Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8916028, 11 pages
https://doi.org/10.1155/2020/8916028

Research Article

A Scene Text Detector for Text with Arbitrary Shapes

Weijia Wu,[1] Jici Xing,[2] Cheng Yang,[1] Yuxing Wang ,[1] and Hong Zhou [1]

[1]*Zhejiang University, Key Laboratory for Biomedical Engineering of Ministry, Hangzhou, China*
[2]*Zhengzhou University, School of Information Engineering Institute, Zhengzhou, China*

Correspondence should be addressed to Yuxing Wang; wangyuxing@zju.edu.cn and Hong Zhou; zhouhong_zju@126.com

Received 11 February 2020; Revised 28 April 2020; Accepted 6 May 2020; Published 11 June 2020

Guest Editor: Qian Zhang

The performance of text detection is crucial for the subsequent recognition task. Currently, the accuracy of the text detector still needs further improvement, particularly those with irregular shapes in a complex environment. We propose a pixel-wise method based on instance segmentation for scene text detection. Specifically, a text instance is split into five components: a Text Skeleton and four Directional Pixel Regions, then restoring itself based on these elements and receiving supplementary information from other areas when one fails. Besides, a Confidence Scoring Mechanism is designed to filter characters similar to text instances. Experiments on several challenging benchmarks demonstrate that our method achieves state-of-the-art results in scene text detection with an F-measure of 84.6% on Total-Text and 86.3% on CTW1500.

1. Introduction

Detecting text in the real world is a fundamental computer vision task that directly determines the subsequent recognition results. Many applications in the real world depend on accurate text detection, such as photo translation [1] and autonomous driving [2]. Now, horizontal- [3–5] and oriented-[6–10] based methods no longer meet our requirements, and more flexible pixel-wise detectors [11, 12] have become mainstream. However, precisely locating text instances is still a challenge because of arbitrary angles, shapes, and complex backgrounds.

The first challenge involves text instances with irregular shapes. Unlike other common objects, the shaped instance often cannot be accurately described by a horizontal box or an oriented quadrilateral. Some typical methods (e.g., EAST [8] and TextBox++ [10]) perform well on the common benchmarks (e.g., ICDAR 2013 [13] and ICDAR 2015 [14]) but degrade in curved text challenges, as shown in Figure 1(a).

The second challenge is separating text character boundaries. Although pixel-wise methods do not suffer from a certain shape, they may still fail to separate text areas with adjacent edges, as shown in Figure 1(b).

The third challenge is that text identification may face false positives [15] dilemma because of the lack of context information. Some symbols or characters similar to text may be misclassified.

To overcome the aforementioned challenges, we propose a novel method, called TextCohesion. As shown in Figure 2, our method treats a text instance as a combination of a Text Skeleton and four Directional Pixel Regions, where the previous one roughly represents the shape and profile, and the latter is responsible for refining the original region from four directions. Notably, a pixel belongs to more than one Directional Pixel Regions (e.g., up, left), which means the instance has more chances to be recovered. Furthermore, the confidence score of every Text Skeleton is reviewed, only higher then a threshold is considered as a candidate.

2. Related Work

Detecting text in the wild has been widely studied in the past few years. Before deep learning era, most detectors adopt Connected Components Analysis [16–21] or Sliding Window-based classification [22–25].

Now, detectors are mainly based on deep neural networks. There are two main trends in the field of text

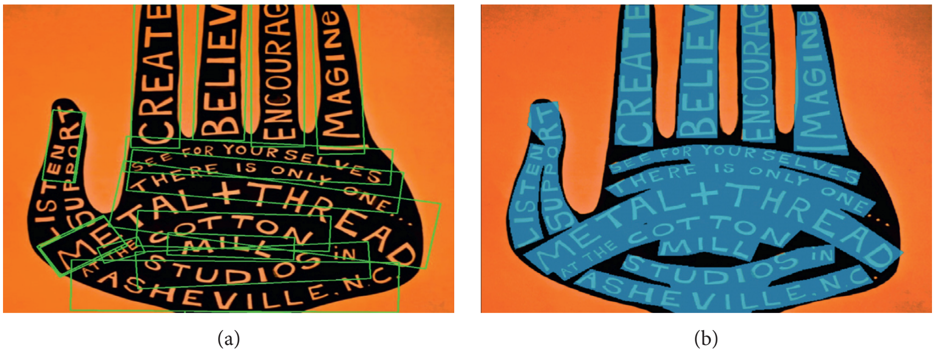


Figure 1: Text detection challenges: (a) bounding box-based methods suffer from a fixed shape and (b) segmentation-based methods may not separate texts with adjacent boundaries.

