A Direct Regression Scene Text Detector With Position-Sensitive Segmentation

Peirui Cheng, *Student Member, IEEE*, Yuanqiang Cai, and Weiqiang Wang, *Member, IEEE*

***Abstract*— Direct regression methods have demonstrated their success on various multi-oriented benchmarks for scene text detection due to the high recall rate for small targets and the direct regression for text boxes. However, too many false positive candidates and inaccurate position regression still limit the performance of these methods. In this paper, we propose an end-to-end method by introducing position-sensitive segmentation into the direct regression method to overcome these shortcomings. We generate the ground truth of position-sensitive segmentation maps based on the information of text boxes so that the position-sensitive segmentation module can be trained synchronously with the direct regression module. Besides, more information about the relative position of text is provided for the network through the training of position-sensitive segmentation maps, which improves the expressiveness of the network. We also introduce spatial pyramid of position-sensitive segmentation into the proposed method considering the huge differences in sizes and aspect ratios of scene texts and we propose position-sensitive COI(Corner area of Interest) pooling into the proposed method to speed up the inference. Experiments on datasets ICDAR2015, MLT-17 and COCO-Text demonstrate that the proposed method has a comparable performance with state-of-the-art methods while it is more efficient. We also provide abundant ablation experiments to demonstrate the effectiveness of these improvements in our proposed method.**

***Index Terms*— Scene text detection, fully convolutional network, direct regression, position-sensitive segmentation.**

I. Introduction

NATURAL scene text detection and recognition is becoming more and more popular with the development of computer vision. As the first step of the process, scene text detection has a great influence on the result of subsequent text recognition and it plays an important role in various applications, such as image retrieval, blind navigation, and mobile OCR. Besides, It is a very challenging task due to multiple orientations, perspective distortions, motion blur and large variation of text sizes, colors and aspect ratios. Therefore, scene text detection is a research direction worth exploring.

With the rapid development of deep neural network, various methods [1]–[8] have demonstrated their success on diverse benchmarks for scene text detection. These methods can be classified into three categories: character based methods [1], [2], anchor based methods [9]–[11] and direct regression methods [12], [13]. The character based methods have a poor performance on challenging benchmarks due to the huge difficulty of segmenting a character with complex background and complex post-processing operations. The anchor based methods need to preset a vast number of anchors due to the multiple orientations and the huge variation of text sizes and aspect ratios, which lowers the efficiency of detection.

Compared with other methods, the direct regression methods perform better on multi-oriented scene text detection datasets. They generally merge the high level and low level feature maps to generate the high-resolution text score maps to detect small text targets. Besides, they directly regress the sizes and orientations of text boxes without presetting anchors so that they have a higher efficiency than other methods.

However, there are still some problems which limit the performance of direct regression methods. First, these methods generate lots of false positive candidates when facing a complex background or a scenario with compact text distributions. As one-stage detection methods, the direct regression methods directly generate text boxes without any refinements. Whether or not a text box is generated just depends on the value of a point on the score map. Hence these methods are not comparable with the two-stage text detection methods [10], [11] in precision on challenging scene text detection datasets. Second, the locations of many text boxes regressed by these methods are inaccurate. Texts are sequences of a series of characters, which means it is easy for these methods to regress text boxes which contain only a portion of the sequence or wrap two sequences together, as shown in Figure 1. Some of these inaccurate text boxes may affect the performance of the methods and some may be considered as true positive samples based on the IOU evaluation criteria with a threshold value of 0.5. However, they still limit the performance of subsequent text recognition due to inaccurate location regression.

In this paper, we propose a direct regression method for scene text detection based on the position-sensitive segmentation [14] to solve the above-mentioned problems. Position-sensitive segmentation module was first proposed in the instance-aware segmentation methods [14], [15].

Manuscript received June 2, 2019; revised August 18, 2019 and September 16, 2019; accepted September 30, 2019. Date of publication October 15, 2019; date of current version October 29, 2020. This work was supported in part by the National Key Research and Development Program of China under Contract 2017YFB1002203, in part by the National Nature Science Foundation of China (NSFC) under Grant 61976201, and in part by the NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant 61860206004. This article was recommended by Associate Editor Y. Yang. *(Corresponding author: Weiqiang Wang.)*

The authors are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: chengpeirui13@mails.ucas.edu.cn; caiyuanqiang15@mails.ucas.ac.cn; wqwang@ucas.ac.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2019.2947475

Fig. 1. The examples of inaccurate text boxes regression. (a) The ground truth of input image. (b) The predicted image. Some candidates contain a portion of the sequence due to the difference of character size or wrap two sequences together due to the compact text distribution.

However, it is not suitable to add the position-sensitive segmentation module mechanically into the direct regression method for scene text detection, because the popular scene text datasets [16]–[18] do not provide the instance-aware segmentation information for the position-sensitive segmentation module. The position-sensitive segmentation module was then applied to object detection methods [19]. However, this version is also not suitable for direct regression methods because it was only applied in the two-stage object detection methods previously while the direct regression methods are the one-stage detection methods with all tasks trained synchronously. Therefore, we improve the position-sensitive segmentation module according to the characteristics of the scene text and the architecture of the direct regression method. First, our proposed method uses an FCN (Fully Convolutional Network) [20] to extract features for training three tasks simultaneously. The first task is to do text or non-text segmentation on the down-sampled feature map and the second task is to regress the distance and angle from the point to the boundaries of the corresponding rectangle on the feature maps [12]. The third task is to do position-sensitive segmentation for all text boxes and obtain a series of segmentation maps corresponding to different relative positions of text boxes. Second, we propose the spatial pyramid of position-sensitive segmentation to do position-sensitive segmentation on different scales considering the huge differences in sizes and aspect ratios of scene texts. Third, we propose the position-sensitive COI (Corner area of Interest) pooling instead of the position-sensitive ROI(Region of Interest) pooling to raise the efficiency of the proposed method.

This paper is an extension of our previous conference paper [21], and the major improvements fall in three respects:

- We improve the position-sensitive segmentation module by introducing the spatial pyramid of position-sensitive segmentation and proposing the position-sensitive COI (Corner area of Interest) pooling operation. Experimental results show that the proposed method achieves better performance on scene text detection datasets with these improvements.
- We stress the effect of generating the ground truth of position-sensitive segmentation maps and the contribution of the position-sensitive segmentation module to the accuracy of text boxes' location regression. These effects are demonstrated experimentally.
- We provide more ablation experiments to analyze the proposed method in different aspects. Besides, we also provide more experiments with different backbone networks and perform experiments on different datasets to verify the effectiveness of our proposed method.

