrics, such as page-level AR and CR, are not applicable to these three datasets because the annotations do not provide the reading order between text lines. However, because OrigamiNet can only be trained with page-level transcripts, we concatenate the line-level transcripts in the annotations based on the spatial location and obtain fake page-level transcripts. Therefore, the results of OrigamiNet are actually page-level AR and CR.

Compared with the fully supervised methods, our method can achieve competitive performance. Specifically, compared with Det + Recog, our method achieves lower accuracy on MTHv2 and SCUT-HCCDoc but performs better on JS-SCUT PrintCC. This is because MTHv2 and SCUT-HCCDoc contain significantly more complex layouts than JS-SCUT PrintCC, which is a big challenge for

the weakly supervised learning. In contrast, the text line detection part of Det + Recog can be trained significantly better under full supervision.

Compared with the weakly supervised methods, our method achieves the best performance. For Start-Follow-Read, the complex layouts lead to the failure of its line follower and end-of-line determination. However, our method can still maintain a relatively high accuracy owing to the bottom-up design and the effectiveness of the reading order module and the graph-based decoding algorithm.

Unlike other datasets, JS-SCUT PrintCC has three unique characteristics: (1) it contains printed documents, (2) 30% of the text lines are totally in English, and (3) the training set of real data is much smaller than other datasets. Therefore, the best result achieved on JS-SCUT PrintCC verifies the capability of our method on printed documents, multilingual texts, and few training samples.

4.5.3 Comparison with Line-level Methods

In Table 4, we compare our method with existing line-level methods on ICDAR13 which directly recognize the text line images cropped from the full pages based on the detection annotations. Although more stringent AR* and CR* take both text line detection and recognition into consideration, our method still achieves the best performance without language model compared with the results reported in previous literature.

4.5.4 Incorporation with Language Models

The proposed graph-based decoding algorithm can also use n-gram language models to improve the recognition performance. For the p -th path in the reading order, the grids corresponding to the nodes and the grids in the search paths

Table 4 Comparison with existing line-level methods on ICDAR13 (LM: language model)

| Method | Without LM | | With LM | |
|------------------------------|--------------|--------------|--------------|--------------|
| | AR | CR | AR | CR |
| Yin et al. (2013) | - | - | 86.73 | 88.76 |
| Messina and Louradour (2015) | 83.50 | - | 89.40 | - |
| Wu et al. (2017) | 86.64 | 87.43 | 90.38 | - |
| Du et al. (2016) | 83.89 | - | 93.50 | - |
| Wang et al. (2016) | 88.79 | 90.67 | 94.02 | 95.53 |
| Wu et al. (2017) | - | - | 96.20 | 96.32 |
| Wang et al. (2018) | 89.66 | - | 96.47 | - |
| Wang et al. (2020a) | 91.58 | - | 96.83 | - |
| Peng et al. (2019) | 89.61 | 90.52 | 94.88 | 95.51 |
| Xiu et al. (2019) | 88.74 | - | 96.35 | - |
| Xie et al. (2020) | 91.55 | 92.13 | 96.72 | 96.99 |
| Wang et al. (2020b) | 87.00 | 89.12 | 95.11 | 95.73 |
| Zhu et al. (2020) | 90.86 | - | 94.00 | - |
| PageNet (Ours) | AR* | CR* | AR* | CR* |
| PageNet (Ours) | 92.83 | 93.23 | 96.24 | 96.66 |

are concatenated as

$$\begin{aligned} grids^{(p)} = & \{(\alpha^{(p,1)}, \beta^{(p,1)})\} \oplus P_{sch}^{(p,1)} \oplus \dots \\ & \oplus \{(\alpha^{(p,N_{ch}^{(p)})}, \beta^{(p,N_{ch}^{(p)})})\} \oplus P_{sch}^{(p,N_{ch}^{(p)})}, \end{aligned} \quad (22)$$

where the elements in $grids^{(p)}$ are the coordinates of the grids. Specifically, as defined in Sec. 3.4.3 and 3.5.2, $N_{ch}^{(p)}$ is the number of characters in the p -th line, $(\alpha^{(p,m)}, \beta^{(p,m)})$ is the coordinate of the grid corresponding to the m -th character of the p -th line, and $P_{sch}^{(p,m)}$ contains the coordinates of the grids in the search path starting from the m -th character of the p -th line. Every element of $grids^{(p)}$ can be viewed as a time step similar to the decoding process of line-level recognizers. For each $(i, j) \in grids^{(p)}$, the blank probability is $1 - O_{dis}^{(i,j)}$ and the classification probabilities of N_{cls} categories are $O_{cls}^{(i,j)}$. Then, we use a trigram language model generated from the same corpus as (Xie et al., 2018) and a decoding algorithm proposed by (Graves and Jaitly, 2014) to obtain the recognition result of the p -th line. As shown in Table 4, the language model significantly improves the recognition performance, boosting AR* from 92.83% to 96.24%.

