Scene text detection using HRNet and spatial attention mechanism

Qingsong Tang (tangqs@mail.neu.edu.cn)
Northeastern University
Zhangyan Jiang
Northeastern University
Bolin Pan
Northeastern University
Jinting Guo
Northeastern University
Wuming Jiang
Beijing Eyecool Technology Co., Ltd

Research Article

Keywords: Text region attention, High resolution networks, Scene text detection, Deep learning

Posted Date: August 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1896482/v2

Scene text detection using HRNet and spatial attention mechanism

Qingsong Tang[1]· Zhangyan Jiang[1]·Bolin Pan[1]· Jinting Guo[1]·Wuming Jiang[2]

Abstract To better extract the features from text instances with various shapes, a scene text detector using High Resolution Net (HRNet) and spatial attention mechanism is proposed in this paper. Specifically, we use HRNetV2-W18 as the backbone network to extract the text feature in text instances with complex shapes. Considering that the scene text instance is usually small, to avoid too small feature size, we optimize HRNet through deformable convolution and Smooth Maximum Unit (SMU) activation function, so that the network can retain more detail information and location information of the text instance. In addition, a Text Region Attention Model (TRAM) is added after the backbone to make it pay more attention to the text location information and a loss function is added to TRAM, so that the network can learn the features better. The experimental results illustrate that the proposed method can compete with the state-of-the-art methods. Code is available at: https://github.com/zhangyan1005/HR-DBNet.

Keywords Text region attention· High resolution networks· Scene text detection· Deep learning

1 Introduction

As the most critical part of scene text recognition, scene text detection refers to the extraction of the text regions from a natural scene image. In recent years, scene text detection and recognition has attracted increased attention due to its wide applications in scene understanding, blind guidance and autonomous driving, etc. Scene text detection is a very challenging task in computer vision. The main difficulties are as follows: (1) There may contain a variety of languages, fonts, shapes, sizes and directions in natural scene text. (2) The defocusing, blurring and degradation arise during image data collection make the basic tasks such as segmentation difficult. (3) The background may be very complex and the signs in the background could be like the text which leads to false detection.

With the tremendous progress in deep learning, there have emerged numerous methods based on convolutional neural networks (CNNs) in scene text detection. These methods can be roughly divided into two categories: regression-based methods and segmentation-based methods.

Regression-based method Such methods first obtain feature maps by the backbone, then predict whether each pixel on the feature maps belongs to a text instance and finally obtain the text boxes by means of the non-maximum-suppression. For example, by modifying the region suggestion and boundary box regression modules of the general detector Faster R-CNN[6] and SSD[7], a scene text detection algorithm is designed to locate text instances directly. By improving the Faster R-CNN, CTPN[8] can detect horizontal words. Poly-

✉ Qingsong Tang
E-mail: tangqs@mail.neu.edu.cn
Zhangyan Jiang
E-mail: 1714203863@qq.com
Bolin Pan
E-mail: 2100126@stu.neu.edu.cn
Jinting Guo
E-mail: 2100107@stu.neu.edu.cn
Wuming Jiang
E-mail: jiangwuming@eyecool.cn
[1] Department of Mathematics, College of Sciences, Northeastern University, Shenyang, Liaoning, 110819, China
[2] Beijing Eyecool Technology Co., Ltd, Beijing, 100089, China

FRCNN[9] can detect bent text. Textboxes[10] and Textboxes++[11] tweaked SSD by defining the default text box as a quadrangle with different width ratio specifications to accommodate different orientations and width ratios of text. Jaderberg et al.[12] used Edge Boxes[13] to generate candidate boxes and then used regression to fine-tune the candidate Boxes. Dai et al.[14] proposed a Progressive Contour Regression (PCR) method for detecting text boxes with arbitrary shapes. PCR generates horizontal text suggestions by estimating the center and size of the text. Then, the text suggestion-oriented corners are predicted from the initial horizontal corners, and finally the text box of arbitrary shape is returned by iteration [14]. Regression-based methods rely heavily on complex heuristic processing, which wastes a lot of computing resources.

Segmentation-based method Such methods use segmentation to obtain text instance directly without regression operation. The boundary learning method[15] divides each pixel into three categories: text, boundary and background[15]. PSENet[1] obtained multiple mask kernel regions with the same center points and different proportions through instance segmentation based on the boundary learning method, and obtained text instance prediction text regions by using progressive scale extension algorithm. In the segmentation framework, Tian et al.[16] added a loss term to maximize the Euclidean distance between pixel embedding vectors belonging to different text instances and pixel embedding vectors belonging to the same instance, so as to better separate adjacent texts[16]. Lyu et al.[17] proposed a Mask TextSpotter that uses character-level tags to detect and recognize both character and instance masks. PixelLink[2] predicts whether two adjacent pixels belong to the same text instance by adding additional output channels to represent links between adjacent pixels. DBNet[3] proposes the differentiable binarization (DB) which makes the process of binarization end-to-end trainable. This simplifies the post-processing steps and greatly saves the time cost.

Due to the capacity of detection arbitrary shapes of scene text and the robustness in practical applications, scene text detection methods based on segmentation have attracted more and more attention in recent years. Accurate feature extraction is very crucial to segmentation-based methods. The previously mentioned segmentation-based methods obtain good performance through label making and post-processing, but few attentions are paid to the backbone networks. ResNet[19] and VGG[20] are usually used as the backbone network to extract features in text detection, and then a structure like feature pyramid (FPN)[21] is built to perform feature fusion. For the feature obtained by networks through continuous stridden convolution or pooling, although the high-resolution representation is obtained by up-sampling, some spatial information is lost since up-sampling cannot make up for the loss of spatial resolution.

In HRNetV2[5] network, the representation of high resolution is always maintained, and then the low resolution is continuously added. HRNet can adapt to complex changes in human posture, so it is often used for human body posture estimation. Considering the shapes of the text are also complex and various, we use HRNet as the backbone to better extract the text feature in text instance with various shapes.