II. Related Work

A. Scene Text Detection

Conventional scene text detection methods used hand-crafted features, such as stroke width [22], extremal region [23], symmetry [24] to detect text. Among them, Stroke Width Transform(SWT) [22] and Maximally Stable Extremal Regions(MSER) [23] were the mainstream. They mainly obtain character candidates by extracting connected components, followed by false positive filtering, character pairing, text line grouping, and word partition. The performance of these approaches heavily degrades on some challenging benchmarks [16], [18] due to the only use of low-level features. Additionally, they suffer from low efficiency due to multiple complex steps in them.

Recently, the progress of deep neural network has tremendously promoted the performance of scene text detection. The character based methods use the FCN to extract proper features for detecting character candidates. Zhang *et al.* [25] use the FCN to generate text saliency map and use component based information and context of the text block to locate text lines. Yao *et al.* [26] use a single network for predicting text regions, character candidates and linking orientation map so that it can detect multi-oriented and curved text from multiple channels. Shi *et al.* [1] modify the SSD [27] framework to make the network be able to detect characters and learn linking information between adjacent characters. These methods often need multiple steps and their robustness suffers from character-level detection on the benchmarks with complex background.

The anchor based methods are mainly based on object detection methods [28]–[32], such as Faster R-CNN [33], SSD [27] and Mask R-CNN [19], and they generally generate many multi-oriented proposals with different sizes and aspect ratios to ensure the recall rate. Liao *et al.* [3] improve the SSD framework for scene text detection with changing the size and shape of convolutional kernels to fit the characteristic of scene text. Zhang *et al.* [9] propose Feature Enhancement Network to fuse global and local information and use adaptively weighted position-sensitive ROI pooling layer to raise the accuracy. Yang *et al.* [10] use FCIS [15] to detect multi-oriented text boxes from the perspective of instance segmentation. Xie *et al.* [11] propose SPCNET based on Mask R-CNN [19] to precisely locate text regions. These methods need a large number of proposals to deal with multi-oriented text detection, which makes the cost of calculation expensive.

The direct regression methods directly classify pixels on score maps into text and non-text, just as recent regression methods [34]–[36], and regress a candidate for each positive pixel. Zhou *et al.* [12] propose an approach which uses FCN to generate a pixel-wise text score map and geometry maps to represent the rotated candidates. He *et al.* [13] propose a deep direct regression method with pixel-wise classification and direct regression to determine the vertex coordinates of quadrilateral text boundaries. Liu *et al.* [8] and He *et al.* [37] add text recognition into direct regression methods and improve the performance of text detection because of joint training and sharing computation. Xue *et al.* [7] add a semantic-aware text border segmentation on the base of direct regression methods for localizing text boxes more accurately. These methods perform well due to the high-resolution score map for classification and have high efficiency due to the direct regression of text boxes.

Some methods [38], [39] try to combine scene text detection with semantic segmentation to improve the performance of text detection by reducing the number of false positive candidates. He *et al.* [38] propose a single shot detector combining semantic segmentation with proposal based text detection. It embeds a saliency network into the SSD framework to improve the performance. Qin and Manduchi [39] cascade a YOLO-like [40] direct regression network to a pixel-wise segmentation network. These methods mainly use semantic segmentation map to weaken background pixels so that they can reduce the number of false positive candidates. However, they can only eliminate the false positive candidates located on the background and they can not eliminate the false positive candidates shown in Figure 1.

B. Position-Sensitive Segmentation

Position-sensitive segmentation is first proposed in the instance-aware semantic segmentation methods [14], [15] and then it is used in the object detection methods [19]. Dai *et al.* [14] propose InstanceFCN to compute a small set of score maps, each of which is the outcome of a pixel-wise classifier of a relative position to instances. Li *et al.* [15] propose the fully convolutional end-to-end solution for instance-aware semantic segmentation task with the position-sensitive inside/outside score maps. Dai *et al.* [19] use position-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection.

Some text detection methods also use the position-sensitive segmentation module. Lyu *et al.* [41] propose to detect scene text by localizing corner points of text bounding boxes and use position-sensitive segmentation to filter out text boxes. However, the number of candidates generated by sampling and grouping corner points is so sizeable that using position-sensitive segmentation maps to score each candidate is a time-consuming process. Yang *et al.* [10] introduce deformable position-sensitive ROI pooling to deal with multi-oriented text detection. It uses RPN [33] architecture to generate proposals and then uses deformable position-sensitive ROI pooling to improve the detection result of text in arbitrary orientation.

In our proposed method, we also use position-sensitive segmentation to predict the relative position of text regions and filter out the false positive candidates. Compared with the instance-aware semantic segmentation methods [14], [15], there are three main differences. First, we optimize the position-sensitive segmentation maps synchronously with score maps and geometry maps. We directly generate the ground truth of position-sensitive segmentation maps based on the information provided by the scene text detection datasets. Second, we introduce the spatial pyramid of position-sensitive segmentation to improve the performance of the proposed method considering the large variations in sizes and aspect ratios of scene texts. Third, we propose the position-sensitive COI pooling instead of the position-sensitive ROI pooling to improve the efficiency of the proposed method with the same effect.

III. Proposed Methodology

A. Overview

The proposed method is an end-to-end method for detecting multi-oriented text in natural scenes. The overall architecture of the proposed method is shown in Figure 2. It consists of four major modules: feature extraction, direct regression, position-sensitive segmentation and post-processing module. The feature extraction module uses the U-shape Network [42] to extract feature maps. The direct regression module is based on EAST [12] to predict and regress text boxes. The position-sensitive segmentation module generates the spatial pyramid position-sensitive segmentation maps. The last post-processing module is executed at the testing stage to merge redundant candidates by the Locality-Aware NMS and score the candidates for filtering out the false positives by the position-sensitive COI pooling. Finally, the remaining candidates are the final output.

B. Feature Extraction

The architecture of the feature extraction network is shown in Figure 3. There are a few differences from the EAST. First, we use ResNet-50 [43] as the backbone of this network. The feature maps outputted by each stage's last residual block of ResNet-50 are extracted for concatenation. Second, an 1×1 bottleneck convolution is used to reduce the number of feature maps. Third, we add a 3×3 convolution before direct regression module and position-sensitive segmentation module respectively. The size of output feature maps is defined as the quarter of an input image in the view of the efficiency and accuracy of the proposed method.