Table 5 Comparison of character-level detection and recognition result on ICDAR13

| Supervision | Method | DetOnly | | | 7356C | | |
|-------------|---|--------------|--------------|--------------|--------------|--------------|--------------|
| | | Precision | Recall | F-measure | Precision | Recall | F-measure |
| Full | Faster R-CNN (DetOnly) (Ren et al., 2017) | 98.93 | 92.12 | 95.41 | - | - | - |
| | Faster R-CNN (7356C) (Ren et al., 2017) | 95.61 | 89.83 | 92.63 | 88.85 | 83.48 | 86.08 |
| | YOLOv3 (DetOnly) (Redmon and Farhadi, 2018) | 93.94 | 98.25 | 96.05 | - | - | - |
| | YOLOv3 (7356C) (Redmon and Farhadi, 2018) | 89.56 | 92.16 | 90.84 | 66.32 | 68.24 | 67.26 |
| Weak | PageNet (Ours) | 95.72 | 94.91 | 95.31 | 90.89 | 90.12 | 90.50 |

4.6 Character-level Detection and Recognition

Because the annotations of ICDAR13 and CASIA-HWDB2.0-2.2 contain the bounding boxes of characters, we can compare the character-level detection and recognition results of our approach with two representative object detection methods, namely Faster R-CNN (Ren et al., 2017) and YOLOv3 (Redmon and Farhadi, 2018). These two methods are trained using CASIA-SR and fully annotated CASIA-HWDB2.0-2.2, whereas PageNet is trained using CASIA-SR and weakly annotated CASIA-HWDB2.0-2.2.

In Table 5, there are two versions of Faster R-CNN and YOLOv3. The one marked with DetOnly is trained to only detect characters, whereas the other marked with 7356C needs to classify 7,356 categories of characters in addition. There are also two sets of evaluation metrics. The one denoted as DetOnly evaluates the character detection regardless of the classification, whereas the other denoted as 7356C requires not only the detection is accurate but also the classification is correct. All evaluation metrics are calculated with an IoU threshold of 0.5.

As shown in Table 5, PageNet achieves better DetOnly and 7356C performances than Faster R-CNN (7356C) and YOLOv3 (7356C). Even compared with Faster R-CNN (DetOnly) and YOLOv3 (DetOnly), PageNet can still achieve comparable DetOnly performance. Note that Faster R-CNN and YOLOv3 are trained with full annotations (containing character bounding boxes), whereas our method is trained under weak supervision (without bounding box annotations). It can be concluded that the decoupled three-branch design of the detection and recognition module can handle the task of character-level detection and recognition very well, especially when the number of categories is very large. Moreover, the proposed weakly supervised learning framework can effectively train the model using only transcript annotations.

4.7 Visualizations

The visualization results are shown in Fig. 8. It can be seen that the character detection and recognition results and the reading order can be accurately predicted, although there is no bounding box annotation for real data.



Fig. 8 Visualization results of PageNet. For each pair of images, the left is the character detection and recognition results and the right is the predicted reading order. In the visualization of the reading order, each circle represents a character (orange circle: start-of-line; green circle: end-of-line). From the first row to the fourth row, the images are from ICDAR13, MTHv2, SCUT-HCCDoc, and JS-SCUT PrintCC, respectively. In the last row, the character recognition results of the images from JS-SCUT PrintCC are not visualized, because the characters are too small and densely distributed. Zoom in for a better view.

The synthetic images of MTHv2 and SCUT-HCCDoc are synthesized by simply placing characters from font files on simple backgrounds. However, the visualization results demonstrate that our method learns to handle diverse handwriting styles, complex backgrounds and layouts, various perspective transformations, and uneven illuminations in real samples through the proposed weakly supervised learning framework. As shown in the visualization results from JS-SCUT PrintCC, our method can also process multilingual texts including both Chinese and English.

4.8 Experiments on Weakly Supervised Learning

4.8.1 Effectiveness of Semantic Matching

In Sec. 3.5.3, semantic matching is proposed to find reliable character-level results, which consists of line matching based on AR and character matching based on edit distance. However, some existing methods (Xing et al., 2019; Baek et al., 2019) simply use the length of predictions to determine the reliable results. Therefore, we compare our semantic matching with two algorithms following their ideas, which are *Page-level Length* and *Line-level Length*.

