SCIENCE CHINA
Information Sciences

December 2019, Vol. 62 220103:1–220103:11
https://doi.org/10.1007/s11432-019-2673-8

• RESEARCH PAPER •
Special Focus on Deep Learning for Computer Vision

Irregular scene text detection via attention guided border labeling

Jie CHEN[1,2], Zhouhui LIAN[1,2*], Yizhi WANG[1,2], Yingmin TANG[1,2] & Jianguo XIAO[1,2]

[1]*Wangxuan Institute of Computer Technology, Peking University, Beijing* 100080*, China;*
[2]*Center For Chinese Font Design and Research, Peking University, Beijing* 100080*, China*

Received 20 June 2019/Revised 8 August 2019/Accepted 25 September 2019/Published online 8 November 2019

Abstract Scene text detection plays an important role in many computer vision applications. With the help of recent deep learning techniques, multi-oriented text detection that was considered to be quite challenging has been solved to some extent. However, most existing methods still perform poorly for curved text detection, mainly due to the limitation of their text representations (e.g., horizontal boxes, rotated rectangles or quadrangles). To solve this problem, we propose a novel method to detect irregular scene texts based on instance-aware segmentation. The key idea is to design an attention guided semantic segmentation model to precisely label the weighted borders of text regions. Experiments conducted on several widely-used benchmarks demonstrate that our method achieves superior results on curved text datasets (i.e., with F-score 80.1% and 78.8% for the CTW1500 and Total-Text, respectively) and obtains comparable performance on multi-oriented text datasets compared to the state-of-the-art approaches.

Keywords scene text detection, weighted border, attention mechanisms, curved text, semantic segmentation

Citation Chen J, Lian Z H, Wang Y Z, et al. Irregular scene text detection via attention guided border labeling. Sci China Inf Sci, 2019, 62(12): 220103, https://doi.org/10.1007/s11432-019-2673-8

1 Introduction

Scene text detection is used frequently in image and video retrieval, autopilot and text translation, and has received intensive attentions from researchers in areas of AI and computer vision. Owing to the variety of text size, shape, texture, and complex background, scene text detection is one of the most challenging tasks in many computer vision applications. In the last decade, large numbers of text detection methods have been proposed that rely heavily on hand-crafted features to distinguish between text and non-text regions. However, those traditional approaches require a lot of feature engineering and do not guarantee the robustness of text detection. With the help of recent deep learning techniques, a great progress has been made in scene text detection.

Generally speaking, text detection methods based on deep neural networks can be classified into two categories. The first one is based on regressing horizontal boxes, oriented rectangles or quadrilaterals by predicting the offsets from text region proposals or the corner points of text instances, such as [1–3]. As illustrated in Figure 1, when detecting irregularly shaped texts (such as curved texts), these methods predict quadrilaterals which are prone to locate excess background regions. The other type of methods is based on segmentation such as approaches proposed in [4–6], which use fully convolution networks to segment text and non-text regions. The main challenge of these methods is how to separate adjacent text regions correctly.

* Corresponding author (email: lianzhouhui@pku.edu.cn)

(a) (b) (c) (d) (e)

Figure 1 (Color online) Detection results of methods with different representations for text instances. (a) Horizontal box; (b) oriented rectangle; (c) quadrilateral; (d) simple text border; (e) ours. The proposed method is able to precisely locate arbitrary-shaped texts, while others tend to locate excess background regions.

In this paper, we propose a novel method for detecting scene texts with arbitrary shapes and orientations. The concept of text border has been reported in [7,8], but these methods fall short when separating text line regions into words and lack the capability of addressing curved text detection problem. Motivated by these studies, we develop the concept of weighted text border to deal with the challenge of separating adhesive text regions. Furthermore, we also introduce attention mechanisms [9], including channel and spatial attention modules, to our model, which effectively improve the performance of scene text detection.

To sum up, major contributions of this paper are threefold:

• First, we propose the weighted text border for separating adhesive text regions. Both qualitative and quantitative studies show its efficiency to handle adhesive text regions.

• Second, we utilize attention modules to boost the detection performance. Attention modules make our model concentrate on the text regions, which significantly improves the precision of detection results.

• Third, we implement an end-to-end trainable deep learning model achieving performance superior/comparable to other state-of-the-art approaches in scene text detection on benchmark datasets with either curved or multi-oriented texts.

2 Related work

Scene text detection has been extensively studied in recent years. Before the popularization of deep learning, a large amount of traditional methods such as stroke width transform (SWT) [10], maximally stable extremal regions (MSER) [11], have been proposed to detect scene texts by extracting text-specific features. As recent deep learning based methods markedly outperform the above-mentioned traditional approaches, here we only discuss those modern text detection methods which can be roughly classified into two categories: regression-based and segmentation-based.