The network usually performs a series of convolution operations with strides to obtain high resolution semantic information. During this process, the network loses some location information, which may affect the final detection results. We propose a text region attention module (TRAM) added after the backbone to compensate for the location information lost in downsampling.

Our main contributions are as follows:

1. We verifies that HRNetV2-W18 network can be used for scene text detection with good results.
2. We propose a text region attention module and improve the DB module to better capture text area

information.

3. We achieve the precision of 87.6%, recall of 79.6% and F-measure of 83.4% on the data set ICDAR2015 without additional data set pre-training. On the data set Total-Text, the precision recall and F-measure reach 85.7%, 77.6% and 81.5%, respectively.

4. We pretrain 300 epochs on ICDAR2017, iterating about 541K times, and then fine-tune the data sets on different data sets (ICDAR2015, Total-Text, MSRA-TD500 and CTW1500) to obtain a better result. The F-measures reach 85.7%, 84.4%, 85.0% and 83.4%, respectively.

The rest of this article is organized as following. We discuss related work in Sec. 2. In Sec. 3, we explain the influence of attention mechanism and activation function on text detection. In Sec. 4, we present the experimental results and compare them with the results of previous methods. We present the conclusions in Sec. 5.

2 Related work

We use HRNet as the backbone and DBNet in the image segmentation process. We introduce them briefly.

2.1 HRNet

HRNet[5] is usually used for human posture estimation which performs well in keypoints detection, posture estimation and multi person posture estimation. It is composed of parallel high-to-low resolution subnetworks with repeated information exchange across multi-resolution subnetworks. While increasing the depth, HRNet still retains the high-resolution feature map, and adds new branches to increase the number of channels to obtain more channel information. Each new branch is formed by 3×3 convolutions with step size of 2. The output size is 1/2 of the input size and the number of channels is doubled. Then the feature maps with different resolutions are fused in the feature fusion stage. Compared with the first block of VGG and ResNet, HRNet uses two 3×3 convolutions to replace the 7×7 convolutions to reduce the number of parameters while keeping the size of receptive field.

2.2. DBNet

The segmentation-based scene text detection methods need to predict whether each pixel belongs to the text instance. The general method is to select a threshold, and then the pixel with predicted probability greater than the threshold is classified as the text area. Usually, this binarization process can be described as follows:

$$B_{i,j} = \begin{cases} 1, & \text{if } S_{ij} \geq t, \\ 0, & \text{Otherwise,} \end{cases} \tag{2.1}$$

where S is the probability map produced by a segmentation network and B is the binary map. t is the predefined threshold and (i, j) indicates the coordinate point in the map. The binarization described in Eq. (2.1) is not differentiable. Therefore, the hyper-parameter t cannot be optimized by the network. To solve this problem, DBNet[3] proposes to approximate the standard binary function by a differentiable function:

$$\tilde{B}_{i,j} = \frac{1}{1+e^{-k\left(S_{i,j}-T_{i,j}\right)}}, \tag{2.2}$$

where $\tilde{B}$ is the approximate binary map, T is the adaptive threshold map learned from the network and k indicates the amplifying factor. Since Eq. (2.2) is differentiable, the network can optimize the threshold T to improve the performance of segmentation.

3 Methodology

The overall architecture of the proposed method is shown in Fig. 3.1. It includes three modules: feature extraction module, text attention module and DB module. First, the input image is fed into HRNetV2-W18 to obtain the feature map. Then, the feature map and att_text_ map is fused by dot product to achieve the new feature map F. Here the att_text_ map is generated by real text

labels. The detail for the generation method of this label will be given in Sec. 3.1. Finally, the new feature *F* is used to predict the segmentation map(*S*) and Threshold map(*T*), and then the approximate binary map is obtained by differentiable binarization function from the prediction map *S* and threshold map *T*. In the inference stage, text boxes can be obtained through post-processing operations.

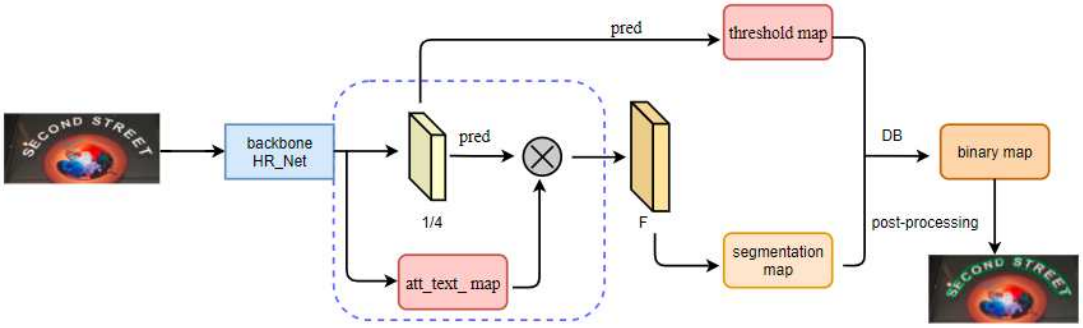


Fig. 3.1 The model framework. “DB” denotes differentiable binarization.

3.1 The label generation

To train the network, we need to generate four maps: att_text_map, segmentation map, threshold_map and approximate binary map.

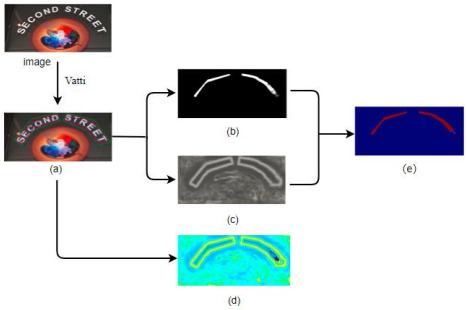


Fig. 3.2 The production of the labels. (a) different text borders: real label text borders (red), shrunken text borders(blue) and expanded text borders(green). (b) segmentation map, (c) threshold_map. (d) att_text_map. (e) approximate binary maps.