Table 6 Comparison of different matching and updating algorithms on ICDAR13

| Part | Algorithm | AR* | CR* |
|-------------------|--|--------------|--------------|
| Semantic Matching | Page-level Length (Xing et al., 2019; Baek et al., 2019) | 15.18 | 15.54 |
| | Line-level Length (Xing et al., 2019; Baek et al., 2019) | 41.90 | 44.70 |
| Spatial Matching | No Spatial Matching | 91.62 | 92.11 |
| | Replace (Xing et al., 2019; Baek et al., 2019; Wigington et al., 2018) | 12.24 | 64.53 |
| | Average | 91.58 | 92.28 |
| Updating | Fixed Ratio | 91.51 | 92.17 |
| | Ours | 92.83 | 93.23 |

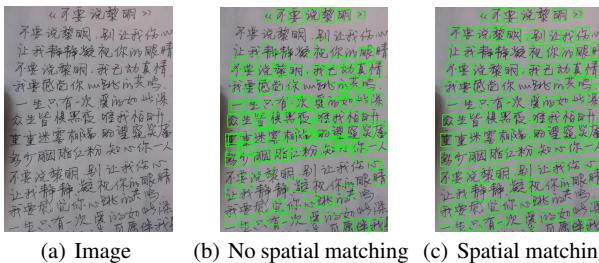
Specifically, in *Page-level Length*, the predicted results are regarded as reliable if the total number of characters on a page in the predictions is equal to that in the annotations. For *Line-level Length*, because the bounding box annotations of text lines required by (Xing et al., 2019; Baek et al., 2019) are not used in our method, we first perform line matching using Algorithm 1. Then, a line-level result is viewed as reliable if the lengths of it and the matched line-level transcript are equal.

The results of our approach and the other two algorithms are presented in Table 6. It can be seen that our semantic matching outperforms both *Page-level Length* and *Line-level Length* by a large margin.

4.8.2 Effectiveness of Spatial Matching

Spatial matching is proposed to solve the matching ambiguity of semantic matching as described in Sec. 3.5.3. As shown in Table 6, the performance decreases from 92.83% to 91.62% when spatial matching is removed.

We also provide qualitative results of spatial matching in Fig. 9. The page in Fig. 9(a) is from SCUT-HCCDoc and contains several lines with identical or similar contents. When conducting semantic matching, owing to the lack of location information in the annotations, one line-level result at the same location may be matched to different line-level transcripts with similar contents at different iterations. Then, the updating algorithm will cause inaccurate pseudo-labels, as shown in Fig. 9(b). Spatial matching can prevent incorrect



(a) Image (b) No spatial matching (c) Spatial matching

Fig. 9 (a): Original image. (b): The visualization of pseudo-labels without spatial matching. (c): The visualization of pseudo-labels with spatial matching.

matches and result in accurate pseudo-labels, as shown in Fig. 9(c).

4.8.3 Effectiveness of Updating

To verify the effectiveness of our updating algorithm, we compare it with three algorithms denoted as *Replace*, *Average*, and *Fixed Ratio*, which replace lines 6-8 of Algorithm 3 with Eq. (23), (24), and (25), respectively.

$$A_{ps}^{(q,n)} = (x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)}), \quad (23)$$

$$A_{ps}^{(q,n)} = \frac{k}{k+1} * A_{ps}^{(q,n)} + \frac{1}{k+1} * (x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)}), \quad (24)$$

$$A_{ps}^{(q,n)} = 0.9 * A_{ps}^{(q,n)} + 0.1 * (x^{(p,m)}, y^{(p,m)}, w^{(p,m)}, h^{(p,m)}), \quad (25)$$

where k is the number of times $A_{ps}^{(q,n)}$ has been updated.

The results in Table 6 show that our updating algorithm achieves the best performance. Especially compared with the *Replace* algorithm which is commonly adopted by previous methods (Xing et al., 2019; Baek et al., 2019; Wigington et al., 2018), our updating algorithm exhibits a significant improvement in performance. The *Replace* algorithm directly copies the new matched bounding boxes from the results as the updated pseudo-labels, which makes the model training easily interfered with by poor predictions. Fig. 10 shows the curves of AR* and CR* on ICDAR13 during the training stage when using the *Replace* algorithm. It can be seen that as the training progresses,

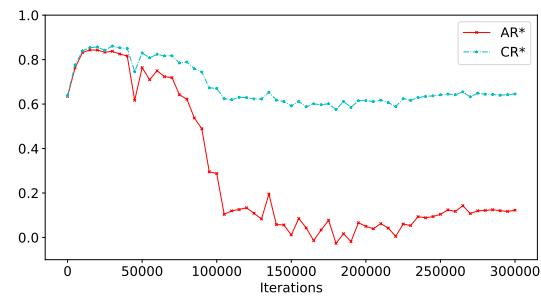
**Fig. 10** Curves of AR* and CR* on ICDAR13 during the training stage when using the *Replace* algorithm.