Regression-based text detection methods mainly take advantage of the recent development in general object detection. WeText [12] detects characters in scene images and groups the detected characters into text lines by using the TextFlow algorithm [13] to locate scene texts. TextBoxes [14] adopts single shot multibox detector (SSD) [15] and adds anchors with large aspect ratio and specific convolution filters to fit the significant variation of aspect ratios of text instances. Rotated regional proposal network (RRPN) [16] adds rotation to both anchors and RoIPooling in Faster R-CNN [17] to cope with multi-oriented texts in natural images. Lyu et al. [18] attempted to regress four corners of text boxes, followed by a series of processes including corners grouping and non-maximum suppression (NMS), to locate multi-oriented texts accurately.

Segmentation-based text detection methods treat text detection as a semantic segmentation problem. Yao et al. [4] proposed to produce multiple score maps such as text regions and character linking orientation by taking fully convolutional neural network (FCN) as the reference framework. In [5], text blocks are predicted via FCN and character candidates are extracted using MSER. Manually designed grouping and filtering rules are used to form words and text lines. TextField [19] learns an image of two-dimensional vectors to detect irregular scene texts. To separate adjacent text lines, some studies

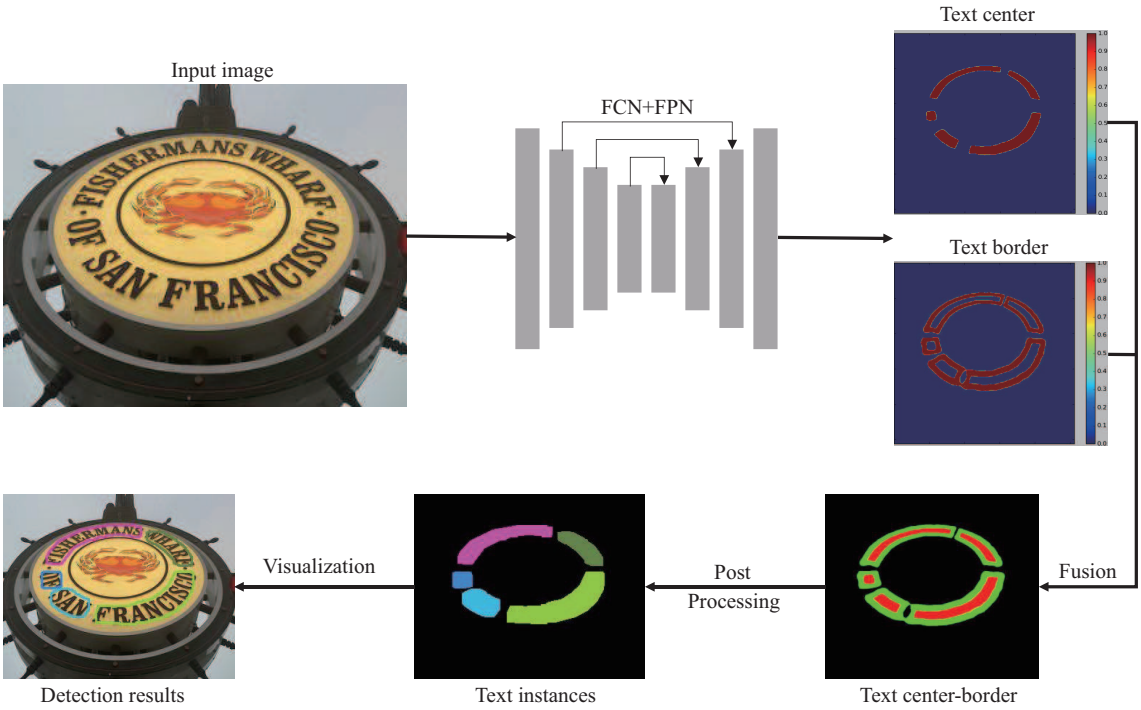

Figure 2 (Color online) Pipeline of the proposed method. Given an image, the network first outputs the text center and border maps which are then fused into one map. Based on the fused map, text instances are obtained via a simple post-processing.

such as [7, 8] introduce the border class to handle sticking text regions. Further, Xue et al. [20] divided text borders into four types with different semantics to localize scene text instances. Although these methods are effective for predicting text lines in natural images, most of them fall short when separating a text line into words.