As shown in Fig. 3.2, the methods to generate the segmentation map, threshold_map and approximate binary map are consistent with those in DBNet[3]. The detail to generate att_text_ map is shown in Table 3.1, where *Gs* denotes the text bounding box shrunk by the ground truth, *Gd* represents the text bounding box dilated by the ground truth and [*Gs*, *Gd*] represents the gap between *Gs* and *Gd*. We use $|\sigma|$ to represent the normalized distance from each pixel to the closest segment in the ground-truth bounding box. The segmentation map and approximate binary map are denoted by *S* and *B,* respectively.

Table 3.1 The production of the label

Maps	<Gs	[Gs, Gd]	> Gd
S	1	0	0
Threshold_map	0	$\|\sigma\|$	0
B		Eq. (2.2)	
att_text_ map	0.3	$\|\sigma\|$	0.3

3.2 Text region attention

Inspired by the spatial attention mechanism[18], the TRAM module is proposed to compensate for the location information lost in downsampling. The TRAM module (shown in Fig. 3.3) is constructed by combining convolution operations with att_text_map, so that the high-resolution semantic information can be obtained by convolution operations while the location information can be learned through the att_text_map feature. Therefore, the attention module of the text area can help the network for information transmission by learning information

that needs to be emphasized or suppressed.

TRAM uses the spatial position relationship of features to generate text area attention, which is different from the channel attention. It focuses more on the spatial position of the text region, and so the network can understand which region of the features should have higher response. As shown in Fig. 3.3 (b), first, convolution operation is used to downsample the input feature image to obtain a new feature image with the size of 1/4 of the original image and the number of channels is not changed. Then, a feature map of size $C*H*W$ is obtained by SMU activation and two 2×2 deconvolution operations with a stride of 2. Finally, the feature is multiplied with att_text_map element by element, and activated by sigmoid to obtain the feature map with attention.

Formally, given an intermediate feature map $F \in R^{C\times H\times W}$ as input, the model predicts a two-dimensional spatial attention map $att\_text\_map \in R^{1\times H\times W}$ and a new feature map $F' \in R^{c\times H\times W}$. The whole process can be summarized as follows:

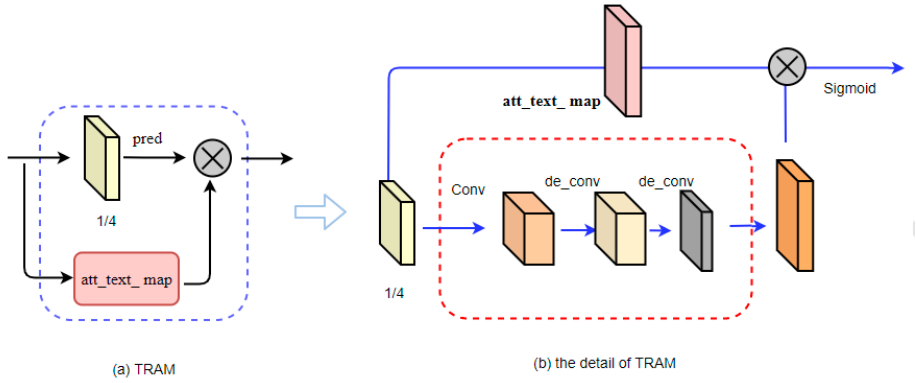


Fig. 3.3 Text area attention module, (a)TRAM structure in blue box as shown in Fig3.1. (b) details of text region attention model.

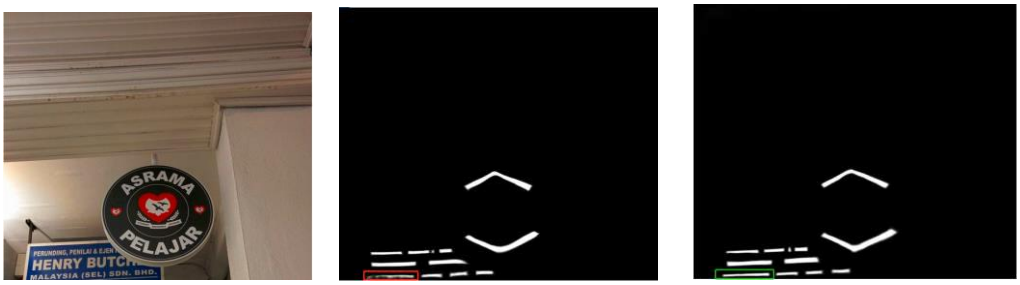
$$F^{'} = deconv_{2\times2}\left(SMU\left(deconv_{2\times2}\left(SMU\left(conv_{3\times3}\left(F\right)\right)\right)\right)\right), \quad (3.1)$$

where $def_{l\times l}$ represents the deconvolution operation of size $l\times l$ and SMU denotes the sigmoid activate function. Then, the output of the TRAM module can be obtained as

$$F_{out} = \sigma\left(att\_text\_map \otimes F^{'}\right), \quad (3.2)$$

where $\otimes$ represents the pointwise product.

In the module of TRAM, att_text_map generated by labels plays a role of like spatial attention which makes the network backbone pay more attention to the features of text area to emphasize the area containing text and suppress information of other locations. Fig. 3.4 shows the heatmaps of the three images from Total-Text. The first column represents the original images. The second column represents the heatmap of DBNet. The third column represents the heatmap of DBNet with TRAM. We can see that the heatmap is clearer and is better fitting for the text instance after adding TRAM.