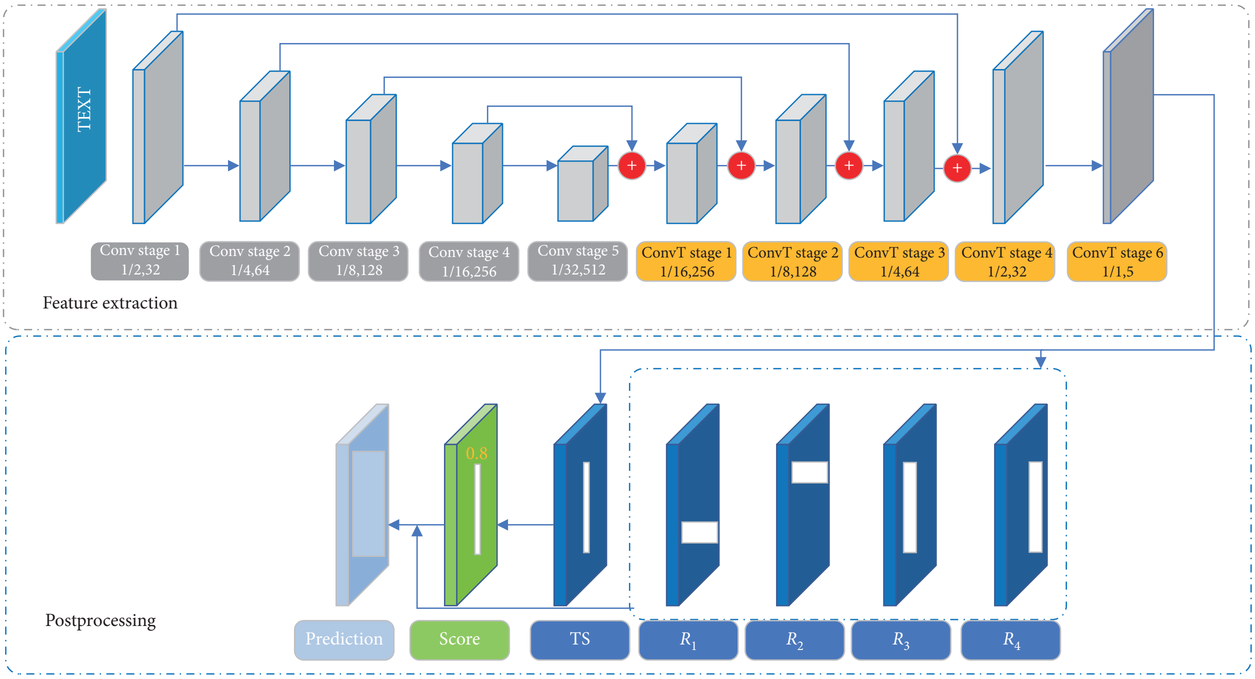


Figure 2: The overall procedure of the proposed method consists of Feature Extraction and Postprocessing. Five feature maps are generated from the backbone (e.g., VGG16) and upsampled in the Feature Extraction step. The DPRs and the TS regions are adopted to reconstruct text instances in the postprocessing step.

detection: regression-based and pixel-based. Inspired by the promising of object detection architectures such as Faster R-CNN [26] and SSD [27], a bunch of regression-based detectors are proposed, which simply regress the coordinates of bounding boxes of candidates as the final prediction. TextBoxes [7] adopts SSD and adjusts the default box to relatively long shape to match text instances. PyrBoxes [28] proposes a SSD-based detector equipped with a grouped pyramid to enrich feature. Sheng [29] proposes a novel text detector with learnable anchors to cover all varieties of texts in natural scene. Lyu [30] detects scene text by localizing corner points of text bounding boxes and segmenting text regions in relative positions. By modifying Faster R-CNN, Rotation Region Proposal Networks [31] insert the rotation branch to fit the oriented shapes of text in natural images. These methods can achieve satisfying performance on horizontal or multioriented text areas. However, they may suffer from the shape of the bounding box, even with rotations. Mainstream pixel-wise methods drew inspirations from the fully convolutional network (FCN) [32], which removes all fully connected layers and is widely used to generate a semantic segmentation map. Convolution transpose operation then helps the shirked feature restore its original size. TextSnake [11] treats a text instance as a sequence of ordered, overlapping disks centered at symmetric order, each of which is associated with potentially variable radius and orientations. It made significant progress on curved text benchmarks. TexeField [33] learns a direction field pointing away from the nearest text boundary to each text point. An image of two-dimensional vectors represents the direction field. SPCNET [34], based on FPN [35] and Mask R-CNN [36], inserts Text Context Module and Rescore mechanism to leave the lack of context information clues and inaccurate classification score. PSENet [37] projects feature into several maps and gradually expand the detected areas from small kernels to large and complete instances. These pixel-based methods significantly improve the performance of curved benchmarks. However, detection failures are still possible in complex situations. Differs from the previous, the proposed method has more opportunities to recover itself. Specifically, the Text Skeleton represents the profile of the instance, which is smaller and less sticky than the original form. Pixels in text areas are divided into two groups according to four directions: the up-down and left-right. Ideally, a TS can be integrated with any group to restore itself. When some regions fail to reproduce, there is also an opportunity to get additional supplementary from others. We conduct extensive experiments on standard benchmarks, including the horizontal the oriented text, and curved text datasets. Evaluations demonstrate that TextCohesion achieves state-of-the-art or very competitive performance.

3. Methodology

The architecture of TextCohesion is depicted in Figure 2, which consists of a feature extraction section and a postprocessing section. For image feature extraction, an FCN-based convolutional backbone followed by an up-sampling step is employed. Five feature maps containing a Text Skeleton (TS) and four Directional Pixel Regions (DPRs) are generated after up-sampling. The TS features are evaluated by a Confidence Scoring Mechanism (CSM), and finally obtaining the predicted text regions incorporated with the DPRs regions. To optimize the proposed network, a corresponding loss function of the TS and DPRs is designed. More details are introduced in the following section.

3.1. Network. The proposed method inherits the popular VGG16 network by keeping the layers from Conv1 to Conv5, converting the last fully connected layers into convolution layers. The input images are first downsampled to the multilevel features with five convolution blocks, and five feature maps (i.e., P_1, P_2, P_3, P_4, P_5) are generated. Then, these features are gradually upsampled to the original size and mixed with the corresponding output of the previous convolution block. The upsampled process can be described by

$$O = U\big(P_1 || U_p\big(P_2 || U_p\big(P_3 || U_p\big(P_4 || U_p(P_5)\big)\big)\big)\big), \tag{1}$$

where O is the output of the network, “” refers to feature concatenation, and U_p is the upsample function (i.e., $\text{Conv}(1,1) - \text{Conv}(3,3) - \text{Deconv} - \text{ReLu}$ used to resize the feature map matching other layers. Five feature maps with the same resolution are leveraged as the prediction of the network (the blue box shown in Figure 2) after the upsample step. Each prediction is composed of a TS and four DPRs in the postprocessing. DPRs contain four feature maps according to different directions: R_1, R_2, R_3, and R_4. The TS is the skeleton of the text instance that is adopted to separate from each other. The CMS is introduced to reduce false positives in terms of evaluating each TS. For clarity, we take a curved text as an example to demonstrate the process of label generation in the rest of Section 3.