Table 7 Performance of PageNet under different supervision

| Supervision | ICDAR13 | | MTHv2 | |
|-------------|--------------|--------------|--------------|--------------|
| | AR* | CR* | AR* | CR* |
| Full | 91.37 | 91.88 | 93.81 | 95.54 |
| Weak | 92.83 | 93.23 | 93.76 | 95.23 |

**Fig. 11** Visualizations of bounding boxes from annotations and pseudo-labels.

the performance decreases, eventually converging at 12.24% AR*.

4.8.4 Comparison with Fully Supervised Learning

Because the CASIA-HWDB2.0-2.2 and MTHv2 datasets provide character-level bounding box annotations, we present the results of our method under full supervision in Table 7. Compared with the weakly supervised PageNet, the fully supervised counterpart uses annotated bounding boxes of real samples for loss calculation instead of leveraging the proposed weakly supervised learning framework. As for other details, the models under different supervision share the same settings.

As shown in Table 7, compared with the fully supervised counterpart, the weakly supervised PageNet achieves better performance on ICDAR13 and comparable performance on MTHv2, which demonstrates the success of the proposed weakly supervised learning framework. The automatically generated pseudo-labels can avoid inaccurate and inappropriate bounding boxes in manual annotations. In Fig. 11(a), the pseudo-labels are more accurate than the annotations. In Fig. 11(b), the bounding box annotations of punctuations in CASIA-HWDB2.0-2.2 are usually very small, which makes it difficult for the detection part to converge, but the pseudo-labels can avoid such inappropriate annotations. Furthermore, the iteratively updated pseudo-labels may serve as data augmentation. Although the images remain unchanged, the annotations are changeable at different iterations, which can improve the robustness of the model.

4.8.5 Ablation Experiments on Training Strategy

In Table 8, ablation experiments on the training strategy are conducted using ICDAR13. The results show that the proposed training strategy, which consists of pretraining, initializing, and training stages as described in Sec. 4.2, achieves the best performance. The pretraining and

Table 8 Ablation experiments on the training strategy (evaluated on ICDAR13)

| Pretraining | Initializing | Training | AR* | CR* |
|-------------|--------------|----------|--------------|--------------|
| ✓ | | ✓ | 64.94 | 65.27 |
| ✓ | | ✓ | 80.17 | 85.60 |
| ✓ | ✓ | ✓ | 91.57 | 92.15 |
| | | ✓ | 92.83 | 93.23 |

initializing stages are aimed at improving the efficiency of the training stage. Specifically, if the pretraining stage is not adopted, the model at early iterations cannot correctly detect and recognize any characters; thus, the real samples loaded at early iterations are wasted because there is no pseudo-label generated. If the initializing stage is not adopted, the early iterations of the training stage will be wasted on assigning the first round of pseudo-labels to the real samples.

4.9 Experiments on Reading Order

4.9.1 Multi-directional Reading Order

Most existing methods simply arrange the recognized characters in a text line from left to right, ignoring the complex reading order in the real world. However, taking advantage of the proposed reading order module and graph-based decoding algorithm, our method is able to recognize pages with multi-directional reading order.

For further verification, a multi-directional reading order experiment is conducted on ICDAR13. All the training and testing images are rotated clockwise by 90°, 180°, and 270°,

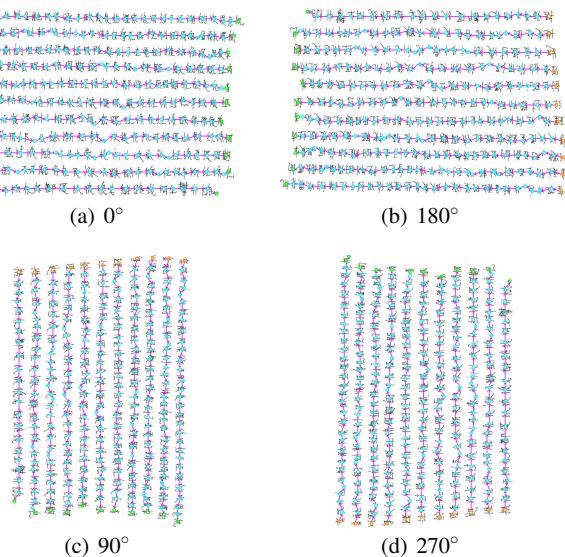
**Fig. 12** Reading orders predicted by PageNet of the same image rotated by different degrees, where each circle represents a character (orange circle: start-of-line; green circle: end-of-line).