(a) Original Image (b) DBNet (c) our

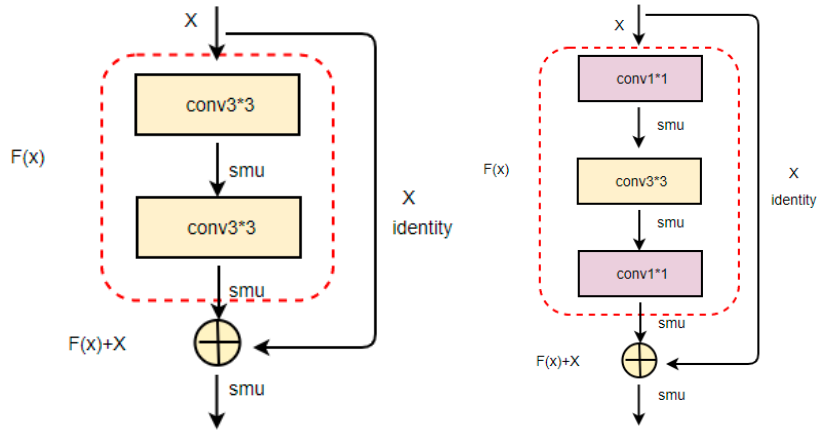
Fig. 3.4 Heatmaps of TRAM_DBNet and DBNet detection results in Total-Text

3.3 Backbone

We choose HRNetV2-W18 as the backbone network and optimize it through deformable convolution and SMU activation function, so that the network can retain more detail information and location information of the text instance.

3.3.1 Residual blocks and bottleneck blocks

The network structure of HRNetV2-W18 is consistent with the basic architecture in [5], which is mainly composed of residual blocks and bottlenecks as in ResNet50[19]. It can be roughly divided into four stages. The first stage is composed of four bottleneck blocks with 64 output channels, and each stage has one more branch than the previous stage. Each new branch is the result of convolution operation and fusion of all feature maps of the previous stage, in which the resolution is half of that of the previous branch and the number of channels is twice of that of the previous branch to fuse feature and repeat exchange of information. The structure of residual block and bottleneck block modified in are shown in Fig. 3.5, where SMU represents the smooth maximum unit which will be defined in Sec. 3.3.2.



(a) basic block (b) bottleneck

Fig. 3.5 Improved basic block and bottleneck. (a) basic block. (b) bottleneck.

3.3.2 Activation function

We select Smooth Maximum Unit (SMU)[22] as the activation function which is defined as

$$f(x,\alpha x:\beta)=\frac{(1+\alpha)x+(1-\alpha)xerf\left(\beta(1-\alpha)x\right)}{2}. \quad (3.3)$$

Here $erf(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$. α and β are hyperparameters, which are set to 0.25 and 1000000, respectively.

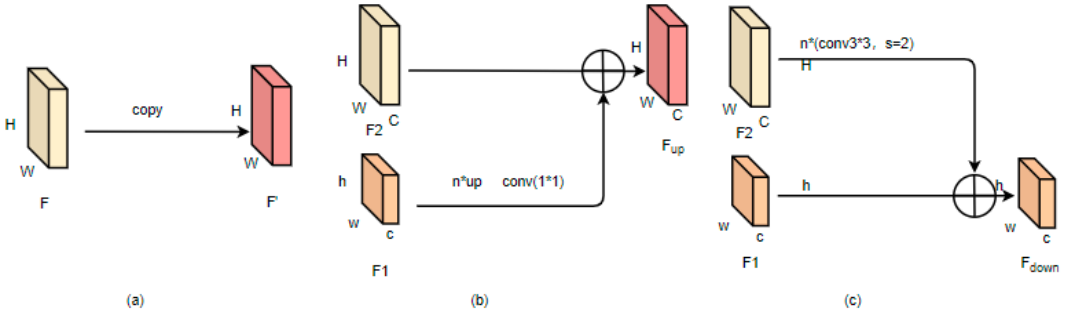


Fig. 3.6 HRNetV2 W18 feature fusion in different ways. (a) The first case: fusion between feature maps of the same resolution. (b) The second case: fusion from low-resolution feature images to high-resolution feature images. (c) The third case: fusion from high resolution feature images to low resolution feature images. The first case is included in both (b) and (c).

3.3.3 Feature fusion

As mentioned above, HRNetV2-18 consists of four stages. The second, third and fourth stages have different branches, which requires the fusion of features obtained from different branches with different resolutions. Fig. 3.6 includes three situations: fusion between feature map of the same resolution, fusion from low-resolution feature map to high-resolution map, and fusion from high-resolution feature map to low-resolution feature map. In the first case, the fusion method is to copy the input feature map directly. In the second case, bilinear interpolation is used to up-sample the target size and convolution operation with convolution kernel size of 1×1 is used to keep the number of channels. In the last case, 3×3 convolution operation with stride 2 is

used to downsample to the target size.

The feature fusion process can be summarized as Eq. (3.4), (3.5) and (3.6):

$$F' = F \tag{3.4}$$

$$F_{\text{up}} = F_2 \oplus f_{1\times1}(n \times up(F_1)) \tag{3.5}$$

$$F_{down} = F_1 \oplus (n \times f_{3\times3}(F_2)) \tag{3.6}$$

where $f_{1\times1}$ and $f_{3\times3}$ represent convolution operation with 1×1 and 3×3 kernel, respectively. F_1 and F_2 represent the input features with different resolution. $\oplus$ represents addition element by element. n denotes the number of the operation. F_{up} and F_{down} represent the fusion from low resolution to high resolution and the fusion from high resolution to low resolution, respectively.

3.3.4 The output form of the backbone

As shown in Fig. 3.7, the number of output channels of HRNetV2-W18 are 18,36, 72 and 144, respectively. The output can be summarized as

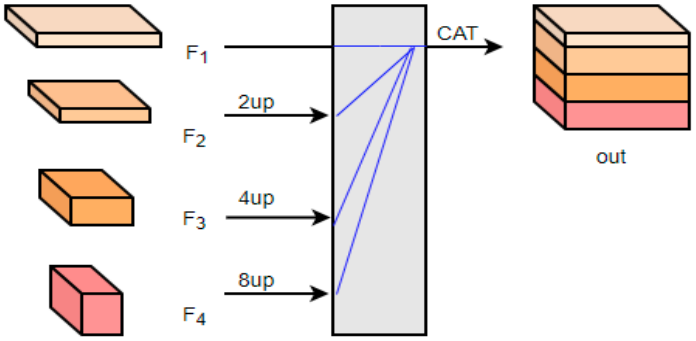


Fig. 3.7 Output structure of HRNet V2

$$2\times f_{1\times1}\left(cat\left(F_1;2\cdot up\left(F_2\right);4\cdot up\left(F_3\right);8\cdot up\left(F_4\right)\right)\right), \tag{3.7}$$

where $4\cdot up$ and $8\cdot up$ represent 4 times up-sampling and 8 times up-sampling, respectively. CAT represents concatenation operation. F_1, F_2, F_3, and F_4 represent the input features with different resolution, respectively.