3.2. Text Skeleton. Text Skeleton (TS) is an essential component representing the center part of the text instance. As shown in Figure 3(b), the gray area is the TS of the instance. The first step of generating TS is to find the head and tail of the text. Similar to [11], we also use the cosine of adjacent vertices to find the head and tail of text instance, and the remaining two longest sides. The longest two sides along with the text instance (e.g., $t_0 t_n$ and $b_0 b_n$) are called sidelines in the proposed method. Then, n vertices of even distribution are sampled from the two sidelines (i.e., Top Sideline and Bottom Sideline in Figure 3(a)), respectively. After that the vertices in the centerline (Head – Tail in Figure 3) can be averaged from these sampled vertices:

$$c_i(x, y) = \frac{t_i(x, y) + b_i(x, y)}{2}, \tag{2}$$

where $\{t_0, t_1, \ldots, t_i, \ldots, t_n\}$ and $\{b_0, b_1, \ldots, b_i, \ldots, b_n\}$ are vertices in two sidelines of the text instance, respectively, and $\{c_0, c_1, \ldots, c_i, \ldots, c_n\}$ are a set of vertices belong to the center line. Finally, TS is bold by the center line infd3

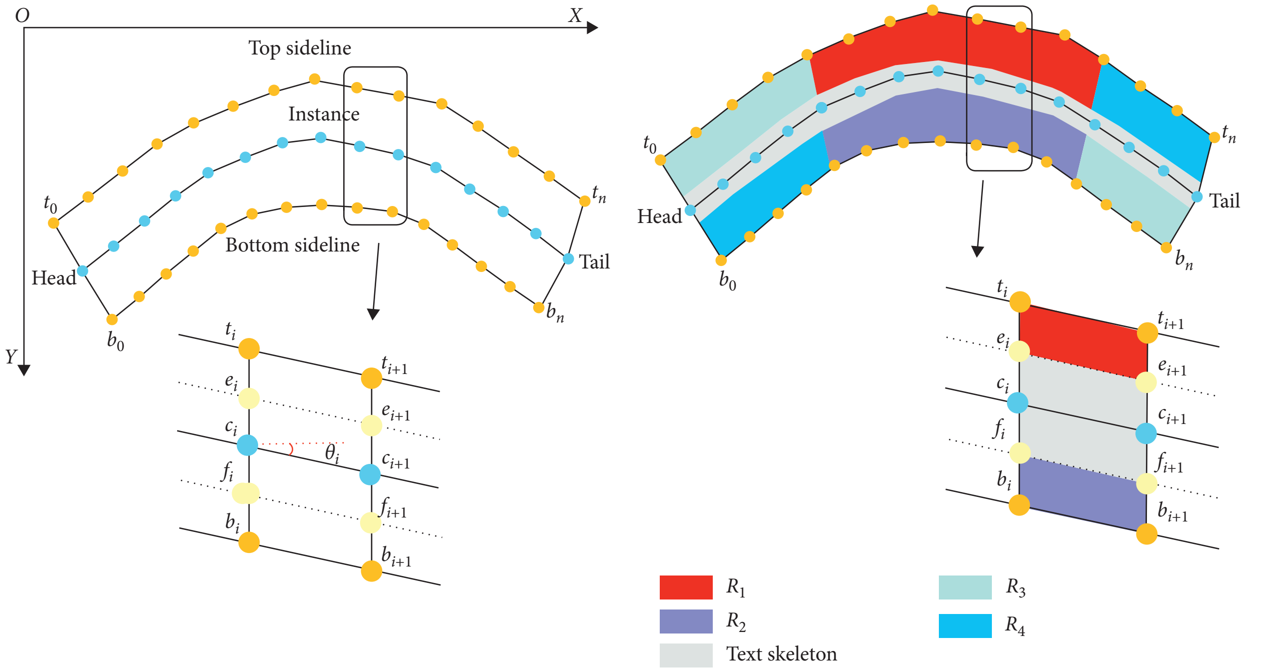
$$\begin{aligned} e_i &= c_i + (t_i - c_i) \times \beta, \\ f_i &= c_i + (b_i - c_i) \times \beta, \end{aligned} \tag{3}$$

where e_i and f_i are pixels that represent the expansion of the center line to both sidelines. The region of $e_i e_{i+1} f_i f_{i+1}$ form a part of TS, as shown in Figure 3(b). β is a parameter that holds the bold rate, and we set it to 0.2 experimentally. When these vertices are completely processed, TS is generated correspondingly.

3.3. Directional Pixel Region. Directional Pixel Regions (DPRs) are used to restore its original form, including R_1, R_2, R_3, and R_4. Pixels in text instance but not in TS are considered as falling into DPR. In Figure 3(b), $t_i t_{i+1} e_{i+1} e_i$ and $f_i f_{i+1} b_{i+1} b_i$ illustrate a fraction of DPR. The direction of every fraction is determined by the tangent angle between its corresponding center vertices (c_i) and the next (c_{i+1}). More specifically, the tangent angle of two adjacent center vertices is calculated by the following equation:

$$\tan(\Theta_i) = \frac{y_{c_{i+1}} - y_{c_i}}{x_{c_{i+1}} - x_{c_i}}, \tag{4}$$

where x and y refer to the coordinates of the center vertices. By comparing the $\tan(\Theta_i)$ of center vertices with α, the regions of $t_i t_{i+1} e_{i+1} e_i$ and $f_i f_{i+1} b_{i+1} b_i$ are labeled as DPRs (R_1, R_2, R_3, R_4) or background. If Θ_i falls into a specific range (e.g., $[-30°, 30°]$), the pixels within its corresponding



(a)

(b)

Figure 3: Label generation: (a) specific mathematical modeling method and (b) a clear example of the TS or the DPRs.