Limited by their text representations, most above-mentioned methods perform poorly when dealing with curved texts. Motivated by [7, 8], we propose the weighted border for texts, which can fit arbitrary shapes and orientations of texts and facilitates the separation of adhesive text regions. Moreover, the attention modules adopted in our method also improve the precision of detection results.

3 The proposed method

3.1 Overview

The proposed method treats the text detection task as a text instance segmentation problem by predicting the geometric attributes of texts to precisely locate texts with arbitrary shapes and orientations. The pipeline of our method is illustrated in Figure 2. Given an image, an FCN [21] based network predicts two score maps of text center and border regions. Benefiting from the weighted text border labeling, words and text lines that stick together can be effectively separated. A simple post-processing procedure including grouping, filtering and expanding operations is applied to the above-mentioned two score maps, which eventually reconstructs the precise shapes of text instances.

3.2 Network architecture

Drawing inspiration from feature pyramid network (FPN) [22] and U-net [23], we adopt the idea of merging feature maps gradually. A schematic illustration of our model is depicted in Figure 3. VGG16 [24] is employed as the basic feature extraction network and feature maps of pooling 1–5 are used in the next stage. During feature merging, we adopt the channel and spatial attention modules before each unpooling

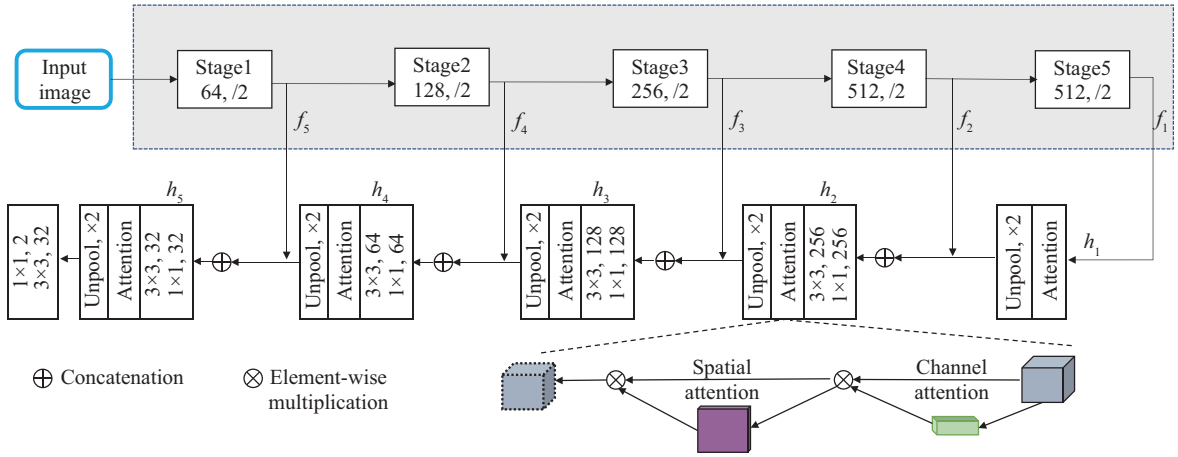

Figure 3 (Color online) Network architecture. We employ VGG16 as the backbone network and gradually merge the features of pooling 1–5. In the feature merging process, we introduce attention mechanisms before each unpooling layer.

layer. The feature merging process is defined by the following equations:

$$g_i = \text{unpool}(A_s(A_c(h_i))), \tag{1}$$

$$h_i = \begin{cases} f_i, & \text{if } i = 1, \\ \text{conv}_{3\times3}(\text{conv}_{1\times1}([g_{i-1}; f_i])), & \text{otherwise}, \end{cases} \tag{2}$$

where g_i is the merge base, f_i is the feature map of the i-th pooling layer and h_i is the merged feature map. A_s and A_c denote the spatial attention module and channel attention module, respectively. We obtain a feature map whose size is the same as the input image after feature merging. This is followed by extra 3×3 and 1×1 convolution layers, resulting a feature map with 2 channels for center and border regions of texts, respectively.

3.3 Weighted text border

Employing traditional types of text representations, most existing methods fail to precisely locate curved texts. Furthermore, the separation of adjacent text instances is also considered to be a tough task.