The number of the output channel in the last layer is 19 in the original version of HRNetV2. We use two convolution operations of size 1×1 to increase it to 256 to obtain more channel information.

3.4 Loss function

Denote the loss on the TRAM, threshold map, approximate binary map and segmentation prediction map in the text area as L_{tram}, L_{thre}, L_B and L_S, respectively. The total loss function is defined as

$$L = \alpha \times \left(L_{tram} + L_{thre}\right) + \beta \times \left(L_B + L_S\right), \tag{3.8}$$

where $\alpha = 10$ and $\beta = 5$. The cross-entropy loss is adopted for L_B and L_S. And the hard case mining is used to balance positive and negative samples. The cross-entropy loss is defined as

$$\sum_{i\in S_l} y_i \log x_i + \left(1-y_i\right)\log\left(1-x_i\right), \tag{3.9}$$

where S_l represents the sampled data set, and the ratio of positive and negative samples is set to 1:3. x_i denotes the pixel value of sampling points in the segmentation probability map or approximate binary graph output by the network and y_i represents the label for x_i. L_1 loss is adopted for L_{tram} and L_{thre} as

$$L_{tram} = L_{thre} = \sum_{i\in R}\left|y_i^* - x_i^*\right|, \tag{3.10}$$

where R is the set of pixels after attention expansion in an expanded polygon or text region. y_i^* denotes the label of the adaptive binarization threshold map or the label of the att_text_map and x_i^* denotes the feature value of network output.

4 Experiments and results

4.1 Datasets

ICDAR2015[24] contains 1500 images, including 1000 training images and 500 test images, respectively. It contains 17,548 instances of English text. The label information of the text includes the coordinates of the four points surrounding the boundary box of the text (clockwise) and the content of the text. In addition,

"###" is used to indicate unattended text content.

ICDAR2017[25] contains a total of 9000 images, including 7200 training images and 1800 test images, respectively. It mainly includes horizontal, vertical and slanted text and most of them are horizontal and long text. The text label information of ICDAR2017 dataset contains four corner points of the text boxes and text contents.

Total-Text[26] contains 1555 images, including 1255 training images and 300 test images, respectively. The Total-Text dataset is characterized by various shapes of text, including horizontal, multi-directional and curved text.

MSRA-TD500[27] contains a total of 500 images, including 300 training images and 200 test images. The TD500 dataset contains both English and Chinese texts with multi-direction. Text labels are annotated at the line level. Due to the lack of training sets in this data set, we also add additional data set HUST-TR400[28] in the training stage.

CTW1500[29] contains 1500 images in total, including 1000 training images and 500 test images. Each text label is given by 32 coordinate values, the first four being the coordinates of the "top left, bottom right" vertices of the rectangular box, and the remaining 28 values representing the polygonal box coordinates of the curved text.

4.2 Data augmentation

To increase the generalization of the model, we adopt data enhancement to enlarge the training sets. Our data augmentation mainly includes the following operations: (1) Random rotation with an angle range of $(-10^{\circ},10^{\circ})$; (2) Random clipping. All the images are resized to 640×640 randomly; (3) Random rotation in horizontal or vertical direction.

4.3 Implementation details

All experiments are performed under the environment of PyTorch 1.10.0 and Python3.7.10. The details of the hyperparameters are shown in Table 4.1. The learning rate per iteration is calculated as

$$l_r = init\_l_r \times \left(1 - \frac{epoch}{\max\_epoch + 1}\right)^{power} . \quad (4.1)$$

Here $init\_l_r$ represents the initial learning rate and we set it to 0.007. l_r represents current learning rate and $\max\_epoch$ represents the maximum number of epochs in the training period. *power* is set to 0.9.

Table 4.1 Experimental hyperparameter setting

Names	parameter
Batch_size	4
The number epochs	1200
Initial learning rate	0.007
The learning rate per iteration	Eq. (4.1)
Gradient descent method	stochastic gradient descent (SGD)
SGD momentum	0.9
SGD weight_decay	0.0001

As usual, we use Precision(P), Recall(R) and F-measure(F)[30] to evaluate the performance of the model.

4.4 Experiment results

The experiments include three parts. In the first part, we conduct some ablation experiments on ICDAR2015. In the second part, we compare the proposed model with other methods. In the third part, we present some visualization analysis.

4.4.1 Ablation experiments

In this section, we conduct some ablation experiments on ICDAR2015 to verify the effectiveness of different backbone networks, activation function and the proposed spatial attention mechanism (i.e. TRAM). The experiment results are shown in Table 4.2.