DPRs ($t_i t_{i+1} e_{i+1} e_i$ and $f_i f_{i+1} b_{i+1} b_i$) are considered belonging to the R_1 or R_2. The R_1 can be calculated as follows:

$$R_{1_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other}, \end{cases}$$
$$\text{condition}_1\colon \tan(-\alpha) < \tan(\Theta_i) < \tan(\alpha),$$
$$\text{condition}_2\colon \left(y_{t_i} + y_{t_{i+1}}\right) < \left(y_{c_i} + y_{c_{i+1}}\right), \tag{5}$$

where condition$_1$ is used to distinguish the angle of adjacent center vertices and condition$_2$ ensures the selected pixels are above the TS. α is a parameter that controls the boundary of specific directional regions, which is discussed in detail in the experiment section. y_{t_i} and y_{c_i} are the vertical coordinates of vertices (x, y) on the sideline and the center line, respectively. The generating process of the R_2 is similar to the R_1, but the only difference is that the pixels are located below the TS. Therefore, condition$_2$ is reversed naturally:

$$R_{2_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other}, \end{cases}$$
$$\text{condition}_1\colon \tan(-\alpha) < \tan(\Theta_i) < \tan(\alpha),$$
$$\text{condition}_2\colon \left(y_{t_i} + y_{t_{i+1}}\right) > \left(y_{c_i} + y_{c_{i+1}}\right), \tag{6}$$

where y_{t_i} and $y_{t_{i+1}}$ are logically equivalent to y_{b_i} and $y_{b_{i+1}}$, which are the vertical coordinates of the sampled vertices on the sidelines. The R_3 and R_4 are generated in the same way, as shown below:

$$R_{3_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other}, \end{cases}$$
$$\text{condition}_1\colon \left(\tan\left(\alpha - \frac{\pi}{2}\right) > \tan(\Theta_i)\right) \cap \left(\tan(\Theta_i) > \tan\left(\frac{\pi}{2} - \alpha\right)\right),$$
$$\text{condition}_2\colon \left(x_{t_i} + x_{t_{i+1}}\right) < \left(x_{c_i} + x_{c_{i+1}}\right),$$
$$R_{4_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other}, \end{cases}$$
$$\text{condition}_1\colon \left(\tan\left(\alpha - \frac{\pi}{2}\right) > \tan(\Theta_i)\right) \cap \left(\tan(\Theta_i) > \tan\left(\frac{\pi}{2} - \alpha\right)\right),$$
$$\text{condition}_2\colon \left(x_{t_i} + x_{t_{i+1}}\right) > \left(x_{c_i} + x_{c_{i+1}}\right), \tag{7}$$

where x_{t_i} and x_{c_i} are the horizontal coordinates of vertices on the sideline and the center line, respectively.

3.4. Confidence Scoring Mechanism. To filter out false positives, the confidence score is utilized to weight every TS. If the score of TS is lower than a threshold, then all components of this instance are discarded:

$$\begin{cases} \text{TP}, & \dfrac{\sum_{i=1}^{n} p_i}{n} > \gamma, \\ \text{FP}, & \text{Other}, \end{cases} \tag{8}$$

where n is the total number of pixels in the TS. p_i is the value of the ith pixel in the TS region. TP and FP refer to the true positives and false positives, respectively. γ is the threshold value to filter out the TS with a lower confidence score, and we set it to 0.6 empirically. TS with high confidence will be retained and processed to form the final prediction with its corresponding DPRs. Instead, TS belonging to FP (FalsePositive) with its components are filtered directly. The TS, as the central area of a text instance, contains the key features of the whole text, which are more valuable to use than the whole features of one text instance.

3.5. Loss Function. The proposed method is trained with the following loss function as the three objectives:

$$L = \lambda_1 L_{\mathrm{TS}} + L_{\mathrm{DPR}} + L_{\mathrm{CSM}}, \tag{9}$$

where L_{DPR} is a smooth L_1 [26] loss and L_{TS} and L_{CSM} are crossentropy classification loss functions. The loss of L_{TS} is computed as follows:

$$L_{\mathrm{TS}} = \sum_{n=1}^{N} w_i \mathrm{CrossEntropy}\left(\mathrm{TS}_i, \widehat{\mathrm{TS}_i}\right), \tag{10}$$

where L_{TS} is a self-adjust crossentropy loss function and w_i in equation (10) is a self-adjust weight [9]. For the ith instance with area $= S_i$, every positive pixels within it have a weight of $w_i = B/S_i$. B is the average area of all text instances in one image. In that case, the pixels in text instances with small areas have a bigger weight than the pixels in big text areas. In our experiments, the weight λ_1 is set to 3 as the TS is essential than other components. Losses for DPR and CSM are calculated:

$$\begin{aligned} L_{\mathrm{DPR}} &= \sum_{n=1}^{4} \sum_{i \in \mathrm{DPR}_n} \mathrm{Smooth} L_1\left(\mathrm{DPR}_i, \widehat{\mathrm{DPR}_i}\right), \\ L_{\mathrm{CSM}} &= \sum_{n=1}^{N} \mathrm{CrossEntropy}\left(\mathrm{CS}_i, \widehat{\mathrm{CS}_i}\right), \end{aligned} \tag{11}$$

where L_{DPR} is optimized by a Smooth L_1 loss, and the pixels losses in R_1, R_2, R_3, and R_4 are calculated, respectively, which means that one pixel can be simultaneously categorized as two regions (e.g., R_1 and R_3). L_{CSM} is a standard crossentropy function. $\mathrm{TS}_i, \mathrm{DPR}_i$, and CS_i are ground truth labels and $\widehat{\mathrm{TS}_i}, \widehat{\mathrm{DPR}_i}$, and $\widehat{\mathrm{CS}_i}$ are predicted values.