Previously, Wu and Natarajan [7] and Polzounov et al. [8] proposed to use text borders to separate adhesive text instances for scene text detection. Through extensive experiments, we find that if all edge pixels of a text border share the same weight, the short edges tend to be undetectable, still causing adjacent text instances to stick together (see Figure 1). As demonstrated in Figure 4, we propose the concept of weighted text border to address these problems. Unlike [7] where the text borders of training data are manually marked, we utilize the common ground-truth data provided by benchmark datasets to automatically label the weighted text borders. Specifically, vertices of each text instance are provided by the ground truth. Then, we link the vertices to produce a number of edges, which form a polygon. Next, we move each edge inside via its vertical direction by $c \times e_s$ pixels, where e_s denotes the length of the polygon's shortest edge and c is a coefficient (set as 0.3 in our experiments). All the intersection points of two previous-adjacent edges are linked together to construct a shrunk polygon inside the original polygon and the region between them is named as the text border region. Finally, we connect the corresponding vertices of the shrunk and original polygons, splitting the border region into several segments. As shown in Figure 4, we assign different weights to pixels in different segments. Specifically, in a text border, the smaller a segment is, the larger the weights of pixels inside are. The details of weight calculation are presented in Subsection 3.5.

3.4 Attention mechanisms

Attention mechanism has achieved great success in image caption, recognition and machine translation, which contributes to guiding models to focus more on important features and neglect unimportant ones.

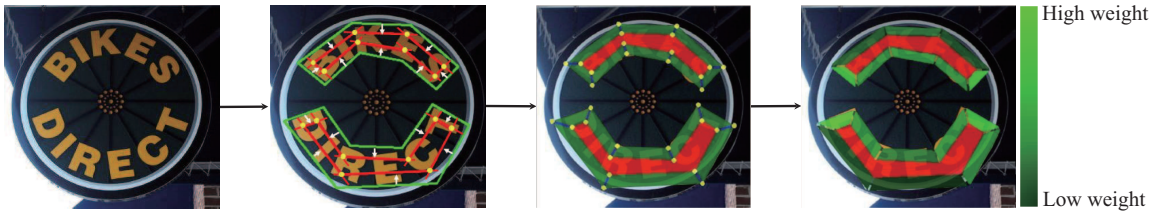

Figure 4 (Color online) An illustration of generating weighted text border.

To make the method compatible to arbitrary shapes and orientations of texts, we introduce channel and spatial attention modules [9] to our model.

Channel attention module. All channels of a feature map are treated without distinction in traditional CNNs. The channel attention mechanism attempts to reduce the interference of background by assigning larger weights to channels which have intenser response to text regions. We first apply average-pooling and max-pooling operations to the input feature map, producing two different descriptors. Then the two descriptors feed forward through a shared network consisting of multi-layer perceptron (MLP) with one hidden layer. And the hidden activation size is set to $\mathbb{R}^{C/r\times1\times1}$, where r is set to 8 as the reduction ratio in our method. Finally, we add the two output feature vectors element-wise to get the channel attention map $M_c \in \mathbb{R}^{C\times1\times1}$. The channel attention module is defined as

$$M_c(f) = \sigma(\mathrm{MLP}(\mathrm{pool}_{\mathrm{avg}}(f)) + \mathrm{MLP}(\mathrm{pool}_{\mathrm{max}}(f))), \tag{3}$$

where σ denotes the sigmoid function, $\mathrm{pool}_{\mathrm{avg}}$ and $\mathrm{pool}_{\mathrm{max}}$ are average-pooling and max-pooling operations, respectively.

Spatial attention module. In natural images, there exist many background regions diverting human attentions from text instances. Spatial attention can highlight the text regions and alleviate disturbance of background regions. First, average-pooling and max-pooling operations are applied along the channel axis to obtain two feature maps. Then we concatenate the two feature maps to form a feature descriptor. Based on the descriptor, we apply a convolution layer to generate the spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$. The spatial attention module is defined as

$$M_s(f) = \sigma(\mathrm{conv}_{7\times7}[\mathrm{pool}_{\mathrm{avg}}(f), \mathrm{pool}_{\mathrm{max}}(f)]), \tag{4}$$

where σ denotes the sigmoid function and $\mathrm{conv}_{7\times7}$ is a convolution layer with the kernel size of 7×7. Given an intermediate feature map f, the whole attention process is implemented by applying the channel attention and spatial attention modules sequentially as follows:

$$f^{\mathrm{ca}} = M_c(f) \otimes f, \tag{5}$$

$$f^{\mathrm{sca}} = M_s(f^{\mathrm{ca}}) \otimes f^{\mathrm{ca}}, \tag{6}$$

where $\otimes$ denotes the element-wise multiplication and f^{sca} is the final output feature map.

3.5 Loss function

The training loss is the sum of losses on the text center and text border:

$$L = \lambda L_{\mathrm{center}} + L_{\mathrm{border}}, \tag{7}$$

where L_{center} and L_{border} denote the losses on the center and border of text, respectively, and λ is set to 1.0 in our experiments.