Table 9 Performance of PageNet on multi-directional texts

| Supervision | 0° | | 90° | | 180° | | 270° | | Total | |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | AR* | CR* |
| Full | 89.41 | 90.29 | 89.50 | 90.38 | 89.43 | 90.32 | 89.39 | 90.32 | 89.43 | 90.33 |
| Weak | 89.48 | 90.46 | 89.55 | 90.52 | 89.55 | 90.52 | 89.49 | 90.50 | 89.52 | 90.50 |

but the reading order in the annotations remains unchanged. Using the original data and three rotated versions, we train and evaluate two models that are under full and weak supervision, respectively. Note that each of the two models is trained for all the four directions.

The experimental results are presented in Table 9. The performance of one model on the rotated data is comparable to that on the unrotated data, and the total performance still maintains a high accuracy. The predicted reading orders of the same image rotated by different angles are visualized in Fig. 12. It can be seen that the reading order is correctly predicted for all directions.

4.9.2 Curved Text Lines

We also conduct experiments on curved text lines. Both the training and testing sets are synthetic pages containing curved text lines, where the y coordinate of the character in a text line is a sine function of the x coordinate. To avoid overfitting, the training set uses the character samples from CASIA-HWDB1.0-1.2, while the testing set uses the character samples from ICDAR13-SC. Because the synthetic data has full annotations, the model is fully supervised. The AR* and CR* on the testing set are 94.04% and 94.36%, respectively. In addition, Fig. 13 shows the visualization results. Owing to the bottom-up design and strong reading order predicting mechanism, curved text lines can be effectively recognized by our method.

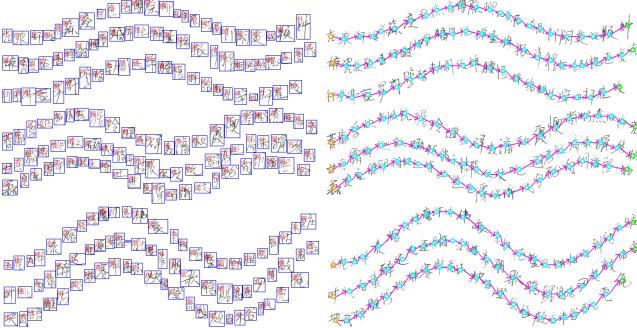


Fig. 13 Visualization results of a page containing curved text lines. The two images present character-level detection and recognition results (left) and reading order (right). In the visualization of reading order, each circle represents a character (orange circle: start-of-line; green circle: end-of-line).

Table 10 Comparison of different decoding algorithms

| Algorithm | ICDAR13 | | SCUT-HCCDoc | |
|------------|--------------|--------------|--------------|--------------|
| | AR* | CR* | AR* | CR* |
| Rule-based | 75.28 | 82.62 | 68.49 | 77.45 |
| Ours | 92.83 | 93.23 | 77.95 | 82.15 |

4.10 Experiments on Decoding Algorithm

In Table 10, we compare the graph-based decoding algorithm with the rule-based algorithm. The rule-based algorithm groups the characters based on their vertical coordinates and reorders the characters in each group according to their horizontal coordinates. It can be seen that the proposed graph-based decoding algorithm achieves significantly better performance owing to the effective design of the reading order prediction.

4.11 Automatic Labeling

Another potential application of the proposed weakly supervised learning framework is automatic labeling. Given the line-level transcripts of a page, the bounding boxes of characters can be automatically annotated by the pseudo-labels generated by our method.

Fig. 14 illustrates the automatically generated annotations for CASIA-HWDB2.0-2.2. After the training stage, 98.75% of the characters in CASIA-HWDB2.0-2.2 have corresponding pseudo-labels, and the average IoU between the pseudo-labels and ground-truth bounding boxes is 86.45%, which indicates the high quality of automatic labeling.

To verify the applicability of automatically generated annotations, we replace the original annotations of CASIA-HWDB2.0-2.2 with automatically labeled bounding boxes.



Fig. 14 Automatically generated annotations for CASIA-HWDB2.0-2.2.

Table 11 Effectiveness of automatically labeled annotations. This table presents the character detection performance on ICDAR13 when Faster R-CNN is trained using different annotations. The performance using original annotations is copied from Table 5.

| Annotations | Precision | Recall | F-measure |
|-----------------------|-----------|--------|-----------|
| Original | 98.93 | 92.12 | 95.41 |
| Automatically labeled | 97.70 | 91.78 | 94.64 |

Then, a Faster R-CNN (Ren et al., 2017) is trained to detect characters using CASIA-HWDB2.0-2.2 with new annotations and CASIA-SR, and is tested on ICDAR13. As shown in Table 11, compared with the Faster R-CNN using original annotations, the counterpart using automatically labeled annotations achieves comparable performance on character detection.