3.6. Postprocessing. TextCohesion treats every text instance as TS and four DPRs previously; hence, these components should be grouped, forming the final prediction. The postprocessing algorithm is depicted in Algorithm 1:

Every TS represents a text instance, and after passing through CSM, instances with higher confidence are reserved as candidates. Based on these candidates, the corresponding DPRs can be obtained. The postprocessing mainly includes three steps. (1) The TS is used to differentiate the different text instances. (2) For each TS, the outer pixels as initial points are used to search the corresponding pixels in the DPRs iteratively. (3) The TS is eventually merged with corresponding searched regions to form the final prediction. The entire postprocessing is shown in Algorithm 1, where Neighbor (.) refers to a function that obtains the directional information of the adjacent pixels.

4. Experiment

To evaluate TextCohesion, we conduct extensive experiments on both oriented and curved benchmarks and give a detailed description of these datasets for model training and inference, experimental implementation, results with comparisons, and ablation study, respectively.

4.1. Datasets. SynthText [38] is a large scale dataset that contains about 800K synthetic images that are created by blending natural images with text rendered with random fonts, sizes, colors, and orientations. These texts look realistic as the overlaying follows carefully set up configurations and a well-set learning algorithm.

ICDAR2015 [14] contains 1000 training and 500 test images captured by wearable cameras with relatively low resolutions. Each image includes several oriented texts annotated by four vertices of the quadrangles.

ICDAR 2017 MLT (IC17-MLT) [39] is a large scale multilingual text dataset, which includes 7200 training images, 1800 validation images, and 9000 testing images. The dataset is composed of complete scene images that come from 9 languages. Similarly, with ICDAR 2015, the text regions in ICDAR 2017 MLT are also annotated by four vertices of the quadrangle.

CTW1500 [40] is a challenging dataset for curve text detection, which is constructed by Yuliang et al. [18]. It consists of 1000 training images and 500 testing images. Different from traditional text datasets (e.g., ICDAR 2015 and ICDAR 2017 MLT), the text instances in SCUT-CTW1500 are labeled by a polygon with 14 points that can describe the shape of an arbitrarily curve text.

Total-Text [41] is another word-level-based English curve text dataset which is split into training and testing sets with 1255 and 300 images, respectively (Figure 4).

4.2. Implementation Details. Training TextCohesion is optimized by SGD with backpropagation [42]. Momentum and weight decay are set to 0.9 and 5×10^{-4}, respectively. Learning rate is initialized to 10^{-4} and decayed by 0.1 every 30 epochs. Following [11], all training images are augmented online with rotated and cropped with areas ranging from 0.24 to 1.69 and aspect ratios ranging from 0.33 to 3. After that noise, blur, and lightness are randomly adjusted and lastly resized to 512 × 512. We ensure that the text on the augmented images is still legible if they are legible before augmentation. TextCohesion is firstly pretrained on SynthText for 2 epochs and fine-tuned on other datasets. All implementations are deployed on PC with (CPU: Intel(R) Core(TM) i7-7800X CPU @ 3.50 GHz; GPU: GTX 1080).

Inferencing to test the ability of detecting arbitrarily shaped text, we evaluate our method on Total-Text and

