Progressive Contour Regression for Arbitrary-Shape Scene Text Detection

Pengwen Dai[1,2], Sanyi Zhang[1,4], Hua Zhang[1], Xiaochun Cao[1,2,3*]
[1]SKLOIS, Institute of Information Engineering, CAS, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China
[4]School of Electrical and Information Engineering, Tianjin University, Tianjin, China
{daipengwen,zhanghua,caoxiaochun}@iie.ac.cn,zhangsanyi@tju.edu.cn

Abstract

State-of-the-art scene text detection methods usually model the text instance with local pixels or components from the bottom-up perspective and, therefore, are sensitive to noises and dependent on the complicated heuristic post-processing especially for arbitrary-shape texts. To relieve these two issues, instead, we propose to progressively evolve the initial text proposal to arbitrarily shaped text contours in a top-down manner. The initial horizontal text proposals are generated by estimating the center and size of texts. To reduce the range of regression, the first stage of the evolution predicts the corner points of oriented text proposals from the initial horizontal ones. In the second stage, the contours of the oriented text proposals are iteratively regressed to arbitrarily shaped ones. In the last iteration of this stage, we rescore the confidence of the final localized text by utilizing the cues from multiple contour points, rather than the single cue from the initial horizontal proposal center that may be out of arbitrary-shape text regions. Moreover, to facilitate the progressive contour evolution, we design a contour information aggregation mechanism to enrich the feature representation on text contours by considering both the circular topology and semantic context. Experiments conducted on CTW1500, Total-Text, ArT, and TD500 have demonstrated that the proposed method especially excels in line-level arbitrary-shape texts. Code is available at https://github.com/dpengwen/PCR.

1. Introduction

Scene text detection has attracted increasing attention in the computer vision community for its ubiquitous applications [46, 11, 10], such as scene understanding, visual search, automatic driving, etc. However, it is a challenging task, due to the effect of scene factors (*e.g.*, complex background, perspective distortion, and various illumination) and the specific characteristics of scene texts (*e.g.*, arbitrary-shape layout, no well-defined closed boundaries, and various aspect ratios).

(a) (b) (c)

Figure 1: Illustration of the contour evolution. It evolves the contour of the horizontal text proposal (a) to the corner points (cyan) of the oriented text proposal (b), and then the contour of the oriented text proposal is further evolved to close to the ground-truth contour (green) for multiple times (c). The red point and dash lines are the center and size of the axis-aligned box of the arbitrary-shape text. Yellow arrows on the contours indicate the information passing.

To exploit an arbitrary-shape scene text detector that requires localizing explicit contours of text instances, the bottom-up methods [6, 27, 42, 43, 51, 38, 44, 7, 53, 17, 59] have become the dominant mainstream. These methods perform the pixel-wise semantic segmentation on the entire or shrunk text regions, and simultaneously predict the auxiliary information of each pixel for clustering text pixels into different instances [6, 42, 53, 51, 38, 43, 44, 17, 63] or local components [27, 1, 7, 59]. However, due to the huge number of pixels in the image, it would involve complicated post-processings to achieve the accurate bounding box for arbitrary-shape texts. Furthermore, the pixels are easy to generate ambiguous predictions when two scene texts are too close or the space between characters in each text is too large. Though the local components are more robust to the pixel-wise noises, they still need to be linked to different text instances by the heuristic rules [27, 7] or the relational

*Corresponding Author.

reasoning network [59]. Most importantly, existing bottom-up methods focus on the local text cues instead of the integrated geometric layout of texts, which leads to the lack of global perception.

To obtain the global geometric layout, the top-down methods [5, 50, 4, 22, 13] are proposed to localize arbitrary-shape scene texts. These methods first carry out the binary segmentation within the text proposals and then utilize the contour extraction algorithm [35] to obtain the contour of the segmentation mask. However, they require elaborate anchors, and are sensitive to the inaccurate localization of text regions. Differently, some other top-down methods [64, 24, 45, 2, 41] regress the key points on text contours within the text proposals. However, these methods are also dependent on the artificially-designed anchors, and ignore the constraint of global geometric layout among key points. To address these issues, the single-shot top-down methods [20, 39] reconstruct the text contour via the control points of Bezier curves, or encode the text contour based on the geometric information under polar space. Nevertheless, these single-shot methods only perceive scene texts with complex geometry layout once, which would generate inaccurate localization. It is inconsistent with the human visual system in which look more than once is usually required [58].