4.12 Effects of Synthetic Data

Recently, in order to improve the performance of text detection and recognition, many methods (Jaderberg et al., 2014; Gupta et al., 2016; Zhan et al., 2018) have been proposed for data synthesis. However, taking advantage of the proposed weakly supervised learning framework, the performance of our method does not rely heavily on the quality of synthetic data. For our method, the synthetic data are synthesized following a simple and unified procedure for all real datasets, which greatly reduces the labor required to design specific synthesis methods for different scenarios.

We analyze the effects of different synthetic data in Table 12. Compared with CASIA-SR, CASIA-SF is synthesized using character samples from font files rather than real character samples. Obviously, CASIA-SR is more similar to the real data than CASIA-SF. Compared with the performance without synthetic data, Det + Recog performs better using CASIA-SR but worse using CASIA-

Table 12 Effects of different synthetic data (evaluated on ICDAR13)

| Method | Synthetic Data | AR* | CR* |
|-------------|--------------------------------|---------------|---------------|
| | No synthetic data | 86.27 | 87.37 |
| Det + Recog | CASIA-SR | 88.36 (2.09↑) | 89.09 (1.72↑) |
| | CASIA-SF | 86.14 (0.13↓) | 87.30 (0.07↓) |
| PageNet | No synthetic data ¹ | 87.03 | 87.63 |
| | CASIA-SR | 92.83 (5.80↑) | 93.23 (5.60↑) |
| | CASIA-SF | 89.56 (2.53↑) | 90.52 (2.89↑) |

¹ PageNet without synthetic data is trained under full supervision.

Table 13 Improvement of the final model compared with the pretrained model

| Model | ICDAR13 | | | | MTHv2 | | SCUT-HCCDoc | | JS-SCUT PrintCC | |
|------------------|----------|--------|----------|---------|--------|--------|-------------|---------|-----------------|--------|
| | CASIA-SR | | CASIA-SF | | AR* | CR* | AR* | CR* | AR* | CR* |
| | AR* | CR* | AR* | CR* | | | | | | |
| Pretrained Model | 63.62 | 64.00 | 38.70 | 39.01 | 54.30 | 61.61 | 29.93 | 32.94 | 75.99 | 79.51 |
| Final Model | 92.83 | 93.23 | 89.56 | 90.52 | 93.76 | 95.23 | 77.95 | 82.15 | 97.25 | 98.19 |
| Improvement | 45.92% | 45.67% | 131.42% | 132.04% | 72.67% | 54.57% | 160.44% | 149.39% | 27.98% | 23.49% |

specifically designed, our method can work well on all the benchmark datasets.

Therefore, it may be safe to say that our method can be easily generalized to different scenarios. Nevertheless, the method in this paper is mainly for document-based text recognition. It may be less effective for other complex scenarios such as end-to-end scene text recognition, which is still an open problem that deserves further study.

5 Conclusion

In this paper, we propose PageNet for solving end-to-end weakly supervised page-level handwritten Chinese text recognition from a new perspective. With only line-level transcripts annotated for real data, PageNet is able to end-to-end predict detection and recognition results at both the character and line levels, as well as the important reading order of each text line. Extensive experiments on five datasets, including CASIA-HWDB, ICDAR2013, MTHv2, SCUT-HCCDoc, and JS-SCUT PrintCC, demonstrate that our method can achieve state-of-the-art performance, even when compared with fully supervised methods. We further show that the proposed PageNet can surpass the line-level methods of handwritten Chinese text recognition which directly recognize the pre-supplied cropped text line images. It is worth mentioning that our method can serve as an automatic annotator that can produce highly accurate character-level bounding boxes. As there are thousands of web images with only transcript labels on the Internet, the powerful generalization ability of PageNet exhibits its promising potential in real-world applications. We hope that this work opens up new possibilities for end-to-end weakly supervised page-level text recognition.

Acknowledgement

This research is supported in part by NSFC (Grant No.: 61936003), GD-NSF (no.2017A030312006, No.2021A1515011870), and the Science and Technology Foundation of Guangzhou Huangpu Development District (Grant 2020GH17).