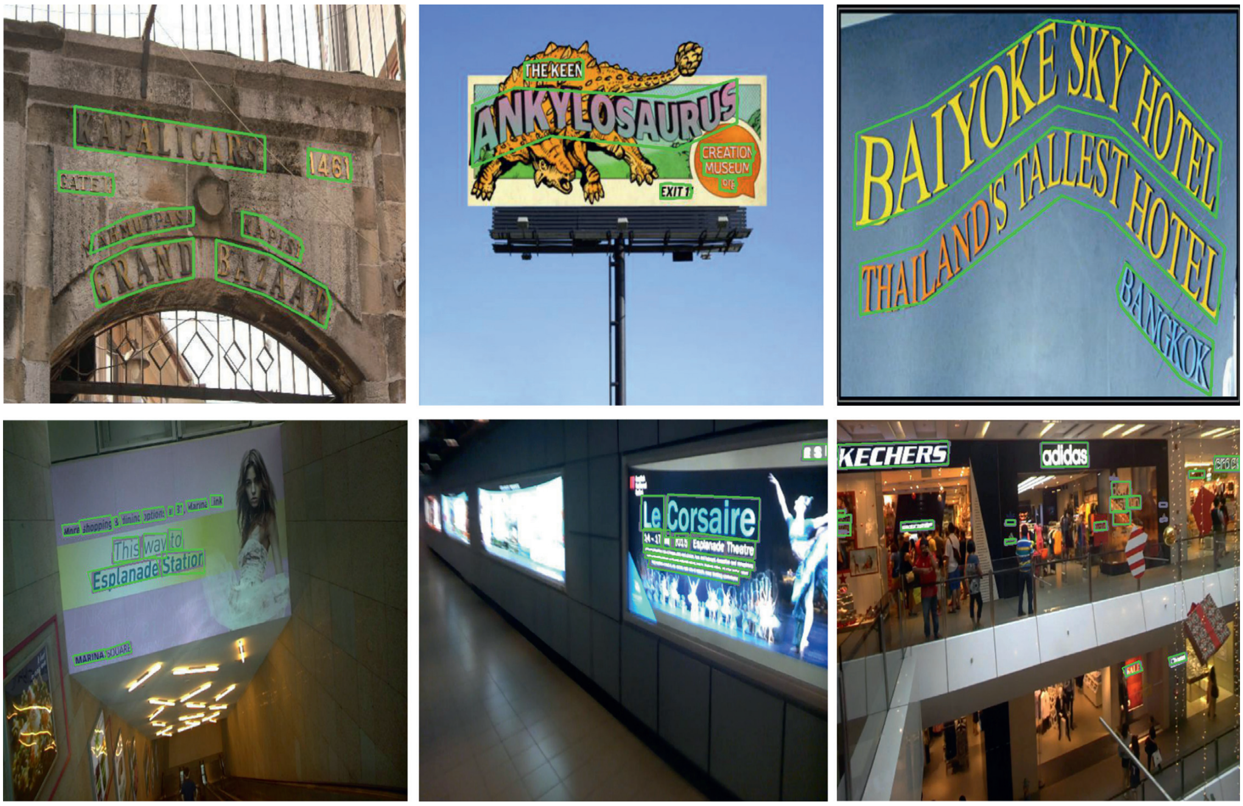


Figure 4: Visualization of the results on curved text datasets.

```
Input:
    $t_i \in \mathrm{TS}$, DPR
Output:
    Result
Function Grouping ($t_i$)
$T \longleftarrow$ Neighbor ($t_i$, up)
$B \longleftarrow$ Neighbor ($t_i$, down)
$L \longleftarrow$ Neighbor ($t_i$, left)
$R \longleftarrow$ Neighbor ($t_i$, right)
if $T$ ! = None and DPR[$T$] == up then
    Tcache $\longleftarrow t_i \cup T$
Grouping (Tcache)
else if $B$! = None and DPR[$B$] == down then
    Bcache $\longleftarrow t_i \cup B$
    Grouping (Bcache)
else if $L$! = None and DPR[$L$] == left then
    Lcache $\longleftarrow t_i \cup L$
    Grouping (Lcache)
else if $R$! = None and DPR[$R$] == right then
    Rcache $\longleftarrow t_i \cup R$
    Grouping (Rcache)
else
    Return $t_i \cup$ Tcache $\cup$ Bcache $\cup$ Lcache $\cup$ Rcache
end if
```

Algorithm 1: Postprocessing algorithm.

SCUT-CTW1500, both of them containing the curved instances. Images in the test stage are also resized to 512 × 512. We report the performance on SCUT-CTW1500 in Table 1, in which we can find that the Precision (88.0%), Recall (84.6%), and F-measure (86.3%) achieved by TextCohesion significantly outperform the ones of other competitors. Remarkably,

Table 1: Experimental results on CTW1500.

Method	Ext.	CTW1500			
		Precision	Recall	F-measure	Time
CTPN [3]	—	60.4	53.8	56.9	0.14
SegLink [44]	—	42.3	40.0	40.8	0.049
EAST [8]	—	78.7	49.1	60.4	0.076
TextSnake [11]	—	65.4	63.4	64.4	0.909
DB-ResNet-18 [43]	—	84.8	77.5	81.0	0.001
PSENet [12]	√	84.8	78.0	80.9	0.429
DB-ResNet-18 [43]	—	84.8	77.5	81.0	0.018
CRAFT [45]	√	87.6	79.9	83.6	0.116
DB-ResNet-50 [43]	√	86.9	80.2	83.4	0.045
TextCohesion (ours)	√	**88.0**	**84.6**	**86.3**	0.206

Ext. indicates external data.

Table 2: Experimental results on Total-Text.

Method	Ext.	Total-Text			
		Precision	Recall	F-measure	Time
SegLink [44]	—	30.0	23.8	26.7	0.049
EAST [8]	—	50.0	36.2	42.0	0.076
MaskSpotter [46]	—	69.0	55.0	61.3	0.208
TextSnake [11]	—	61.5	67.9	64.6	0.909
PSENet [12]	√	84.0	78.0	80.9	0.429
SPCNet [47]	√	83.0	**82.8**	82.9	—
CRAFT [48]	√	87.6	79.9	83.6	0.116
DB-ResNet-50 [43]	√	87.1	82.5	**84.7**	**0.031**
TextCohesion (ours)	√	**88.1**	81.3	84.6	0.206

Ext. indicates external data.

Table 3: Experimental results on ICDAR2015.

Method	Ext.	ICDAR2015			
		Precision	Recall	F-measure	Time
CTPN [3]	—	74.2	51.6	60.9	0.14
SegLink [44]	—	73.1	76.8	75.0	0.048
EAST [8]	—	83.6	73.5	78.2	0.076
PixelLink [9]	—	82.9	81.7	82.3	0.333
DB-ResNet-18 [43]	—	86.8	78.4	82.3	**0.024**
TextSnake [11]	√	84.9	80.4	82.6	0.909
Mask textspotter [46]	√	85.8	81.2	83.4	0.208
PSENet [12]	√	86.9	84.5	85.7	0.429
CRAFT [48]	√	89.8	84.3	86.9	0.116
SPCNet [15]	√	88.7	85.8	87.2	—
DB-ResNet-50 [43]	√	**91.8**	83.2	87.3	0.083
PMTD [49]	√	91.3	87.4	89.3	—
TextCohesion (ours)	√	89.2	**90.2**	**89.7**	0.206

Ext. indicates external data.

the recall and F-measure surpass the second-best record by 4.7% and 2.7%, respectively. Besides, the inference time of the proposed method is also compared with other methods, i.e., DB [43]. The testing scale of the input image is resized to 512 × 512 pixels, and the batch size is set to 1 during all the comparison experiments. The main results are reported in Tables 1–4, where an acceptable inference time can be found.

4.3. Experiments on Curved Text Benchmarks. To test the ability to detect arbitrarily shaped text, we evaluate our method on Total-Text and CTW1500, both of them containing the curved instances. Images in the test stage are also resized to 512 × 512. We report the performance on CTW1500 in Table 1, in which we can find that the Precision (88.0%), Recall (84.6%), and F-measure (86.3%) achieved by TextCohesion significantly outperform the ones of other competitors. Remarkably, the Recall and F-measure surpass the second-best record by 4.7% and 2.7%, respectively.