Actually, the prediction of text center and border can be regarded as a pixel-wise binary classification problem. We adopt the Dice coefficient loss [25] with instance-balanced strategy to optimize the parameters of our network as follows:

$$L_{\mathrm{pixel\_dice}}(G, P, W) = 1 - 2\times\frac{|(G\cap P)W|}{|GW| + |PW|}, \tag{8}$$

$$L_{\text{center}} = L_{\text{pixel\_dice}}(G_c, P_c, W_c), \tag{9}$$

$$L_{\text{border}} = L_{\text{pixel\_dice}}(G_b, P_b, W_b), \tag{10}$$

where G, P and W denote the ground truth region, predicted region and weight map, respectively, c and b denote the center and border of text, respectively.

As the sizes of text instances may vary significantly, if all text pixels share the same weight, small text instances are not easy to be detected because they contribute little to the total loss. As described in Subsection 3.3, it is also hard to detect the short edges of a text border for the same reason. To cope with these problems, we assign specific weights to pixels in different regions. Given an image containing N text instances, the weights of pixels in the text center and border are defined as follows:

$$w_c(p) = \begin{cases} \max\left(\dfrac{\text{Area}(\mathbb{C})}{N \times \text{Area}(C_p)}, 1\right), & \text{if } p \in \mathbb{C}, \\ 1, & \text{otherwise}, \end{cases} \tag{11}$$

$$w_b(p) = \begin{cases} \max\left(\dfrac{\text{Area}(\mathbb{B})}{N \times \text{Edges}_p \times \text{Area}(S_p)}, 1\right), & \text{if } p \in \mathbb{B}, \\ 1, & \text{otherwise}, \end{cases} \tag{12}$$

where $w_c(p)$ and $w_b(p)$ denote the weights of pixel p in center and border weight maps, respectively, Area() represents the total number of pixels of the designated region, $\mathbb{C}$ and $\mathbb{B}$ are sets of pixels from text center and border regions, respectively, C_p stands for the center region containing pixel p, S_p is the segment containing pixel p and Edges_p is the number of edges in the text border containing pixel p. As described in Subsection 3.3, a text border is split into several segments.

3.6 Inference and post-processing

As shown in Figure 2, after feed-forwarding, our network outputs two score maps of text center regions (TCR) and text border regions (TBR). TCR and TBR are combined to form a new map of text center and border regions (CBR) where red and green points mean text center and border points, respectively. Based on CBR, we adopt a simple post-processing pipeline to effectively reconstruct the text instances. First, we group the connected red points on CBR to form several red regions. Next, we select valid regions from them based on the following steps. Let the total number of marginal points of a red region be N and the number of marginal points which have a green point within a circle with the radius of 3 pixels be M. If $M/N \geqslant 0.8$, the red region is a valid region. Then we merge the green points to the nearest red region. Finally, a dilation operation is employed to expand each red region to cover 90% of attached green points, which reconstructs the shapes of text instances.

4 Experiments

We evaluate our method on several widely-used benchmark datasets for scene text detection and compare it with existing methods. Some examples of our detection results are shown in Figure 5.

4.1 Benchmark datasets

SynthText [26] contains about 800000 synthetic images which are mainly annotated at word level. These images are created by blending natural images with artificial texts. This dataset is used to pre-train our model.

CTW1500 [27] consists of 1000 training images and 500 test images that contain a large number of curved text instances which are annotated at text-line level by using polygons with 14 points.

Total-Text [28] is a dataset that contains 1255 images for training and 500 images for testing. It includes multi-oriented and curved text instances which are annotated at word level.