References

- Rodriguez-Serrano JA, Gordo A, Perronnin F (2015) Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision* 113(3):193–207
- Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116(1):1–20
- Feng W, Yin F, Zhang XY, He W, Liu CL (2021) Residual dual scale scene text spotting by fusing bottom-up and top-down processing. *International Journal of Computer Vision* 129(3):619–637
- Liu Z, Lin G, Goh WL (2020) Bottom-up scene text detection with Markov clustering networks. *International Journal of Computer Vision* pp 1–24
- Liu Y, He T, Chen H, Wang X, Luo C, Zhang S, Shen C, Jin L (2021) Exploring the capacity of an orderless box discretization network for multi-orientation scene text detection. *International Journal of Computer Vision* 129(6):1972–1992
- Luo C, Lin Q, Liu Y, Jin L, Shen C (2021) Separating content from style using adversarial learning for recognizing text in the wild. *International Journal of Computer Vision* 129(4):960–976
- Baek Y, Lee B, Han D, Yun S, Lee H (2019) Character region awareness for text detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp 9365–9374
- Bluche T (2016) Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In: *Proceedings of Advances in Neural Information Processing Systems*, pp 838–846
- Bluche T, Louradour J, Messina R (2017) Scan, Attend and Read: End-to-end handwritten paragraph recognition with MDLSTM attention. In: *Proceedings of International Conference on Document Analysis and Recognition*, vol 01, pp 1050–1055
- Carbonell M, Mas J, Villegas M, Fornés A, Lladós J (2019) End-to-end handwritten text detection and transcription in full pages. In: *Proceedings of International Conference on Document Analysis and Recognition Workshops*, vol 5, pp 29–34
- Chung J, Delteil T (2019) A computationally efficient pipeline approach to full page offline handwritten text recognition. In: *Proceedings of International Conference on Document Analysis and Recognition Workshops*, vol 5, pp 35–40
- Du J, Zi-Rui Wang, Zhai J, Hu J (2016) Deep neural network based hidden Markov model for offline handwritten Chinese text recognition. In: *Proceedings of IEEE International Conference on Pattern Recognition*, pp 3428–3433
- Graves A, Jaitly N (2014) Towards end-to-end speech recognition with recurrent neural networks. In: *Proceedings of International Conference on Machine Learning*, pp 1764–1772
- Graves A, Fernández S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *Proceedings of International Conference on Machine Learning*, pp 369–376
- Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5):855–868
- Gupta A, Vedaldi A, Zisserman A (2016) Synthetic data for text localisation in natural images. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp 2315–2324
- He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: *Proceedings of IEEE International Conference on Computer Vision*, pp 2961–2969
- Huang Y, Xie Z, Jin L, Zhu Y, Zhang S (2019) Adversarial feature enhancing network for end-to-end handwritten paragraph recognition. In: *Proceedings of International Conference on Document Analysis and Recognition*, pp 413–419
- Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2014) Synthetic data and artificial neural networks for natural scene text recognition. In: *Proceedings of Advances in Neural Information Processing Systems Deep Learn. Workshop*
- Keyser D, Deselaers T, Rowley HA, Wang L, Carbune V (2017) Multi-language online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6):1180–1194
- Liao M, Lyu P, He M, Yao C, Wu W, Bai X (2021) Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43(2):532–548
- Liu C, Yin F, Wang D, Wang Q (2011) CASIA online and offline Chinese handwriting databases. In: *Proceedings of International Conference on Chinese Handwriting Recognition*, pp 1–10