Our method achieves 88.1%, 81.4%, and 84.6% in Precision, Recall, and F-measure, respectively, outperforming

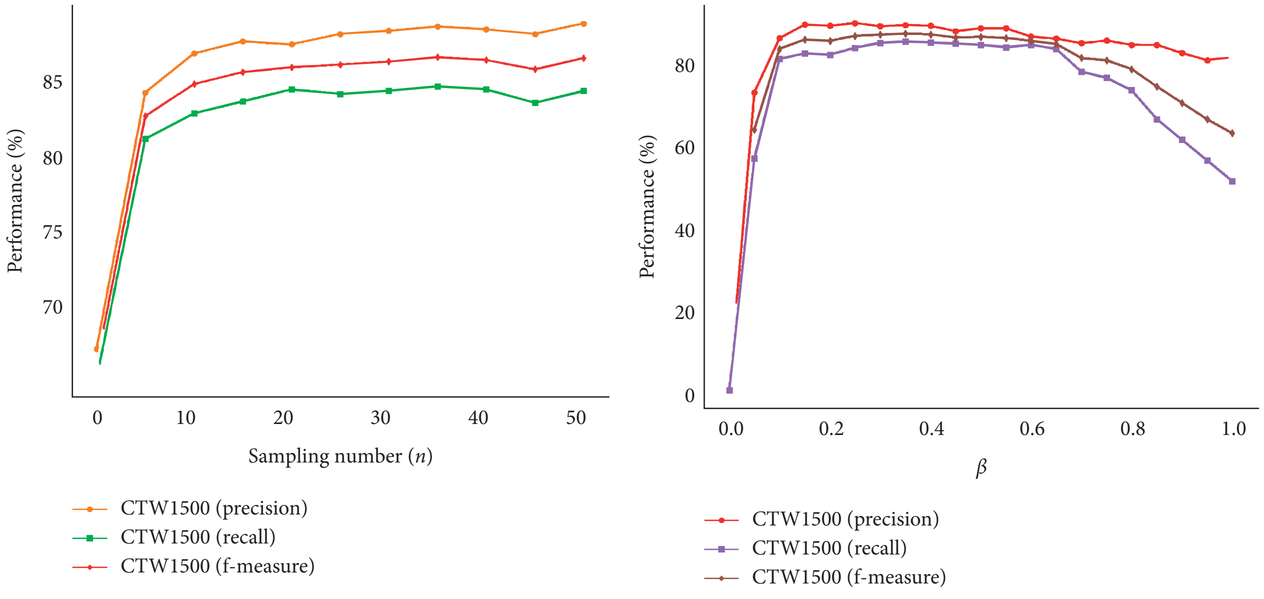
Table 4: Experimental results for ICDAR2017.

Method	Ext.	ICDAR2017			
		Precision	Recall	F-measure	Time
Lyu et al. [30]	—	**83.8**	55.6	66.8	0.175
FOTS [50]	—	81.0	57.6	67.2	0.041
DB-ResNet-18 [43]	—	81.9	63.8	71.7	**0.020**
PSENet [12]	√	77.0	**68.4**	72.5	0.429
CRAFT [51]	√	80.6	68.2	73.9	0.116
DB-ResNet-50 [43]	√	83.1	67.9	**74.7**	0.053
TextCohesion (ours)	√	81.8	66.0	73.1	0.206

Ext. indicates external data.

Table 5: Model results for different values of α in equation (3) and when using the CSM on CTW1500.

Dataset	α	CSM (γ)	Precision	Recall	F-measure
CTW1500	$\pi/6$ (30°)	√(0.6)	88.0	84.6	86.3
CTW1500	$\pi/4$ (45°)	√(0.6)	88.5	82.0	85.2
CTW1500	$\pi/3$ (60°)	√(0.6)	89.5	81.9	85.5
CTW1500	$\pi/6$ (30°)	×	85.2	84.8	85.0
CTW1500	$\pi/6$ (30°)	√(0.4)	85.3	85.1	85.2
CTW1500	$\pi/6$ (30°)	√(0.5)	86.3	84.9	85.6
CTW1500	$\pi/6$ (30°)	√(0.6)	88.0	84.6	86.3
CTW1500	$\pi/6$ (30°)	√(0.7)	88.2	83.3	85.7



(a)

(b)

Figure 5: Ablation study on sampling number (n) and β.

the second competitor with an F-measure of 1.0% on Total-Text. We attribute this excellence to the proposed flexible representation. Instead of taking the text as a whole, the representation treats text as a serial of components and integrates them together to form the final prediction.

4.4. Experiments on Oriented Text Benchmarks. In this section, we evaluate TextCohesion on oriented text datasets. The performance of ICDAR2015 and ICDAR2017 are demonstrated in Tables 3 and 4, which also achieves F-measure of 89.1% and 73.1%, respectively. From these results, it can be observed that our method also achieves very competitive performance in dealing with oriented text. Meanwhile, thanks to the robust feature representation, TextCohesion can as well locate the text instance with small instances and in complex illuminations and variable scales.

4.5. Analyses and Discussion

4.5.1. Influence of the Number of Samples (n). We sample n points on the top sideline and bottom sideline for each text instance, and use these points to split text instances better. To further study the Influence of the number of points on the sampling precision, an ablation experiment is performed, as shown in Figure 5(a). Theoretically, the performance of the model will improve with the increase of sampling precision. In the experiment, we found that the performance of the model hardly improve further (around 85%) when the sampling number (n) is greater than 10. n is set to 40 in all experiments.

4.5.2. Influence of β in Equation (2). β as an important parameter is used to control the ratio of the TS area to the DPR area. As shown in Figure 5(b), when the value of β is within the range of [0.1, 0.6], the network performs well. In all experiments, β is set to 0.2.

4.5.3. Influence of α in Equation (3). α is used to delineate the top, bottom, left, and right regions. 30°, 45°, and 60° are the three specific angles used to investigate the influence of α. As shown in Table 1, the F-measure is relatively good when α is 30°, so we set α to 30° in all experiments.