In this paper, we develop a novel scene text detection method via **P**rogressive **C**ontour **R**egression, called **PCR**, to effectively localize the arbitrary-shape scene text. Specifically, we first generate horizontal text proposals by estimating their center points and sizes. Then we regress the global contours of the horizontal text proposals to the corner points of oriented text proposals. After that, we evolve the contours of the oriented text proposals into arbitrary-shape text contours and iteratively refine them, as described in Fig. 1. This progressive strategy is helpful to perceive texts with complex layouts, thus can generate accurate localization for arbitrary-shape scene texts. To facilitate the regression of text contour points, we exploit a contour information aggregation technique. It not only makes full use of the cyclicity of text contours in geometric topology, but also assembles the contour information into sink nodes in semantic to avoid the influence of redundant or noisy points on text contours. This technique can effectively gather rich information and distribute them to each contour point to enhance the feature representation. Additionally, the center points of some horizontal bounding boxes of texts with complex geometric layouts (*e.g.*, extremely-curved texts, texts with large character spaces, etc.) are not on texts, as shown in Fig. 1 (a). Meanwhile, the single center point is also insufficient to represent arbitrary-shape texts based on local cues. Thus, based on the predicted centers, the generated horizontal text proposals would contain some false detections. To increase the confidence of the final localized contours, we propose a reliable contour localization mechanism, which is performed by the scoring mechanism based on multiple sampled points on text contours. Our proposed method is an anchor-free model, and can be trained in an end-to-end manner. The model can directly output the polygonal detection with only one simple NMS post-processing.

The main contributions of this work are as follows:

i) We propose a novel progressive contour regression framework to detect arbitrary-shape scene texts, which has achieved state-of-the-art performances on multiple public benchmarks, *e.g.*, CTW1500, Total-Text, ArT, and TD500.

ii) A contour information aggregation is exploited to enrich the contour feature representation, which can restrain the effect of redundant and noisy contour points and generate more accurate localization for arbitrary-shape texts.

iii) A reliable contour localization mechanism is developed to rescore the localized contours, which can effectively relieve the false detections.

2. Related Work

Scene text detection has been extensively studied for many years. Comprehensive reviews of scene text detection methods are illustrated in [56, 26]. In the era of deep learning, these scene text detectors can be classified into two categories: bottom-up methods and top-down methods.

Bottom-up Scene Text Detectors: In the early stage of deep learning, some scholars [60] regard the scene text detection as a semantic segmentation problem, and then exploit a complex heuristic grouping algorithm to separate different text instances. Recently, some auxiliary information [6, 42, 53, 38, 43, 51, 44, 17, 63] of each pixel in the entire or shrunk text region is also predicted in an end-to-end framework to better separate pixels belonging to different text regions. For example, in PSENet [43], a progressive scale expansion algorithm is applied to fuse different-scale segmentation maps. In SAE [38] and PAN [44], the embedding vectors of pixels are learned by pulling the pixel embedding of the same text instance and pushing the pixel embedding of different text instances.

Similarly, some researchers [37, 34, 36, 27, 1, 7, 59] decompose a text instance into a series of simple local components, and explore the relationships among these components before grouping them into an entire text instance. The local components are constructed by two strategies. One is regressed from the simple anchors, *e.g.*, CTPN [37], SegLink [34], and ICG [36]. The other is grouped from pixels based on their local geometric attributes, *e.g.*, TextSnake [27], CRAFT [1], TextDragon [7], and DRRGN [59]. Then, the relationships of these components could be fully explored in an end-to-end framework. For example, SegLink [34] regards the link between components as a classification task, while DRRGN [59] employs a deep relational reasoning graph network to deduce the linkages.