The effectiveness of HRNetV2-W18 as the backbone network Since ResNet is the most common backbone network in scene text detection, we compare HRNetV2-W18 with ResNet18 and ResNet50 to verify the effectiveness of HRNetV2-W18 as the backbone

network. As shown in Table 4.2, DBNet with HRNetV2-W18 as the backbone network achieves better performance than ResNet18 and

Table 4.2 Ablation results of ATT DBNet at ICDAR2015

Method	P (%)	R (%)	F (%)	#Par(M)	FPS
ResNet18+Relu	86.1	74.2	79.8	51.11	24.57
ResNet18+SMU	**88.0**	74.0	80.5	51.14	**26.70**
ResNet18+ Relu+TRAM	87.5	72.0	79.0	52.77	26.56
ResNet18+SMU+TRAM	85.7	**77.6**	**81.4**	51.14	23.76
ResNet50+Relu	87.0	74.5	80.2	107.09	15.94
ResNet50+SMU	85.9	76.9	81.2	107.09	**20.63**
Resnet50+ Relu+TRAM	**86.3**	75.0	80.2	107.08	20.15
Resnet50+SMU+TRAM	85.9	**77.7**	**81.6**	107.09	15.82
HRNetV2-W18+Relu	85.9	76.2	80.7	42.03	11.50
HRNetV2-W18+SMU	**87.8**	77.8	82.5	42.03	**17.76**
HRNetV2-W18+Relu+TRAM	86.9	79.3	82.9	42.11	17.69
HRNetV2-W18+SMU+TRAM	87.6	**79.6**	**83.4**	**40.84**	12.26

ResNet50. Specifically, compared to DBNet with ResNet50 and ResNet18 as backbone network, the F-measure increases by 0.5% and 0.9%, respectively. Moreover, the number of parameters in DBNet with HRNetV2-W18 as the backbone network is less than half of that with ResNet50 as the backbone network.

The effectiveness of SMU activation function In Table 4.2, we can see that using SMU as activation function also achieves better performance than Relu. For ResNet18 backbone, although the Recall decreases by 0.2%, the Precision and F-measure increases by 1.9% and 0.7%, respectively. For ResNet50 backbone network, the Precision decreases 1.1%, but the Recall and the F-measure increases by 2.4% and 1.0%, respectively. For HRNetV2-W18 backbone, the Precision, Recall and F-measure increases by %1.9, 1.6% and 1.8%, respectively.

The effectiveness of TRAM Table 4.2 shows that TRAM cannot improve DBNet with Resnet them on corresponding data set. #Par represents the number of arguments.

Detection of long text ICDAR2015 composed mainly long text images. As shown in Table 4.3 and Fig 4.1, we can see that our method achieves higher F-measure than DBNet (85.7% vs 85.4%). Compared with other methods, although the speed is not the fastest, our model achieves the highest accuracy with the smallest number of parameters.

backbone, but it improves DBNet with HRNetV2-W18 backbone significantly. The Precision, Recall and F-measure are increased by 1.0%, 3.1% and 2.2%, respectively when TRAM is added to DBNet with HRNetV2-W18 as bone network and Relu as activation function. The Precision decreased by 1.0%, but the Recall and F-measure are increased by 0.3% and 0.5%, respectively when TRAM is added to DBNet with HRNetV2-W18 as bone network and SMU activation function.

In general, the HRNetV2-W18 backbone network, SMU activation and TRAM improve the performance of text detection on ICDAR2015.

4.4.2 Comparisons with other methods

In this section, we compare the proposed model with prior methods. In this series of experiments, we first pre-training the models on ICDAR2017 for 300 epochs and then fine-tune

Curved text Total-Text dataset and CTW1500 composed of mainly multi-directional and bent text. Almost every image contains an instance of bent text. As can be seen from in Table 4.4 and

Table 4.3 Detection results on ICDAR2015

Method	P(%)	R(%)	F (%)	#Par(M)	FPS
CTPN[8]	74.2	51.6	60.9	-	7.1
EAST[31]	83.6	73.5	78.2	-	13.2
FCENet[32]	85.1	84.2	84.6	-	-
TextSnake[33]	84.9	80.4	82.6	218.9	1.1
PixelLink[2]	85.5	82.0	83.7	234.9	3.0
SegLink[34]	73.1	76.8	75.0	170.0	-
PSENet[1]	86.9	**84.5**	**85.7**	229.3	1.6
DBNet[3]	88.2	82.7	85.4	110.4	26
Ours	**88.5**	83.0	**85.7**	**42.3**	8.2

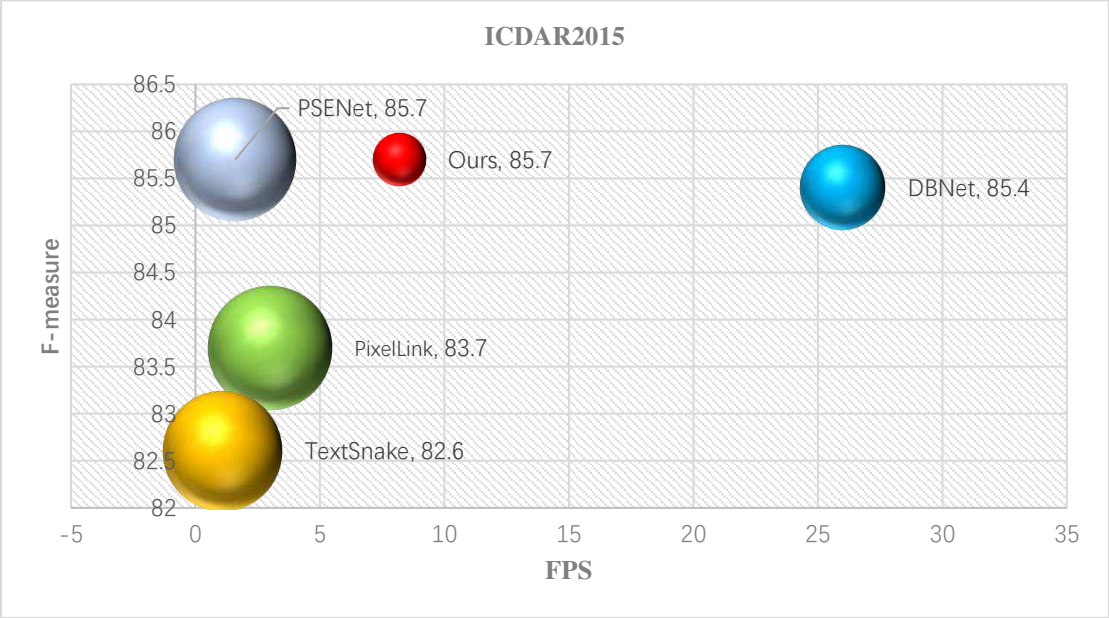


Fig. 4.1 The bubble charts of various methods on ICDAR2015

Fig. 4.2, our method achieves the highest Recall with the smallest number of parameters, but the F-measure is lower than ABCNet (84.4% vs 84.5%). As can be seen from Table 4.5, the Precision, Recall and F-measure of our methods are 85%, 81.1% and 83% on CTW1500, and the number of the parameters is still the smallest.