C. Direct Regression

In the direct regression module, a score map and multi-channel geometry maps are obtained [12]. The positive area of one text box on the score map is a shrunk version of the original quadrangle. The geometry maps are made up of five channels. Among them, four channels respectively represent the distances from a pixel location to the top, right, bottom, left edges of the relevant minimum enclosing rectangles of scene

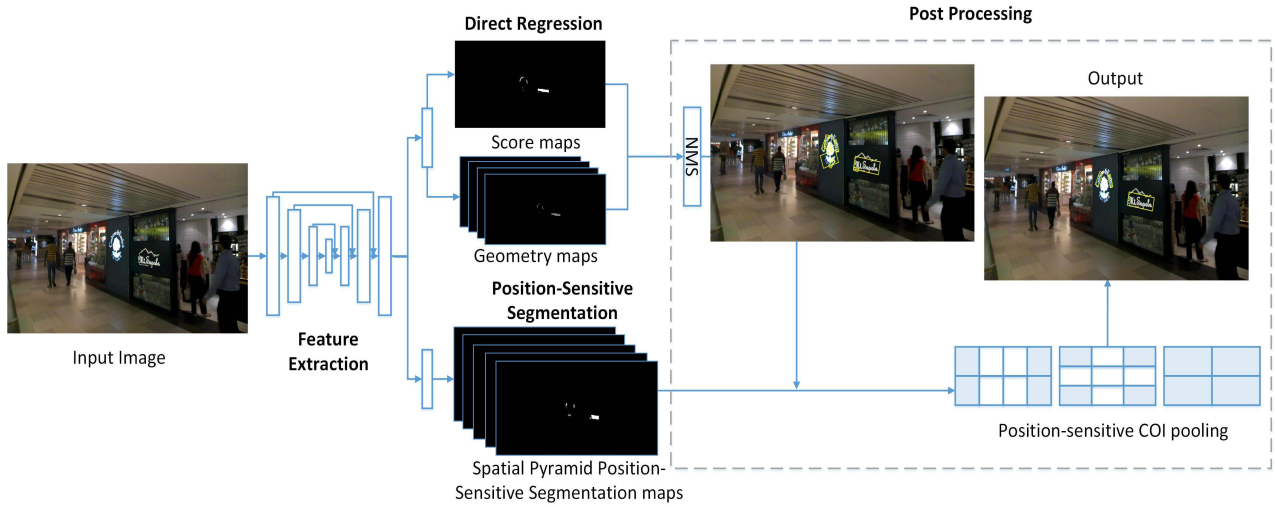


Fig. 2. Overall architecture of the proposed method. Feature extraction is an FCN and features are then input into direct regression and position-sensitive segmentation module respectively. Post-processing module is executed at the testing stage to merge redundant candidates and filter out false positive candidates using position-sensitive COI pooling.

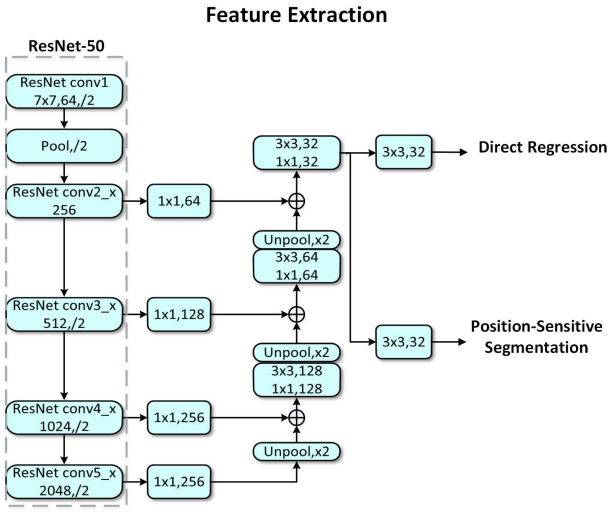


Fig. 3. Architecture of feature extraction module. ResNet-50 consists of $conv1 - conv5$. The upsampling operation is implemented by linear interpolation. The feature maps of the same size are concatenated.

text, and the fifth channel represents the angle of inclination of the relevant rectangle. If the score of a pixel on the score map is higher than a threshold, the corresponding candidate is a positive one. The location of the candidate is defined by the corresponding values on the geometry maps.

D. Position-Sensitive Segmentation

1) Motivation: Too many false positive candidates usually limits the performance of direct regression methods. For the scene text datasets, the complex background is one of the major reasons. Many background areas are easily mistaken for text areas because many non-text targets have a similar texture to the text, such as railings, icons, and strips. Besides, the compact distribution of scene texts also causes this problem. For example, several lines of texts are particularly close together or the gaps between the words are especially small, which is a common situation in scene text detection. For the direct regression methods, the direct regression of candidates and the lack of refinement of candidates result in numerous false positive candidates. Therefore, we intend to introduce a certain post-processing operation to filter out false positive candidates to improve the performance of the proposed method.

A simple way to filter out false positive candidates is to average the values of pixels in each candidate on the text segmentation map to score each candidate and use a threshold to remove false positive candidates. As shown in Figure 4(b), the blue box and the red box are the candidates generated by the direct regression module. The average value of pixels on the segmentation map involved by the blue box is large since the candidate exactly overlaps with a word. The red box can be eliminated easily by comparing its score against the threshold. This method can effectively eliminate the false positive candidates located on the background areas. However, it can not filter out the false positive candidates located between text lines or words. For example, a false positive candidate is located between text lines and it does not exactly overlap with a certain word, just as the yellow candidate in Figure 4(b). It is hard to filter out the candidate only according to this simple operation since most of the pixels on the segmentation map involved by the candidate are salient though they belong to different text instances.

Considering this problem, we intend to use the instance-aware segmentation of texts instead of the semantic segmentation to filter out the false positive candidates located between text lines. In the instance-aware segmentation of texts, pixels belonging to different instances are marked as different classes. In this case, we can calculate the instance score of candidates based on instance-aware segmentation maps and each candidate has a different instance score for different text instances. Then the instance score can be used to filter out the false positive candidates. According to instanceFCN [15], if the segmentation maps of different

Fig. 4. Position-sensitive segmentation module. (a) The input image. (b) The schematic map of the score map. (c) The schematic map of position-sensitive segmentation maps. (d) The schematic map of position-sensitive ROI pooling.

channels can represent the segmentation of different relative positions of text instances respectively, the ensemble of these position-sensitive segmentation maps can represent the instance-aware segmentation. Therefore, we use the position-sensitive segmentation maps to calculate the instance score of candidates. As shown in Figure 4(c), we divide each text box into four relative positions (top-left, top-right, bottom-left, bottom-right) and mark these areas in different colors. For each candidate, we also divide it into four parts and only average the values of the pixels of the corresponding color in each part. Then we score the candidate through average pooling. The yellow candidate can be eliminated easily while the blue candidate is preserved based on their scores, just as shown in Figure 4(d).