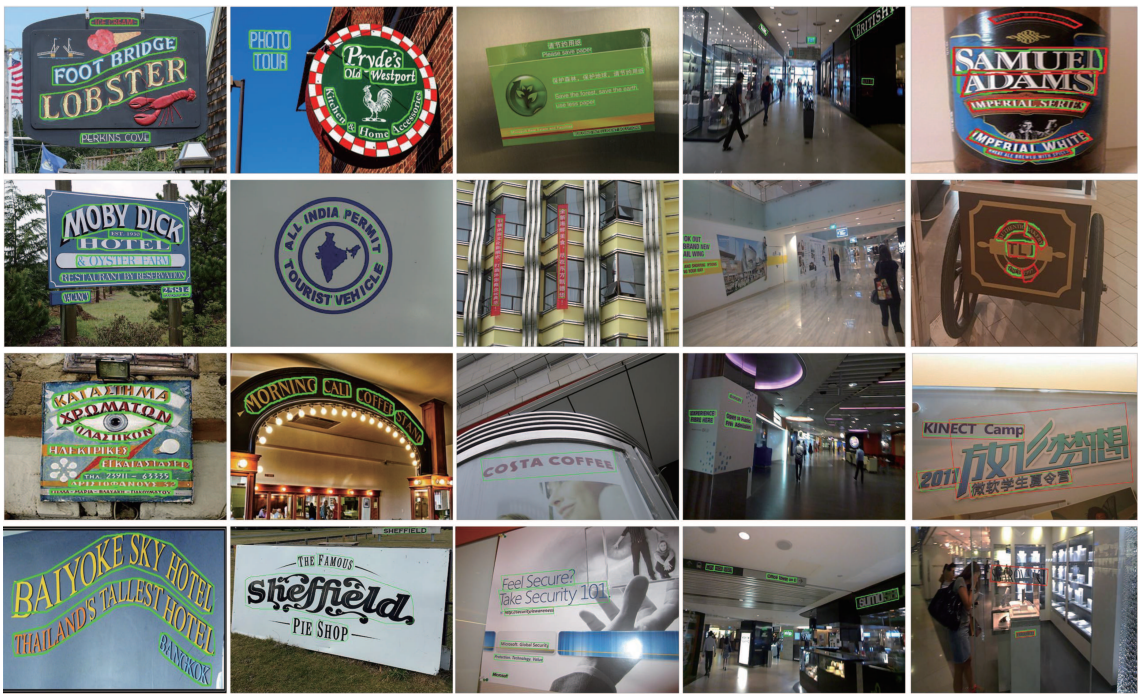

Figure 5 (Color online) Detection results of the proposed method. Sample images in column 1–4 are from CTW1500, Total-Text, MSRA-TD500 and ICDAR2015, respectively. Some failure cases are also presented in the last column, where red contours are ground truth annotations and green contours are our detection results.

ICDAR2015 [29] was used in the challenge 4 of 2015 robust reading competition. There are 1000 images for training and 500 images for testing. The text instances are annotated as quadrilaterals at word level.

MSRA-TD500 [30] is a dataset that includes 300 training images and 200 test images, and the text instances are annotated at text-line level. Similar to previous methods, we also utilize the training images of HUST-TR400 in the training phase.

4.2 Implementation details

The proposed method is implemented in Tensorflow on a regular workstation with Nvidia Geforce GTX 1080 Ti, Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz and 256 GB RAM. We train our model on 2 GPUs in parallel with the batch size of 16. The network is pretrained on SynthText for one epoch with a starting learning rate of 10^{-3}, then fine-tuned on other datasets. During the training process, we adopt the Adam optimizer [31] to optimize the network and the online hard example mining (OHEM) is also used to balance the positive and negative samples.

4.3 Experimental results

4.3.1 *Curved text detection*

We first evaluate the performance of the proposed method on two datasets: CTW1500 and Total-Text. When testing, we resize the images in the two datasets to a 512×512 box, while keeping their original aspect ratios. Tables 1 and 2 compare the proposed method with other state-of-the-art methods on CTW1500 and Total-Text, respectively. Our method achieves 80.1% F-score on CTW1500 and significantly outperforms most existing methods (except for the recently reported TextField [19]) designed for curved texts such as CTD, CTD+TLOC and TextSnake. For Total-Text, the proposed method also achieves better performance (78.8% F-score) than many other methods. As we can observe from results listed in Tables 1 and 2, our method is capable of detecting arbitrary-shaped texts at both word level and text-line level.

Table 1 Comparing text detection performance of different methods on CTW1500

Method	Recall	Precision	F-score
SegLink [1]	40.0	42.3	40.8
CTPN [2]	53.8	60.4	56.9
EAST [32]	49.1	78.7	60.4
DMPNet [33]	56.0	69.9	62.2
CTD [34]	65.2	74.3	69.5
CTD+TLOC [34]	69.8	77.4	73.4
TextSnake [35]	**85.3**	67.9	75.6
TextField [19]	79.8	83.0	**81.4**
Ours	76.6	**83.9**	80.1

Table 2 Comparing text detection performance of different methods on Total-Text

Method	Recall	Precision	F-score
SegLink [1]	23.8	30.3	26.7
EAST [32]	36.2	50.0	42.0
DeconvNet [36]	56.0	69.9	62.2
CTD+TLOC [34]	71.0	74.0	73.0
TextSnake [35]	74.5	82.7	78.4
TextField [19]	**79.9**	81.2	**80.6**
Ours	73.5	**84.9**	78.8