Multilingual data MSRA-TD500 contains multi-directional texts in both Chinese and English. The detection results on this data set are

Table 4.4 Detection results on Total-Text

Method	P(%)	R(%)	F (%)	#Par(M)	FPS
TextSnake[33]	82.7	74.5	78.4	218.9	-
SAST[35]	83.8	76.9	80.2	-	-
LOMO[36]	87.6	79.3	83.3	-	-

PSENet[1]	84.0	78.0	80.9	229.3	3.9
FCENet[32]	87.4	79.8	83.4	-	-
CRNet[37]	85.8	82.5	84.1	-	-
ABCNet[38]	87.9	81.3	**84.5**	141.00	11
TextField[39]	81.2	79.9	80.6	-	-
DBNet[3]	**88.3**	77.9	82.8	52.8	**50**
Ours	87.2	**81.7**	84.4	**42.3**	12.3

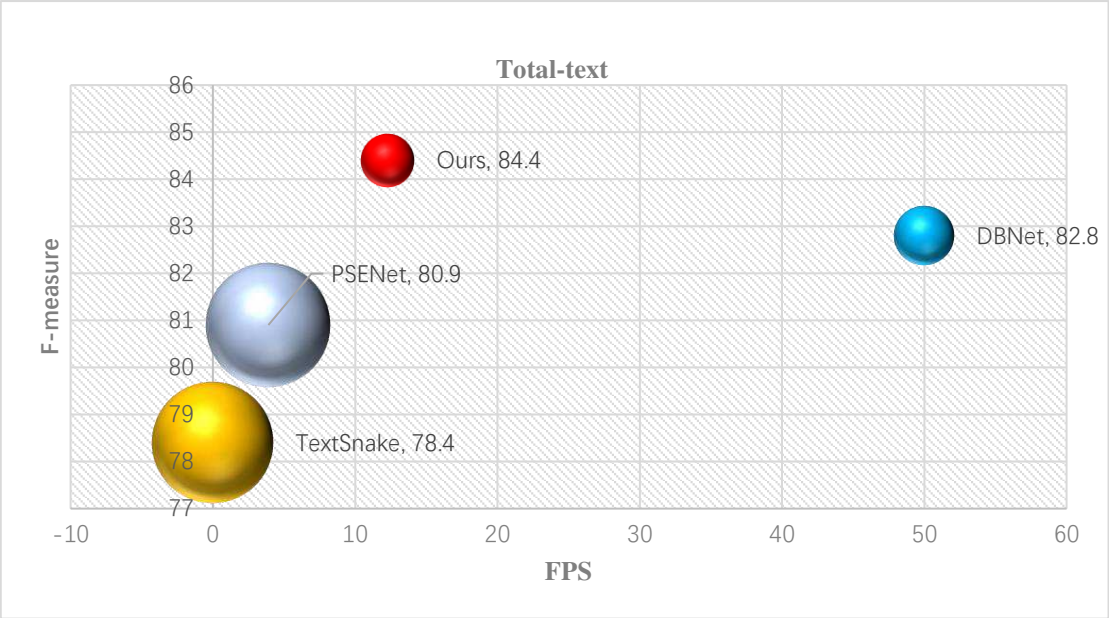


Fig. 4.2 The bubble charts of various methods on Total-text

Table 4.5 Detection results on CTW1500

Method	P(%)	R(%)	F(%)	#Par(M)	FPS
CTPN[8]	60.4	53.8	56.9	-	7.14
FCENet[32]	**85.7**	80.7	**83.1**	-	-
TextSnake[33]	67.9	85.3	75.6	-	-
PSENet[1]	84.8	79.7	82.2	230.3	3.9
LOMO[36]	85.7	76.5	80.8	-	-
ABCNet[38]	84.4	78.5	81.4	-	-
DBNet[3]	84.8	77.5	81.0	52.8	**55**
Ours	85.0	**81.1**	**83.0**	**42.3**	11.2

shown in Table 4.6. Compared with other methods, our method achieves the highest F-measure with the smallest number of parameters, but the Precision is lower than DBNet.

Table 4.6 Detection results on MSRA-TD500

Method	P(%)	R (%)	F(%)	#Par(M)	FPS
PixelLink[2]	83.0	73.2	77.8	234.9	3
TextSnake[33]	83.2	73.9	78.3	218.9	1.1

CRAFT[40]	88.2	78.2	82.9	-	8.6
SegLink[34]	86.0	70.0	77.0	170.0	8.9
MCN[41]	88	79	83	-	-
Conner[42]	87.6	76.2	81.5	-	-
DBNet[3]	**91.5**	79.2	84.9	110.4	32
Ours	90.3	**80.0**	**85.0**	**42.3**	11.7

4 .4.3 Visualization analysis

In this section, we present some TRAM_DBNet detection results on ICDAR2015 and Total-Text, and compare them with those of DBNet. Fig. 4.1 shows the detection results of TRAM_DBNet and DBNet in ICDAR2015. Comparing the second column and the right-most column, we can see that TRAM_DBNet detect real text instances more accurate, especially for the image corners containing text edges. Fig. 4.2 shows the test results on data set Total-Text. Compared with the red boxes obtained by DBNet, the green boxes obtained by TRAM_DBNet can encircle the text more completely.