However, the position-sensitive segmentation module can not be directly added into the direct regression methods without any modification because scene text detection datasets do not provide pixel-level information for instance-aware segmentation. Besides, there are some characteristic features of scene text which differ from general targets. Therefore, we make several modifications to the position-sensitive segmentation module to adapt it to the direct regression methods.

2) Generating Ground Truth: The instance-aware segmentation methods [14], [15] need instance-aware segmentation information for training the position-sensitive segmentation maps. However, scene text detection datasets do not provide pixel-wise instance-aware segmentation information. Unlike [44], the proposed method does not need to recognize text instances, and it only needs to segment text areas of text instance instead of character-level segmentation on the position-sensitive segmentation maps. The two-stage detection methods [10], [19] assemble the feature of position-sensitive segmentation maps in the proposals which are obtained by RPN so that they can train the position-sensitive segmentation maps indirectly by calculating the classification loss and the position regression loss of proposals. For our proposed method, the position-sensitive segmentation maps are generated for filtering candidates rather than for character-level segmentation and we only need to divide the text areas into pieces on position-sensitive segmentation maps, which can be easy to implement by the location information of the text boxes. Therefore, our proposed method intends to use the supervisory information of text detection to generate the ground truth of position-sensitive segmentation maps for direct training.

First, we generate a rough text segmentation map by treating the area covered by each text box as the foreground and other areas as the background. Then we generate the mask for each relative position of text areas respectively. Finally, we use a series of masks to multiply the rough text segmentation map point-to-point to generate the ground truth of position-sensitive segmentation maps. Specifically, the steps to generate masks for relative positions of text areas are as follows. First, we draw a minimum enclosing rectangle for each foreground quadrangle area on the rough text segmentation map. Then each rectangle is cut evenly into several pieces. The number of pieces is the same as the number of the corresponding position-sensitive segmentation maps. Finally, we generate blank masks of the same size as the rough text segmentation map. For each mask, we set the value to 1 for the pixels which belong to the corresponding piece of each rectangle.

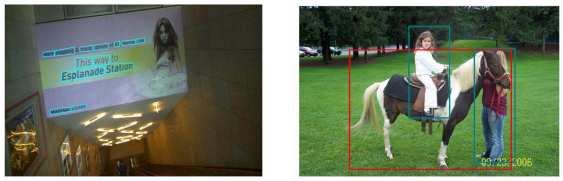


Fig. 5. The examples of scene text detection and object detection. (a) The scene text detection image marked with ground truth. (b) The object detection image marked with ground truth.

Compared with object detection methods [19], [45], the proposed method can use the generated ground truth to directly train the position-sensitive segmentation maps because of some characteristics of scene text which differ from the characteristics of general objects. The shape of scene texts is usually a rectangle or quadrangle generated by the projection transformation of a rectangle and the quadrilateral region surrounded by four vertices contains a little bit of background, just as shown in Figure 5(a). Hence we can regard the quadrilateral region roughly as a text area. By contrast, general objects are usually irregularly shaped and the rectangle which frames an object always contains some background. Besides, there are often some overlaps between the general object boxes because of the irregular shape of objects, just as shown in Figure 5(b). In this case, one pixel is often the foreground in one box and the background in another overlapping box. In contrast, text boxes rarely overlap with each other due to the regular shape of scene text, as shown in Figure 5(a). Therefore, we can directly use a rough text segmentation map to generate the ground truth of position-sensitive segmentation maps for scene text.

3) Spatial Pyramid of Position-Sensitive Segmentation: The scene text detection methods always face the problem of the huge difference in sizes and aspect ratios of scene text. The previous methods use a uniform parameter $k \times k$ to divide all candidates and generate a set of position-sensitive segmentation maps based on the parameter k. However, it is hard to select an appropriate parameter k for the scene text detection due to the great variety of sizes and aspect ratios. Therefore, we introduce the spatial pyramid of position-sensitive segmentation into the direct regression method.

First, we divide each minimum enclosing rectangle of the text box into windows of different sizes when generating the ground truth of position-sensitive segmentation maps. We set the parameters of spatial pyramid position-sensitive segmentation considering the character of scene text datasets. Due to the large difference in the sizes of scene texts and the large number of small texts, we choose two kinds of windows, 2×2 and 3×3. Besides, texts with different aspect ratios also need to be considered and long text is also a frequent example in text detection datasets. Therefore, we add another kind of window, 2×4. In the spatial pyramid of position-sensitive segmentation, we share these three kinds of windows 2×2, 3×3 and 2×4 to divide text boxes. Then we generate all the masks corresponding to different kinds of windows and use these masks to generate the ground truth of the spatial pyramid position-sensitive segmentation maps. Finally, the loss of spatial pyramid of position-sensitive segmentation maps is calculated respectively.

4) Calculating Loss Function: The proposed method directly calculate the loss for spatial pyramid position-sensitive segmentation maps and the loss L_{ps} is defined as

$$L_{ps} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T_i}\sum_{j=1}^{T_i}\left(1 - 2\frac{M_j^i G_{ps}^* S_j^i}{M_j^i G_{ps}^* + S_j^i}\right)\right), \tag{1}$$

which is based on dice loss [46]. N denotes the number of kinds of windows corresponding to different parameters and the value is 3 in the proposed method. T_i denotes the number of i-th set of position-sensitive segmentation maps. M_j^i is the mask of the j-th position-sensitive segmentation map in the i-th set. Pixels in mask M_j^i are set to 1 if they belong to the corresponding window of any text's minimum enclosing rectangle and 0 otherwise. G_{ps}^* denotes the rough text segmentation map, where pixels belonging to original quadrangles are set to 1 and 0 otherwise. S_j^i represents the j-th predicted position-sensitive segmentation map in the i-th set.