Table 3 Comparing text detection performance of different methods on ICDAR2015

Method	Recall	Precision	F-score	FPS
Zhang et al. [5]	43.0	70.8	53.6	0.48
CTPN [2]	51.6	74.2	60.9	7.1
Yao et al. [4]	58.7	72.3	64.8	1.61
DMPNet [33]	68.2	73.2	70.6	–
SegLink [1]	76.8	73.1	75.0	–
EAST [32]	72.8	80.5	76.4	6.52
RRPN [16]	73.0	82.0	77.0	–
WordSup [37]	77.0	79.3	78.2	2
ITN [38]	74.1	**85.7**	79.5	–
TextField [19]	80.5	84.3	82.4	5.2
TextSnake [35]	80.4	84.9	82.6	1.1
PixelLink [39]	**82.0**	85.5	**83.7**	3.0
Ours	81.0	84.3	82.6	4.1

4.3.2 *Multi-oriented text detection*

To further evaluate the effectiveness of the proposed method, we conduct experiments on ICDAR2015 and MSRA-TD500. In testing phase, images are uniformly scaled to boxes of 1280×720 and 768×768 for ICDAR2015 and MSRA-TD500, respectively. Based on the output text regions of our method, we fit a minimum bounding rectangle in our experiments. For the images of ICDAR2015 dataset, the proposed method runs at 4.1 FPS using VGG16 as the backbone network on the workstation described in Subsection 4.2. For the sake of fairness, we mainly choose the methods employing the same backbone network (i.e., VGG16) with a single scale in the testing phase for comparison. As shown in Tables 3 and 4, our method achieves comparable performance against other methods on the two datasets, demonstrating the generalization ability of the proposed method.

Table 4 Comparing text detection performance of different methods on MSRA-TD500

Method	Recall	Precision	F-score
He et al. [40]	61.0	76.0	69.0
EAST [32]	61.6	81.7	70.2
ITN [38]	65.6	80.3	72.2
RRPN [16]	68.0	82.0	74.0
Zhang et al. [5]	67.0	83.0	74.0
Yao et al. [4]	75.3	76.5	75.9
Xue et al. (ResNet) [20]	73.3	80.7	76.8
SegLink [1]	70.0	86.0	77.0
PixelLink [39]	73.2	83.0	77.8
TextSnake [35]	73.9	83.2	78.3
TextField [19]	**75.9**	**87.4**	**81.3**
Ours	72.0	86.6	78.6

Table 5 Ablation studies of our method conducted on Total-Text

Weighted border	Attention mechanisms	Recall	Precision	F-score
×	×	72.9	78.9	75.8
✓	×	73.2	82.1	77.4
×	✓	72.1	**85.3**	78.1
✓	✓	**73.5**	84.9	**78.8**

Figure 6 (Color online) Effects of the proposed weighted text border. Detection results using our method with and without weighted text border are shown in the first and second rows, respectively. Red contours are ground-truth annotations and green contours are the predicted results.

Figure 7 (Color online) Effects of the attention mechanisms. Detection results using our method with and without attention mechanisms are shown in the first and second rows, respectively. Red contours are ground-truth annotations and green contours are the predicted results.

4.3.3 *Impact of weighted text border and attention mechanisms*

We perform an ablation study over Total-Text to evaluate the effects of the weighted text border and attention mechanisms adopted in our method. Table 5 shows the results of our model with different settings on Total-Text.

Weighted text border. The weighted text border is proposed to separate text instances that lie closely to each other. As shown in Figure 6, we find that the weighted text border not only helps the separation of adhesive text instances but also prevents text instances from being cut off to some extent. It can be observed that our model is able to predict the border regions of text instances more precisely

with the weighted text border. For the proposed method with attention mechanisms, the weighted text border improves the recall by 1.4% as shown in Table 5.

Attention mechanisms. Attention mechanisms are used to guide models to focus on specific layers and regions of feature maps. As shown in Table 5, the introduction of attention mechanisms significantly improves the precision of detection results by 6.4%, enhancing the ability of our model for distinguishing between text and non-text regions. Figure 7 shows that our model with attention modules can more accurately predict text regions and exclude the interference of background regions.