4.5.4. Influence of the Confidence Scoring Mechanism. The CSM is used to filter out the false positives (e.g., those symbols or characters that are similar to text). The influence in the results of the model when using the CSM is shown in Table 5. The precision improves 2.8% after the CSM (0.6) is used. To test the robustness of the proposed model while changing the γ in equation (8), a comparison experiment is set in Table 5, and the F-measure is relatively good when γ is 0.6. In all experiments, γ is set to 0.6.

5. Conclusion and Outlook

In this paper, we propose a novel text detector, which achieves upto 86.3% F-measure among common text benchmarks, including text instance with irregular shapes. The text instance modeling method utilized in this detector could precisely detect text with arbitrary boundaries by splitting one text instance into four DPRs and a TS region. Moreover, a Confidence Scoring Mechanism is incorporated into this detector to filter out false positives, which further improves its detection precision. Simulation experiment results show that the proposed text detector performs well in scene text detection. The proposed method might have potential applications in the field of photo translation, autonomous driving, and product identification.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Weijia Wu and Jici Xing contributed equally to this work.

Acknowledgments

This work was supported by the National Key Research and Development Project (Grant no. 2019YFC0118202), National Natural Science Foundation of China (Grant no. 61803332), and Scientific Research Fund of Zhejiang Provincial Education Department (Grant no. Y201941642).

References

[1] C. Yi and Y. Tian, "Scene text recognition in mobile applications by character descriptor and structure configuration," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2972–2982, 2014.

[2] Y. Zhu, M. Liao, M. Yang, and W. Liu, "Cascaded segmentation-detection networks for text-based traffic sign detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 209–219, 2018.

[3] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016.

[4] D. Wu, R. Wang, P. Dai, Y. Zhang, and X. Cao, "Deep strip-based network with cascade learning for scene text localization," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.

[5] X. Zhu, Y. Jiang, S. Yang, and Wang, "Deep residual text detection network for scene text," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[7] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: a fast text detector with a single deep neural network," *Computer Vision and Pattern Recognition, AAAI* , 2017, http://arxiv.org/abs/1611.06779.

[8] X. Zhou, C. Yao, H. Wen et al., "East: an efficient and accurate scene text detector," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[9] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: detecting scene text via instance segmentation," *Computer Vision and Pattern Recognition, AAAI*, 2018, http://arxiv.org/abs/1801.01315.

[10] M. Liao, B. Shi, and X. Bai, "TextBoxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[11] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: a flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.

[12] W. Wang, E. Xie, X. Li et al., "Shape robust text detection with progressive scale expansion network," in *Proceedings of the*

Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 2018.

[13] D. Karatzas, F. Shafait, S. Uchida et al., "Icdar 2013 robust reading competition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Washington, DC, USA, August 2013.

[14] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, and S. Ghosh, "Icdar 2015 competition on robust reading," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, August 2015.

[15] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *Computer Vision and Pattern Recognition, AAAI*, 2019, http://arxiv.org/abs/1811.08605.

[16] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, CA, USA, pp. 2963–2970, 2010.

[17] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1241–1248, Sydney, Australia, 2013.

[18] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognition*, vol. 31, no. 12, pp. 2055–2076, 1998.

[19] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images,," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, IEEE, Providence, RI, USA, June 2012.

[20] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, 2011.

[21] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[22] A. Coates, B. Carpenter, C. Case et al., "Text detection and character recognition in scene images with unsupervised feature learning," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, vol. 11, pp. 440–445, Beijing, China, November 2011.

[23] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 429–434, IEEE, Beijing, China, September 2011.

[24] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 1457–1464, IEEE, Barcelona, Spain, November 2011.

[25] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, Tsukuba, Japan, pp. 3304–3308, 2012.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[27] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016.

[28] F. Sheng, Z. Chen, W. Zhang, and B. Xu, "Pyrboxes: an efficient multi-scale scene text detector with feature pyramids," *Pattern Recognition Letters*, vol. 125, pp. 228–234, 2019.

[29] F. Sheng, Z. Chen, T. Mei, and B. Xu, "A single-shot oriented scene text detector with learnable anchors," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1516–1521, IEEE, Shanghai, China, July 2019.

[30] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "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, Salt Lake City, UT, USA, June 2018.

[31] J. Ma, W. Shao, H. Ye et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[33] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.

[34] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9038–9045, 2019.

[35] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, Honolulu, HI, USA, July 2017.

[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

[37] W. Wang, E. Xie, X. Li et al., "Shape robust text detection with progressive scale expansion network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9336–9345, Long Beach, CA, USA, June 2019.

[38] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2315–2324, Las Vegas, NV, USA, June 2016.

[39] N. Nayef, F. Yin, I. Bizid et al., "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1454–1459, IEEE, Kyoto, Japan, November 2017.

[40] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognition*, vol. 90, pp. 337–345, 2019.

[41] C. K. Ch'ng and C. S. Chan, "Total-text: a comprehensive dataset for scene text detection and recognition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.

[42] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[43] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," 2019, http://arxiv.org/abs/1911.08947.

[44] B. Shi, X. Bai, S. Belongie, and Xiang, “Detecting oriented text in natural images by linking segments,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[45] B. Shi, X. Bai, and S. Belongie, “Detecting oriented text in natural images by linking segments,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2550–2558, Honolulu, HI, USA, July 2017.

[46] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–88, Munich, Germany, September 2018.

[47] Q. Wang, J. Gao, M. Zhang, J. Xing, and W. Hut, “SPCNet: scale position correlation network for end-to-end visual tracking,” in *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1803–1808, IEEE, Beijing, China, August 2018.

[48] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character region awareness for text detection,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.

[49] J. Liu and X. Liu, “Pyramid mask text detector,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.

[50] X. Liu, D. Liang, S. Yan et al., “Fast oriented text spotting with a unified network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[51] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character region awareness for text detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9365–9374, Long Beach, CA, USA, June 2019.