Through calculating the loss for spatial pyramid position-sensitive segmentation maps directly, the proposed method can train the direct regression module and the position-sensitive segmentation module synchronously. This strategy does not increase much training time compared with the previous detection methods [10], [19]. Therefore, it is more suitable for the direct regression method. Besides, a more detailed pixel-level segmentation containing information about the targets' relative position is carried out based on text segmentation. In this case, pixels are not only classified as text or non-text but also classified as whether they belong to a particular part of text instances or not, which provides more relative position information for the network so that it improves the expressiveness of network.

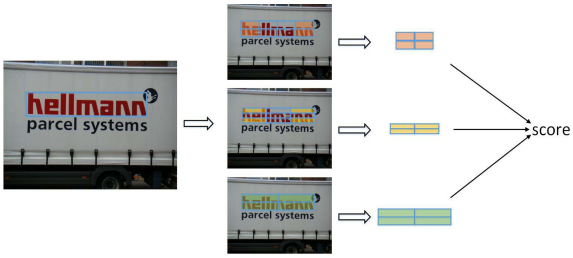


Fig. 6. Architecture of position-sensitive COI pooling. The candidate extracts the features of corresponding windows on the spatial pyramid position-sensitive segmentation maps to generate a score.

E. Post-Processing

The post-processing operation of the proposed method is as follows. First, redundant candidates generated by the direct regression module are merged and removed by the Locality-Aware NMS [12]. The merged candidate's coordinates are calculated by the weighted sum of the corresponding candidates' coordinates with corresponding score values as the weights. Then the proposed method needs the pooling operation to assemble the features of each candidate on the position-sensitive segmentation maps and obtain a score for each candidate. Finally, the score of each candidate is used to filter out false positive candidates.

For our proposed method, the position-sensitive ROI pooling operation for all the spatial pyramid position-sensitive segmentation maps will reduce the efficiency of the method seriously. Therefore, we propose the position-sensitive COI pooling instead of the position-sensitive ROI pooling to improve the efficiency of the method. In the position-sensitive COI pooling, we use only the windows near the vertices of each candidate to measure the accuracy of the candidate's location. In the position-sensitive COI pooling, we divide candidates into windows with corresponding parameters on a different set of position-sensitive segmentation maps and assemble features on the corresponding position-sensitive segmentation maps in only four windows near the vertices to score these candidates. Taking parameter 3×3 as an example, we divide candidates into 3×3 windows on the corresponding 9 position-sensitive segmentation maps. We only assemble the features and average the values of pixels in four windows near the vertices on corresponding feature maps respectively for each candidate, as shown in Figure 6. The scores of all windows near the vertices of each candidate are averaged to obtain a position-sensitive score. Finally, the candidates which have lower position-sensitive scores than a threshold will be filtered out and the remaining candidates are the final output text boxes. In the experiment, the threshold is set as 0.25.

Compared with the position-sensitive ROI pooling, the position-sensitive COI pooling can improve the efficiency of the method by reducing computational cost. Besides, it improves the performance of the method a little bit.

Because text boundary features are more obvious and the segmentation of text area is more precise for the window near the vertex than those in the internal window, they can bring a better filtering effect, if an appropriate threshold is chosen.

F. Loss Function

In the proposed method, the loss L is formulated as

$$L = L_{cof} + \lambda_{loc} L_{loc} + \lambda_{ps} L_{ps}, \tag{2}$$

where L_{cof}, L_{loc}, L_{ps} denote the loss for score map F_s, geometry maps F_g and spatial pyramid position-sensitive segmentation maps, respectively. λ_{loc} and λ_{ps} are the weights of relevant losses. In our system, we set λ_{loc} as 10 and set λ_{ps} as 1.

In our method, We use the dice loss [46] function as the loss for score map. Compared with the class-balanced cross-entropy loss [47], the dice loss is more robust to the shape of texts for segmentation because it is calculated in terms of regions rather than each pixel individually. It is defined as

$$L_{cof} = 1 - 2\frac{P^* P}{P^* + P}, \tag{3}$$

where P^* denotes the binary ground truth of the score map and $P = F_s$ denotes the predicted score map. $P^* P$ represents the sum of the products of corresponding pixels in two maps.

We adopt the loss for geometry maps L_{loc} in [12] and it is given by

$$L_{loc} = \mathrm{IOU}(R, R^*) + \lambda_\theta (1 - \cos(\theta - \theta^*)), \tag{4}$$

where $\mathrm{IOU}(R, R^*)$ denotes the IOU loss [48] between the predicted bounding box R and the ground truth R^*. λ_θ is the weight parameter and it is set to 10 in our system. θ denotes the predicted rotation angle and θ^* denotes the ground truth orientation.

The loss for position-sensitive segmentation maps L_{ps} has been described above.

IV. Experiments

To verify the effectiveness of the proposed algorithm, we conducted quantitative experiments on three public benchmark datasets: ICDAR2015 [16], MLT-17 [17] and COCO-Text [18].

A. Implementation Details

The ResNet-50 [43] and VGG-16 [49], as the backbone of the proposed network, are pre-trained on the ImageNet [50]. Data augmentation is implemented in three steps. First scales of images are resized with ratio from 0.5 to 2.0, then the heights of images are rescaled with ratio from 0.8 to 1.2 while keeping widths unchanged. Finally, we crop random regions from the transformed images and resize them to 512×512 pixels with an invariant aspect ratio by padding. The whole experiments are conducted on Tensorflow and run on a work-station with 2.8GHz CPU, 16G RAM and GTX 1080Ti. The network is trained using Adam optimizer. The learning rate starts from 10^{-4} and decays with a factor of 0.94 every 10000 steps and the max iteration is 10^5. We first pre-train our proposed network on COCO-Text datasets and then we use the augmented ICDAR2015 training images and the augmented MLT-17 to fine-tune our network respectively.