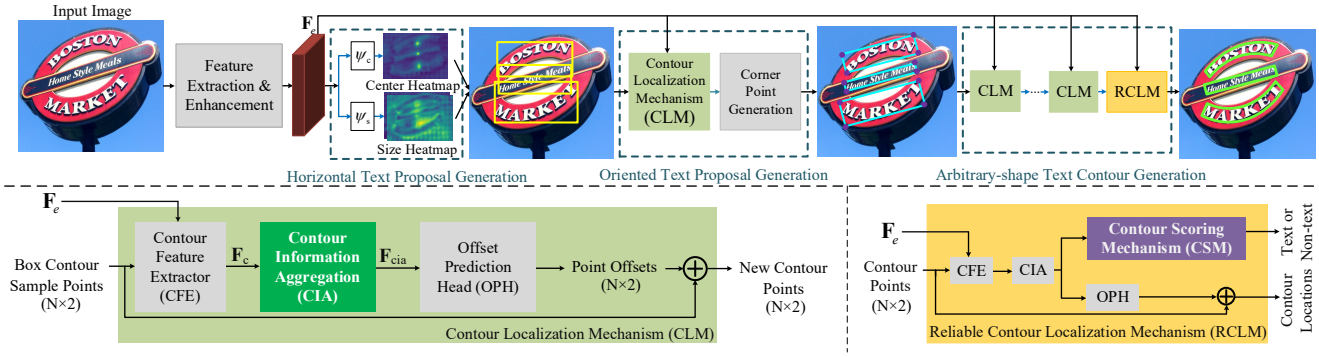


Figure 2: Overview of our proposed architecture. Given an input image, the feature extraction and enhancement module is first utilized to extract multi-scale features, and fuse these features to obtain the representative feature $\mathbf{F}_e$. Then, $\mathbf{F}_e$ is fed into the horizontal text proposal generation module to estimate the center and size heatmap of the text proposal. Next, we employ the Contour Localization Mechanism (CLM) to regress the locations of contour points based on the initialized horizontal proposal contour points, before generating the corner points of the oriented text proposal. After that, the contour points of the oriented proposal are initialized and are iteratively forced to close to the contour of the arbitrary-shape text by multiple CLMs. Finally, the Reliable Contour Localization Mechanism (RCLM) is exploited to generate final text contours with high confidence learned by the contour scoring mechanism. Note that in CLM, we exploit the Contour Information Aggregation (CIA) technique to enrich the raw contour feature $\mathbf{F}_c$ for obtaining the representative contour feature $\mathbf{F}_{cia}$, before learning contour point offsets.

Top-down Scene Text Detectors: These methods usually treat texts as a special type of object. Some well-known text detectors [16, 21, 15, 30, 18, 40] have been exploited to detect the horizontal or multi-oriented scene texts. In RRPN [30], the authors design the rotated anchors to facilitate the regression of the arbitrary-oriented scene texts. RRD [18] rotates the convolutional filters to learn rotation-sensitive features while ITN [40] learns the affine matrix to obtain the geometry-aware representations.

As the layouts of some texts can be curved or wavy in the wild, the horizontal, rotated or quadrilateral bounding boxes are not fit for them well. Thus some methods [5, 50, 28, 13, 4, 49, 22] perform the pixel-wise binary segmentation on the candidate boxes to localize arbitrary-shape texts. For example, Mask-TextSpotter [13] incorporates the character recognition branch in the Mask-RCNN [8] framework to formulate an end-to-end trainable model for better filter text-like regions, and the character recognition is achieved by the multi-class semantic segmentation. In Mask-TTD [22], a tightness prior is utilized to adjust text proposals for better covering the entire text region, and the text frontier information is skillfully exploited to enhance the text mask prediction.

Differently, some contour-based methods [64, 24, 45, 2, 41] are proposed to localize the key points on the contours of arbitrary-shape texts within the proposals. For example, CTD-TLOC [24] regresses the offsets between the top-left point of the circumscribed box and the key points on text contours, and utilizes the Recurrent Neural Network (RNN) to smooth the horizontal and vertical offsets. Considering that regressing a fixed number of key points is not suitable for some various-shape texts, ATRR [45] introduces RNN to adaptively regress multiple point pairs until meeting a stop token. Additionally, to directly localize the text contours at one pass, TextRay [39] regresses contour points under the polar space.