(a) Original Image (b) DBNet (c) Ours

Fig. 4.1 Some detection results of TRAM DBNet and DBNet in ICDAR2015

(a) Original Image (b) DBNet (c) Ours

Fig. 4.2 Some detection results of TRAM DBNet and DBNet in Total-Text

5 Conclusion

In this paper, we propose a scene text detection model using HRNetV2-W18 as the backbone. We also propose a text area attention module to make the network learning more informative features. The experiment results show that the proposed method achieve good performance, especially in long text and curve text, which verifies that the potential of HRNET for scene text detection. In the following research, we can use lighter HRNetV2-W18 instead of HRNetV2-W18, so that the network can achieve higher accuracy in real-time text detection.

Data availability Data used in this work is available at:
https://github.com/Yuliang-Liu/Curve-Text-Detector
https://rrc.cvc.uab.es/?ch=2&com=tasks
http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500)
https://github.com/cs-chan/Total-Text-Dataset

Code Availability Code used in this work is available at:
https://github.com/zhangyan1005/HR-DBNet.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

[1] Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., & Shao, S. (2019). Shape robust text detection with progressive scale expansion network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9336-9345).

[2] Deng, D., Liu, H., Li, X., & Cai, D. (2018, April). Pixellink: Detecting scene text via instance segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[3] Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2020, April). Real-time scene text detection with differentiable binarization. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 07, pp. 11474-11481).

[4] Vatti, B. R. (1992). A generic solution to polygon clipping. Communications of the ACM, 35(7), 56-63.

[5] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., & Xiao, B. (2020). Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 43(10), 3349-3364.

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[8] Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016, October). Detecting text in natural image with connectionist text proposal network. In European conference on computer vision (pp. 56-72). Springer, Cham.

[9] Ch'ng, C. K., Chan, C. S., & Liu, C. L. (2020). Total-text: toward orientation robustness in scene text detection. International Journal on Document Analysis and Recognition (IJDAR), 23(1), 31-52.

[10] Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017, February). Textboxes: A fast text detector with a single deep neural network. In Thirty-first AAAI conference on artificial intelligence.

[11] Liao, M., Shi, B., & Bai, X. (2018). Textboxes++: A single-shot oriented scene text detector. IEEE transactions on image processing, 27(8), 3676-3690.

[12] Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. International journal of computer vision, 116(1), 1-20.

[13] Zitnick, C. L., & Dollár, P. (2014, September). Edge boxes: Locating object proposals from edges. In European conference on computer vision (pp. 391-405). Springer, Cham.

[14] Dai, P., Zhang, S., Zhang, H., & Cao, X. (2021). Progressive contour regression for arbitrary-shape scene text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7393-7402).

[15] Wu, Y., & Natarajan, P. (2017). Self-organized text detection with minimal post-processing via border learning. In proceedings of the IEEE international conference on computer vision (pp. 5000-5009).

[16] Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., & Jia, J. (2019). Learning shape-aware embedding for scene text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4234-4243).

[17] Lyu, P., Liao, M., Yao, C., Wu, W., & Bai, X. (2018). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 67-83).

[18] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham.

[20] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[21] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[22] Biswas, K., Kumar, S., Banerjee, S., & Pandey, A. K. (2021). SMU: smooth activation function for deep networks using smoothing maximum technique. arXiv preprint arXiv:2111.04682.

[23] Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5693-5703).

[24] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., ... & Valveny, E. (2015, August). ICDAR 2015 competition on robust reading. In 2015 13th international conference on document analysis and recognition (ICDAR) (pp. 1156-1160). IEEE.

[25] Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., ... & Ogier, J. M. (2017, November). Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1454-1459). IEEE.

[26] Chee, C. K., & Chan, C. S. (2017, November). Total-text: A comprehensive dataset for scene text detection and recognition. In 2017 14th IAPR international

conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 935-942). IEEE.

[27] Yao, C., Bai, X., Liu, W., Ma, Y., & Tu, Z. (2012, June). Detecting texts of arbitrary orientations in natural images. In 2012 IEEE conference on computer vision and pattern recognition (pp. 1083-1090). IEEE.

[28] Yao, C., Bai, X., & Liu, W. (2014). A unified framework for multioriented text detection and recognition. IEEE Transactions on Image Processing, 23(11), 4737-4749.

[29] Liu, Y., Jin, L., Zhang, S., Luo, C., & Zhang, S. (2019). Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition, 90, 337-345.

[30] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[31] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: an efficient and accurate scene text detector. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 5551-5560).

[32] Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., & Zhang, W. (2021). Fourier contour embedding for arbitrary-shaped text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3123-3131).

[33] Long, S., Ruan, J., Zhang, W., He, X., Wu, W., & Yao, C. (2018). Textsnake: A flexible representation for detecting text of arbitrary shapes. In Proceedings of the European conference on computer vision (ECCV) (pp. 20-36).

[34] Shi, B., Bai, X., & Belongie, S. (2017). Detecting oriented text in natural images by linking segments. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2550-2558).

[35] Wang, P., Zhang, C., Qi, F., Huang, Z., En, M., Han, J., & Shi, G. (2019, October). A single-shot arbitrarily-shaped text detector based on context attended multi-task learning. In Proceedings of the 27th ACM international conference on multimedia (pp. 1277-1285).

[36] Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., & Ding, X. (2019). Look more than once: An accurate detector for text of arbitrary shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10552-10561).

[37] Zhou, Y., Xie, H., Fang, S., Li, Y., & Zhang, Y. (2020, October). CRNet: A center-aware representation for detecting text of arbitrary shapes. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 2571-2580).

[38] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., & Wang, L. (2020). Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9809-9818).

[39] Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., & Bai, X. (2019). Textfield: Learning a deep direction field for irregular scene text detection. IEEE Transactions on Image Processing, 28(11), 5566-5579.

[40] Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9365-9374).

[41] Liu, Z., Lin, G., Yang, S., Feng, J., Lin, W., & Goh, W. L. (2018). Learning markov clustering networks for scene text detection. arXiv preprint arXiv:1805.08365.

[42] Lyu, P., Yao, C., Wu, W., Yan, S., & Bai, X. (2018). Multi-oriented scene text detection via corner localization and region segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7553-7563).