TABLE I

RESULTS OF SEVERAL METHODS ON ICDAR2015 CHALLENGE 4 INCIDENTAL SCENE TEXT LOCALIZATION TASK

Algorithm	R	P	F	FPS
EAST(VGG-16)[12]	0.73	0.80	0.76	6.5
EAST(PVANET2x)[12]	0.73	0.84	0.78	13.2
He[38]	0.80	0.82	0.81	–
Lyu[41]	0.70	0.94	0.81	–
PixelLink[6]	0.82	0.86	0.84	3.0
FOTS[8]	0.82	0.89	0.85	7.8
IncepText[10]	0.81	0.91	0.85	–
TextSnake[52]	0.80	0.85	0.83	1.1
Textspotter(det only)[53]	0.81	0.86	0.83	4.8
SPCNET[11]	0.86	0.89	0.87	–
Proposed Method(VGG-16)	0.81	0.87	0.84	6.7
Proposed Method(ResNet-50)	0.82	0.90	0.86	8.6

B. Datasets

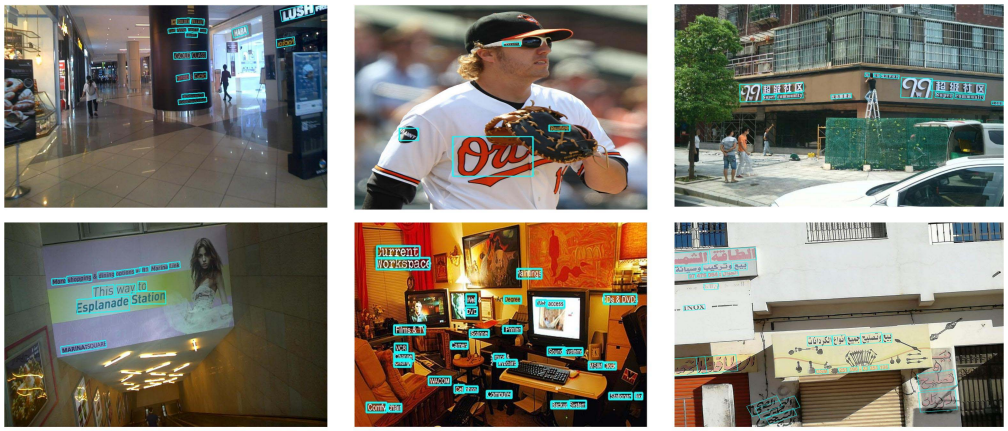
ICDAR2015 is used in the Challenge 4 of ICDAR2015 Robust Reading Competition [16]. It contains 1000 training images and 500 testing images. The scene text regions are annotated by 4 vertices of word-level irregular quadrangles. The images in this dataset are taken by Google Glasses from various scenarios in an incidental way. Therefore, it is a very challenging dataset due to motion blur, large variation in text scales, multiple orientations and complex background.

COCO-Text is the largest scene text detection dataset and it is from MS-COCO [51] dataset. The whole dataset consists of a total of 63686 images annotated with 43686 images as the training set and the other 20000 images as the test set. In this dataset, since word-level regions are annotated by horizontal rectangles, we set the angle to 0.

MLT-17 is the dataset focused on multi-oriented and multi-lingual aspects of scene text [17]. It consists of 7200 training images, 1800 validation images, and 9000 test images. In the training step, we use both training set and validation set to train our model.

C. Results on ICDAR2015

The testing images are resized with ratio 1.3 and then input into the network due to a large number of small texts. Table I lists the results of the proposed method compared with the state-of-the-art methods on ICDAR2015. The proposed method achieves an F-score of 0.86 which is comparable with the state-of-the-art methods. It is worth noting that our proposed method also has high efficiency. The IncepText [10] takes about 270ms on a Nvidia Tesla M40 GPU to process an image with the original resolution (1280 * 720) and the FPS of SPCNET [11] is around 5.0 with the resolution (1507 * 848). The proposed method runs at a speed of 8.6 FPS which is much faster than the two-stage state-of-the-art methods [10], [11].



(a) ICDAR2015 (b) COCO-Text (c) MLT-17

Fig. 7. Results of the proposed method on (a)ICDAR2015, (b)COCO-Text and (c)MLT-17.

TABLE II
RESULTS OF SEVERAL METHODS ON COCO-TEXT

Algorithm	R	P	F
Yao[26]	0.27	0.43	0.33
He[38]	0.31	0.46	0.37
WordSup[2]	0.31	0.45	0.37
EAST[12]	0.32	0.50	0.39
Lyu[41]	0.26	0.70	0.38
Proposed Method(ResNet-50)	0.39	0.61	0.47

TABLE III
RESULTS OF SEVERAL METHODS ON ICDAR2017 MLT INCIDENTAL SCENE TEXT LOCATION TASK

Algorithm	R	P	F
TH-DL[17]	0.35	0.68	0.46
He[13]	0.58	0.77	0.66
Lyu[41]	0.56	0.84	0.67
Border(ResNet-50)[7]	0.61	0.74	0.67
FOTS[8]	0.58	0.81	0.67
SPCNET[11]	0.67	0.73	0.70
Proposed Method(ResNet-50)	0.62	0.74	0.68

D. Results on COCO-Text

The testing images are resized with ratio 1.5 and input into the network without any other modification. Table II lists the results of the proposed method with state-of-the-art methods on COCO-Text. The proposed method achieves an F-score of 0.47 which is significantly higher than state-of-the-art methods.

E. Results on MLT-17

The testing images are resized with ratio 1.5 and input into the network without any other modification. Table III lists the results of the proposed method with state-of-the-art methods on MLT-17. The proposed method achieves an F-score of 0.68 which is slightly lower than Border [7] and SPCNET [11]. SPCNET is a two-stage detection method, so it has good performance but not high efficiency. Border uses bootstrapping to augment training samples, which is of great benefit to the detection of the languages other than English in this multilingual dataset.

TABLE IV
THE EFFECTIVENESS OF POSITION-SENSITIVE SEGMENTATION MODULE ON ICDAR2015

Algorithm	R	P	F	FPS
DR(baseline)	0.7849	0.8049	0.7947	11.6
DR+PS-Maps	0.8079	0.8205	0.8142	–
DR+PS-Maps+PS-ROI	0.7882	0.8689	0.8266	7.4

F. Ablation Study

We do several ablation experiments to demonstrate the effectiveness of generating ground truth of position-sensitive segmentation maps, spatial pyramid of position-sensitive segmentation and position-sensitive COI pooling. Besides, we also explore the role of pre-training. The details are discussed as follows.

1) Position-Sensitive Segmentation Module: Table IV shows the effectiveness of position-sensitive segmentation module. These experiments focus on demonstrating the effectiveness of providing ground truth for position-sensitive segmentation maps and the effectiveness of position-sensitive segmentation for filtering out false positive candidates. In these ablation experiments, spatial pyramid of position-sensitive segmentation and position-sensitive COI pooling are not added. We only use the 3×3 windows for position-sensitive segmentation and we use the position-sensitive ROI pooling for post-processing. We only train the model on ICDAR2015 without pre-training. In the Table IV, 'DR' denotes the direct regression module and it is the baseline of our proposed method. 'PS-Maps' denotes providing the ground truth of position-sensitive segmentation maps for multi-task training and 'PS-ROI' denotes using position-sensitive ROI pooling to filter out false positive candidates.