In this paper, we propose a novel contour-based method to detect arbitrary-shape scene texts in a progressive regression manner from the top-down perspective. Different from adopting the progressive contour regression for the semi-automatic annotation [19] and the instance segmentation [32], our method enriches the feature representations of text contours by considering both the cyclicity in geometric topology and the contexts in semantic. Moreover, our model estimates the reliable score for the localized text contour, which can effectively suppress false positives.

3. Methodology

In this section, we first introduce the architecture of our proposed method. After that, we present our novel contour information aggregation technique and illustrate the reliable contour localization mechanism. Finally, the details of the training and inference of our model are described.

3.1. Progressive Contour Regression Architecture

As illustrated in Fig. 2, the input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ (H and W are the height and width of the image) is first fed into the the feature extraction and enhancement module [61] to extract multi-scale visual features. Then we fuse them

to obtain a more representative feature $\mathbf{F}_e \in \mathbb{R}^{\frac{H}{\sigma} \times \frac{W}{\sigma} \times D_e}$, where σ is the output stride and D_e is the feature dimension. Next, the horizontal text proposal generation module predicts the center and size of axis-aligned bounding boxes of texts. After that, the oriented text proposal generation module is exploited to regress the corner points of oriented bounding boxes, based on the contours of horizontal text proposals. Finally, the arbitrary-shape text contour generation network is utilized to evolve the contours of oriented bounding boxes to localize arbitrary-shape scene texts.

(1) Horizontal Text Proposal Generation: This module generates the center heatmap $\widehat{\mathbf{P}} = \Psi_c(\mathbf{F}_e) \in \mathbb{R}^{\frac{H}{\sigma} \times \frac{W}{\sigma}}$ and the size heatmap $\widehat{\mathbf{Q}} = \Psi_s(\mathbf{F}_e) \in \mathbb{R}^{\frac{H}{\sigma} \times \frac{W}{\sigma} \times 2}$ of scene texts, where Ψ_c and Ψ_s consist of several convolution layers like [61]. In this network, we regard the text center localization as a pixel-wise logistic regression with focal loss [61], which is defined as,

$$\mathcal{L}_{center} = \frac{-1}{N_t} \sum_i \begin{cases} (1-\widehat{\mathbf{P}}_i)^{\alpha} log(\widehat{\mathbf{P}}_i), & if \ \ \mathbf{P}_i = 1, \\ (1-\mathbf{P}_i)^{\beta} (\widehat{\mathbf{P}}_i)^{\alpha} log(1-\widehat{\mathbf{P}}_i), & o.w., \end{cases} \tag{1}$$

where N_t is the number of texts; i denotes the position index on the heatmap; $\mathbf{P}$ is the ground-truth center heatmap, which is generated following [61]; α and β are the penalty hyper-parameters, which are set to 2 and 4 in experiments.

Besides, the box size regression only considers the center points of the axis-aligned bounding boxes, whose training objective is formulated as,

$$\mathcal{L}_{size} = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathcal{S}_{L_1}(\widehat{\mathbf{Q}}_i - \mathbf{Q}_i), \tag{2}$$

where $\mathcal{S}_{L_1}$ is the smooth L1 loss [33]; $\widehat{\mathbf{Q}}_i$ denotes the predicted sizes at the i-th center point; $\mathbf{Q}_i$ is the corresponding ground truth.