- Conference on Document Analysis and Recognition, pp 37–41
- Liu X, Liang D, Yan S, Chen D, Qiao Y, Yan J (2018) FOTS: Fast oriented text spotting with a unified network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 5676–5685
- Lyu P, Liao M, Yao C, Wu W, Bai X (2018) Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of European Conference on Computer Vision, pp 67–83
- Ma W, Zhang H, Jin L, Wu S, Wang J, Wang Y (2020) Joint layout analysis, character detection and recognition for historical document digitization. In: Proceedings of International Conference on Frontiers in Handwriting Recognition, pp 31–36
- Messina R, Louradour J (2015) Segmentation-free handwritten Chinese text recognition with LSTM-RNN. In: Proceedings of International Conference on Document Analysis and Recognition, pp 171–175
- Moysset B, Kermorvant C, Wolf C (2017) Full-page text recognition: Learning where to start and when to stop. In: Proceedings of International Conference on Document Analysis and Recognition, vol 01, pp 871–876
- Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: Proceedings of IEEE International Conference on Pattern Recognition, pp 850–855
- Peng D, Jin L, Wu Y, Wang Z, Cai M (2019) A fast and accurate fully convolutional network for end-to-end handwritten Chinese text segmentation and recognition. In: Proceedings of International Conference on Document Analysis and Recognition, pp 25–30
- Redmon J, Farhadi A (2018) YOLOv3: An incremental improvement. arXiv preprint arXiv:180402767
- Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6):1137–1149
- Shi B, Bai X, Yao C (2017) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(11):2298–2304
- Su TH, Zhang TW, Guan DJ, Huang HJ (2009) Off-line recognition of realistic Chinese handwriting using segmentation-free strategy. Pattern Recognition 42(1):167–182
- Tensmeyer C, Wigington C (2019) Training full-page handwritten text recognition models without annotated line breaks. In: Proceedings of International Conference on Document Analysis and Recognition, pp 1–8
- Wang Q, Yin F, Liu C (2012) Handwritten Chinese text recognition by integrating multiple contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(8):1469–1481
- Wang S, Chen L, Xu L, Fan W, Sun J, Naoi S (2016) Deep knowledge training and heterogeneous CNN for handwritten Chinese text recognition. In: Proceedings of International Conference on Frontiers in Handwriting Recognition, pp 84–89
- Wang ZR, Du J, Wang WC, Zhai JF, Hu JS (2018) A comprehensive study of hybrid neural network hidden Markov model for offline handwritten Chinese text recognition. International Journal on Document Analysis and Recognition 21(4):241–251
- Wang ZR, Du J, Wang JM (2020a) Writer-aware CNN for parsimonious HMM-based offline handwritten Chinese text recognition. Pattern Recognition 100:107102
- Wang ZX, Wang QF, Yin F, Liu CL (2020b) Weakly supervised learning for over-segmentation based handwritten Chinese text recognition. In: Proceedings of International Conference on Frontiers in Handwriting Recognition, pp 157–162
- Wigington C, Tensmeyer C, Davis B, Barrett W, Price B, Cohen S (2018) Start, Follow, Read: End-to-end full-page handwriting recognition. In: Proceedings of European Conference on Computer Vision, pp 367–383
- Wu Y, Yin F, Chen Z, Liu C (2017) Handwritten Chinese text recognition using separable multi-dimensional recurrent neural network. In: Proceedings of International Conference on Document Analysis and Recognition, vol 01, pp 79–84
- Wu YC, Yin F, Liu CL (2017) Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognition 65:251–264
- Xie C, Lai S, Jin L, Liao Q (2020) High performance offline handwritten Chinese text recognition with a new data preprocessing and augmentation pipeline. In: Proceedings of International Workshop on Document Analysis Systems, pp 45–59
- Xie Z, Sun Z, Jin L, Ni H, Lyons T (2018) Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(8):1903–1917
- Xie Z, Huang Y, Jin L, Liu Y, Zhu Y, Gao L, Zhang X (2019a) Weakly supervised precise segmentation for historical document images. Neurocomputing 350:271–281
- Xie Z, Huang Y, Zhu Y, Jin L, Liu Y, Xie L (2019b) Aggregation cross-entropy for sequence recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 6538–6547
- Xing L, Tian Z, Huang W, Scott MR (2019) Convolutional character networks. In: Proceedings of IEEE International Conference on Computer Vision, pp 9126–9136
- Xiu Y, Wang Q, Zhan H, Lan M, Lu Y (2019) A handwritten Chinese text recognizer applying multi-level multimodal fusion network. In: Proceedings of International Conference on Document Analysis and Recognition, pp 1464–1469
- Yang H, Jin L, Huang W, Yang Z, Lai S, Sun J (2018) Dense and tight detection of Chinese characters in historical documents: Datasets and a recognition guided detector. IEEE Access 6:30174–30183
- Yang H, Jin L, Sun J (2018) Recognition of Chinese text in historical documents with page-level annotations. In: Proceedings of International Conference on Frontiers in Handwriting Recognition, pp 199–204
- Yin F, Wang Q, Zhang X, Liu C (2013) ICDAR 2013 Chinese handwriting recognition competition. In: Proceedings of International Conference on Document Analysis and Recognition, pp 1464–1470
- Yousef M, Bishop TE (2020) OrigamiNet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 14710–14719
- Zhan F, Lu S, Xue C (2018) Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In: Proceedings of European Conference on Computer Vision, pp 249–266
- Zhang H, Liang L, Jin L (2020) SCUT-HCCDoc: A new benchmark dataset of handwritten Chinese text in unconstrained camera-captured documents. Pattern Recognition 108:107559
- Zhang R, Zhou Y, Jiang Q, Song Q, Li N, Zhou K, Wang L, Wang D, Liao M, Yang M, Bai X, Shi B, Karatzas D, Lu S, Jawahar CV (2019) ICDAR 2019 robust reading challenge on reading Chinese text on signboard. In: Proceedings of International Conference on Document Analysis and Recognition, pp 1577–1581
- Zhang X, Yin F, Zhang Y, Liu C, Bengio Y (2018) Drawing and recognizing Chinese characters with recurrent neural network. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(4):849–862
- Zhou X, Wang D, Tian F, Liu C, Nakagawa M (2013) Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(10):2413–2426
- Zhu ZY, Yin F, Wang DH (2020) Attention combination of sequence models for handwritten Chinese text recognition. In: Proceedings of

International Conference on Frontiers in Handwriting Recognition,
pp 288–294
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image
recognition. In: Proceedings of the IEEE Conference on Computer
Vision and Pattern Recognition, pp 770–778