Compared with the baseline, providing the ground truth of position-sensitive segmentation maps for multi-task training achieves an improvement of 2 percents on F-score.

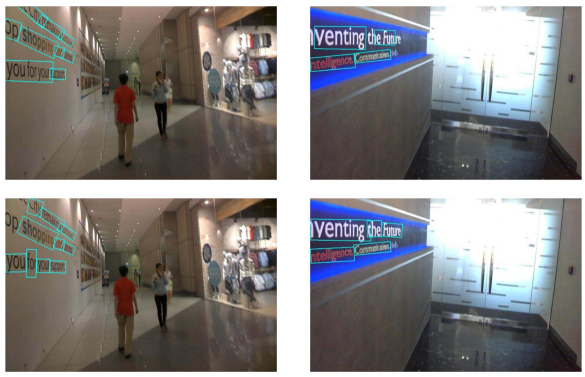


Fig. 8. Qualitative results of baseline and our method on ICDAR2015. Top: results of the baseline. Bottom: results of our method. Our proposed method performs significantly better than the baseline in the face of compact texts.

TABLE V

THE RESULTS OF THE BASELINE AND THE METHOD WITH POSITION-SENSITIVE SEGMENTATION MODULE ON ICDAR2015 UNDER DIFFERENT IOU THRESHOLD CRITERIA

IOU	Algorithm	R	P	F
0.3	DR(baseline)	0.8430	0.8647	0.8537
	DR+PS-Maps+PS-ROI	0.8339	0.9193	0.8745
0.4	DR(baseline)	0.8247	0.8459	0.8352
	DR+PS-Maps+PS-ROI	0.8185	0.9023	0.8584
0.5	DR(baseline)	0.7849	0.8049	0.7947
	DR+PS-Maps+PS-ROI	0.7882	0.8689	0.8266
0.6	DR(baseline)	0.7208	0.7393	0.7299
	DR+PS-Maps+PS-ROI	0.7323	0.8073	0.7680
0.7	DR(baseline)	0.5903	0.6054	0.5978
	DR+PS-Maps+PS-ROI	0.6206	0.6842	0.6508

This result demonstrates that providing more information about the relative position of text for the network can effectively improve the expressiveness of the network. Besides, adding the position-sensitive ROI pooling also achieves an improvement of 1.2 percents on F-score and 4.8 percents on precision. Since post-processing is added on the basis of baseline, the FPS of the proposed method is reduced by 3. This result proves the filtering effect of position-sensitive segmentation for false positive candidates. Through adding the position-sensitive segmentation module, the method achieves a total improvement of 3.2 percents on F-score and 6.4 percents on precision, which demonstrates the effectiveness of position-sensitive segmentation module for the accuracy of text detection.

Besides, the position-sensitive segmentation module can also improve the accuracy of the text location. Table V shows the performance of the baseline and the method with position-sensitive segmentation module under different IOU threshold criteria. In these experiments, we evaluate the test results of these two methods according to different IOU threshold criteria$(0.3, 0.4, 0.5, 0.6, 0.7)$ on ICDAR2015. When the IOU threshold is 0.3, the position-sensitive segmentation module increases the F-score by 2.08%. As the IOU threshold increases, the position-sensitive segmentation module increases the Recall, Precision and F-score values more. When the IOU threshold goes to 0.7, the improvement of F-score value goes up to 5.3%. The position-sensitive segmentation module enables the network to learn more accurate location information, which improves the accuracy of text position regression. Besides, the position-sensitive segmentation maps can be used to filter out some candidates which are not positioned very accurately, which also improves the performance at the high IOU threshold criteria. Therefore, the position-sensitive segmentation module can effectively improve the accuracy of the text location, which will greatly improve the performance of subsequent text recognition.

TABLE VI

THE RESULTS OF THE PROPOSED METHOD WITH DIFFERENT SPATIAL BINS OF POSITION-SENSITIVE SEGMENTATION ON ICDAR2015

Algorithm	R	P	F
3×3	0.7882	0.8689	0.8266
2×2	0.8087	0.8481	0.8280
2×4	0.7920	0.8658	0.8273
3×3, 2×4, 2×2	0.8096	0.8664	0.8371

2) Spatial Pyramid Position-Sensitive Segmentation: Table VI shows the results of the proposed method with different spatial bin of position-sensitive segmentation on ICDAR2015. In these experiments, the candidates are divided by spatial bins with different parameters and relevant position-sensitive ROI pooling are used to filter out false positive candidates. '3×3' denotes the proposed method with 3×3 position-sensitive segmentation module and the same goes for the others. For the method on the last column, we add the spatial pyramid position-sensitive segmentation into the method and use relevant position-sensitive ROI pooling for each spatial bin position-sensitive segmentation map to score candidates respectively. Then the scores of each candidate are averaged to represent the accuracy of candidate's location.

The results in these experiments demonstrate the effectiveness of spatial pyramid position-sensitive segmentation. In these experiments, the recall rate of 2×2 is higher than the recall rate of other parameters by nearly 2 percents while the precision is lower than the others by 2 percents. For a candidate, the smaller the number of windows is, the larger each window is, and the lower the location accuracy requirement for the candidate is. As the number of windows increases, the requirement becomes more stringent so that the recall rate drops while the precision rate rises. Therefore, we use the spatial pyramid position-sensitive segmentation to use different spatial bins so that we can improve the precision rate while keeping the recall rate unchanged.