5 Conclusion

We proposed a novel end-to-end approach for detecting irregular scene texts based on instance-aware segmentation. Specifically, we developed a novel concept of weighted text border to fit arbitrary shapes of texts and separate adhesive text instances. We also introduced attention mechanisms to induce our model to focus more on the text regions. The proposed method outperforms most existing approaches on curved text datasets (Total-Text and SCUT-CTW1500) and achieves competitive performances on multi-oriented datasets (ICDAR2015 and MSRA-TD500). In the future, we would like to combine our detection framework with recognition modules to develop an end-to-end recognition system for arbitrary-shaped texts in natural images.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61672056, 61672043) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

1 Shi B G, Bai X, Belongie S. Detecting oriented text in natural images by linking segments. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 2550–2558
2 Tian Z, Huang W L, He T, et al. Detecting text in natural image with connectionist text proposal network. In: Proceedings of European Conference on Computer Vision, Amsterdam, 2016. 56–72
3 Lyu P Y, Yao C, Wu W H, et al. Multi-oriented scene text detection via corner localization and region segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 7553–7563
4 Yao C, Bai X, Sang N, et al. Scene text detection via holistic, multi-channel prediction. 2016. ArXiv:1606.09002
5 Zhang Z, Zhang C Q, Shen W, et al. Multi-oriented text detection with fully convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 4159–4167
6 He D F, Yang X, Liang C, et al. Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 3519–3528
7 Wu Y, Natarajan P. Self-organized text detection with minimal post-processing via border learning. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 5000–5009
8 Polzounov A, Ablavatski A, Escalera S, et al. Wordfence: text detection in natural images with border awareness. In: Proceedings of IEEE International Conference on Image Processing, Beijing, 2017. 1222–1226
9 Woo S, Park J, Lee J Y, et al. Cbam: convolutional block attention module. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 3–19
10 Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, 2010. 2963–2970
11 Neumann L, Matas J. A method for text localization and recognition in real-world images. In: Proceedings of Asian Conference on Computer Vision, Queenstown, 2010. 770–783
12 Tian S X, Lu S J, Li C S. Wetext: scene text detection under weak supervision. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 1492–1500
13 Tian S X, Pan Y F, Huang C, et al. Text flow: a unified text detection system in natural scene images. In: Proceedings of IEEE International Conference on Computer Vision, Santiago, 2015. 4651–4659
14 Liao M H, Shi B G, Bai X, et al. Textboxes: a fast text detector with a single deep neural network. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017
15 Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: Proceedings of European Conference on Computer Vision, Amsterdam, 2016. 21–37
16 Ma J Q, Shao W Y, Ye H, et al. Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans Multimedia, 2018, 20: 3111–3122
17 Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems, Palais, 2015. 91–99

18 Lyu P Y, Yao C, Wu W H, et al. Multi-oriented scene text detection via corner localization and region segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 7553–7563

19 Xu Y C, Wang Y K, Zhou W, et al. TextField: learning a deep direction field for irregular scene text detection. IEEE Trans Image Process, 2019, 28: 5566–5579

20 Xue C H, Lu S J, Zhan F N. Accurate scene text detection through border semantics awareness and bootstrapping. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 355–372

21 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, 2015. 3431–3440

22 Lin T Y, Dollar P, Girshick R, et al. Feature pyramid networks for object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 2117–2125

23 Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, 2015. 234–241

24 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556

25 Milletari F, Navab N, Ahmadi S A. V-net: fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 4th International Conference on 3D Vision (3DV), California, 2016. 565–571

26 Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 2315–2324

27 Yuliang L, Lianwen J, Shuaitao Z, et al. Detecting curve text in the wild: new dataset and new solution. 2017. ArXiv:1712.02170

28 Ch'ng C K, Chan C S. Total-text: a comprehensive dataset for scene text detection and recognition. In: Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017. 935–942

29 Karatzas D, Gomez-Bigorda L, Nicolaou A, et al. ICDAR 2015 competition on robust reading. In: Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, 2015. 1156–1160

30 Yao C, Bai X, Liu W Y, et al. Detecting texts of arbitrary orientations in natural images. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2012. 1083–1090

31 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980

32 Zhou X Y, Yao C, Wen H, et al. EAST: an efficient and accurate scene text detector. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 5551–5560

33 Liu Y L, Jin L W. Deep matching prior network: toward tighter multi-oriented text detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 1962–1969

34 Liu Y L, Jin L W, Zhang S T, et al. Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recogn, 2019, 90: 337–345

35 Long S B, Ruan J Q, Zhang W J, et al. Textsnake: a flexible representation for detecting text of arbitrary shapes. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 20–36

36 Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. In: Proceedings of IEEE International Conference on Computer Vision, Santiago, 2015. 1520–1528

37 Hu H, Zhang C Q, Luo Y X, et al. Wordsup: exploiting word annotations for character based text detection. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 4940–4949

38 Wang F F, Zhao L M, Li X, et al. Geometry-aware scene text detection with instance transformation network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 1381–1389

39 Deng D, Liu H F, Li X L, et al. Pixellink: detecting scene text via instance segmentation. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018

40 He W H, Zhang X Y, Yin F, et al. Deep direct regression for multi-oriented scene text detection. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 745–753