3) Position-Sensitive COI Pooling: Table VII shows the effectiveness of the position-sensitive COI pooling compared with the position-sensitive ROI pooling. In these experiments, these models are first pre-trained on COCO-Text and then fine-tuned on ICDAR2015. The first line denotes the proposed method with the post-processing of position-sensitive ROI pooling and the other line denotes the proposed method with the post-processing of position-sensitive COI pooling. As can be seen from the results, the position-sensitive COI pooling improves the precision rate by 0.4 percents while reducing the recall rate by 0.3 percents and the F-score stays the same. It is

TABLE VII

THE RESULTS OF THE PROPOSED METHOD WITH POSITION-SENSITIVE ROI POOLING/POSITION-SENSITIVE COI POOLING ON ICDAR2015

Algorithm	R	P	F	FPS
Proposed method(ROI)	0.8304	0.8792	0.8541	7.4
Proposed method(COI)	0.8194	0.8986	0.8572	8.6

TABLE VIII

THE RESULTS OF DIFFERENT THRESHOLDS OF THE POSITION-SENSITIVE SCORE ON ICDAR2015

Threshold	R	P	F
0.1	0.8359	0.8727	0.8539
0.15	0.8311	0.8787	0.8542
0.2	0.8262	0.8863	0.8552
0.25	0.8194	0.8986	0.8572
0.3	0.8077	0.9060	0.8541
0.35	0.7947	0.9127	0.8496
0.4	0.7766	0.9237	0.8438

TABLE IX

THE RESULTS OF THE PROPOSED METHOD WITH/WITHOUT PRE-TRAINING ON ICDAR2015

Algorithm	R	P	F
Proposed method	0.8096	0.8664	0.8371
Proposed method+Pre-training	0.8194	0.8986	0.8572

worth noting that the position-sensitive COI pooling improves the efficiency of the method by 1.1 frames per second. Therefore, we use the position-sensitive COI pooling instead of the position-sensitive ROI pooling for the proposed method.

4) Thresholds of the Position-Sensitive Score: We conduct experiments to study the threshold effect of the position-sensitive score. Table VIII shows the results of the proposed method on ICDAR2015 with different thresholds. The results show that the recall rate decreases and the precision rate improves with the increase of threshold. The F-score dose not change much within a reasonable threshold (0.1-0.3).

5) Pre-Training: We conduct experiments to explore the role of pre-training. Table IX shows the results and the first line denotes training the model only on ICDAR2015 and the second line denotes training the model on ICDAR2015 with pre-training on COCO-Text. The results show that the pre-training effectively improve the performance of the proposed method by 2 percents on F-score.

V. CONCLUSION

In the work, we present an end-to-end method for multi-oriented scene text detection by introducing position-sensitive segmentation into the direct regression method. The proposed method involves three prediction tasks: the first classification task performs down-sampled segmentation to locate text regions, the second regression task regresses the text boxes and the third classification task performs position-sensitive segmentation of the text to achieve more refined text location. To improve the training efficiency and improve the expressiveness of the network, we generate the ground truth of position-sensitive segmentation maps to achieve synchronous training and provide more information about the relative position of text. We also introduce spatial pyramid position-sensitive segmentation considering the large differences in sizes and aspect ratios of scene texts. We also propose the position-sensitive COI pooling to solve the problem of reduced test efficiency caused by spatial pyramid position-sensitive segmentation. These improvements were demonstrated effective in experiments. The proposed method achieved state-of-the-art performance on ICDAR2015, MLT-17 and COCO-Text and it is also superior to other methods according to test efficiency. The ablation experiments show the effectiveness of position-sensitive segmentation module, the spatial pyramid of position-sensitive segmentation and the position-sensitive COI pooling.

REFERENCES

[1] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2550–2558.

[2] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4940–4949.

[3] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[4] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1962–1969.

[5] Y. Jiang *et al.*, "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: https://arxiv.org/abs/1706.09579

[6] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6773–6780.

[7] C. Xue, S. Lu, and F. Zhan, "Accurate scene text detection through border semantics awareness and bootstrapping," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 355–372.

[8] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[9] S. Zhang, Y. Liu, L. Jin, and C. Luo, "Feature enhancement network: A refined scene text detector," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2612–2619.

[10] Q. Yang *et al.*, "IncepText: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection," 2018, *arXiv:1805.01167*. [Online]. Available: https://arxiv.org/abs/1805.01167

[11] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," 2018, *arXiv:1811.08605*. [Online]. Available: https://arxiv.org/abs/1811.08605

[12] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5551–5560.

[13] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, Nov. 2018.

[14] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 534–549.

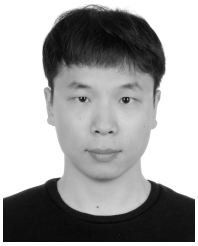
[15] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2359–2367.

[16] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, Aug. 2015, pp. 1156–1160.

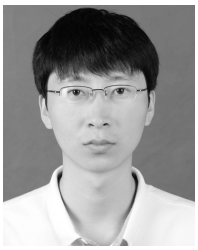
[17] N. Nayef *et al.*, "ICDAR 2017 robust reading challenge on multi-lingual scene text detection and script identification—RRC-MLT," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 1454–1459.

[18] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*. [Online]. Available: https://arxiv.org/abs/1601.07140

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
[21] P. Cheng and W. Wang, "A multi-oriented scene text detector with position-sensitive segmentation," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2018, pp. 152–159.
[22] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.
[23] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.
[24] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2558–2567.
[25] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4159–4167.
[26] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016, *arXiv:1606.09002*. [Online]. Available: https://arxiv.org/abs/1606.09002
[27] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
[28] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.
[29] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 391–405.
[30] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 128–140, Jan. 2017.
[31] Z. Zhang *et al.*, "Sequential optimization for efficient high-quality object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1209–1223, May 2018.
[32] M.-M. Cheng *et al.*, "BING: Binarized normed gradients for objectness estimation at 300fps," *Comput. Vis. Media*, vol. 5, no. 1, pp. 3–20, 2019.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[34] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3476–3483.
[35] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.
[36] Z. Shi *et al.*, "Crowd counting with deep negative correlation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5382–5390.
[37] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "Single shot text spotter with explicit alignment and attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mar. 2018, pp. 1–10.
[38] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3047–3055.
[39] S. Qin and R. Manduchi, "Cascaded segmentation-detection networks for word-level text spotting," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 1275–1282.
[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[41] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.
[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[44] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2872–2881, Jun. 2019.
[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.
[46] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2017, pp. 240–248.
[47] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1395–1403.
[48] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. ACM Multimedia Conf.*, 2016, pp. 516–520.
[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[51] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[52] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 20–36.
[53] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 67–83.



Peirui Cheng received the B.S. degree from the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, in 2013, and the M.S. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. His research interests mainly include scene text detection and object detection.



Yuanqiang Cai received the B.E. and M.E. degrees from the Xi'an University of Science and Technology, Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing. His research interests include computer vision, multimedia content analysis, and text localization in images and videos.

Weiqiang Wang received the B.E. and M.E. degrees from Harbin Engineering University in 1995 and 1998, respectively, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), China, in 2001, all in computer science. He is currently a Professor with the School of Computer and Control Engineering, University of Chinese Academy of Sciences. His research interests include multimedia content analysis, computer vision, pattern recognition, and human–computer interaction.