(2) Oriented Text Proposal Generation: In this module, we first evenly sample N_o points along the contour for each horizontal text proposal. Note that the horizontal text proposal denotes the ground-truth box for training while is the predicted box for testing. Then, we estimate the new locations of these sampled points via the Contour Localization Mechanism (CLM). Specifically, in CLM, the contour feature extractor projects the contour points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N_o}$ on the feature $\mathbf{F}_e$ to generate the semantic feature $\mathbf{F}^{sem} \in \mathbb{R}^{N_o \times D_e}$. At the same time, the location information $\mathbf{F}^{loc} \in \mathbb{R}^{N_o \times 2}$ of contour points, formulated as $\mathbf{F}_i^{loc} = \mathbf{x}_i - \mathbf{x}_{min}$ in which i means the index of contour points and $\mathbf{x}_{min}$ means the most top-left coordinate of contour points, is also considered. The semantic feature $\mathbf{F}^{sem}$ and the contour location information $\mathbf{F}^{loc}$ would be concatenated to generate raw contour feature $\mathbf{F}_c \in \mathbb{R}^{N_o \times (D_e+2)}$. Next, the contour information aggregation (CIA) module takes $\mathbf{F}_c$ as input to generate a more representative contour feature $\mathbf{F}_{cia} \in \mathbb{R}^{N_o \times D}$ (more details are illustrated in Section 3.2). The contour feature $\mathbf{F}_{cia}$ is then fed into an Offset Prediction Head (OPH) to generate contour point offsets $\mathbf{O} \in \mathbb{R}^{N_o \times 2}$. Note that OPH is composed of three 1×1 convolution layers (former two layers are equipped with ReLU), whose number of filters are 256, 64 and 2, respectively. After that, new locations of contour points $\mathbf{X}' \in \mathbb{R}^{N_o \times 2}$ are obtained by $\mathbf{X} + \mathbf{O}$. Finally, the corner point generation module calculates the corner points of each text as $\mathbf{X}'[i * \lfloor N_o / N_c \rfloor]$, where $i \in \{0, 1, ..., N_c - 1\}$. N_c is the number of corner points of the oriented text proposal. $\lfloor \cdot \rfloor$ denotes the floor operation. Thus, the predicted corner points of all texts in each image can be termed as $\widehat{\mathbf{X}}^{corner} \in \mathbb{R}^{N_t \times N_c \times 2}$, whose loss function is formulated as,

$$\mathcal{L}_{corner} = \frac{1}{N_t N_c} \sum_{i=1}^{N_t} \sum_{j=1}^{N_c} \mathcal{S}_{L_1}(\widehat{\mathbf{X}}_{ij}^{corner} - \mathbf{X}_{ij}^{corner}), \tag{3}$$

where $\mathbf{X}^{corner}$ denotes the ground-truth corner points of the oriented bounding box.

(3) Arbitrary-Shape Text Contour Generation: In this module, we first initialize the contour of the oriented text proposal with N_a points, and then we employ K CLMs to progressively regress the oriented text proposal contour to the arbitrary-shape text contour. It is worth noting that the oriented text proposal is the ground-truth for training while denotes the box constructed from the predicted corner points for testing. Considering that the contours may be evolved from some false detections, we exploit a Reliable Contour Localization Mechanism (RCLM) to increase the confidence of detected contours (more details are illustrated in Section 3.3). RCLM outputs new contour point locations $\widehat{\mathbf{X}}^{final} \in \mathbb{R}^{N_t \times N_a \times 2}$ and the contour confidence $\mathbf{s} \in \mathbb{R}^{N_t \times 2}$. Therefore, the loss function of the contour location evolution is expressed as,

$$\mathcal{L}_{evolution} = \frac{1}{N_t N_a} \sum_{i=1}^{N_t} \sum_{j=1}^{N_a} \mathcal{S}_{L_1}(\widehat{\mathbf{X}}_{ij}^{final} - \mathbf{X}_{ij}^{final}), \tag{4}$$

where $\mathbf{X}_{ij}^{final}$ is the j-th contour point of the i-th ground-truth text; it is evenly sampled from the contour of the arbitrary-shape scene text. Moreover, the training objective of the contour scoring mechanism is regarded as the text/non-text classification task, which is formulated as,

$$\mathcal{L}_{csm} = -\frac{1}{N_t} \sum_{i=1}^{N_t} log(\mathbf{s}_i^l), \tag{5}$$

where l is the classification label of the contour; $\mathbf{s}_i^l$ is the score of the region enclosed by the i-th contour belonging to the background ($l = 0$) or text ($l = 1$).

3.2. Contour Information Aggregation

Scene text contours should form closed shapes. However, some points along the contour of texts are redundant and contain noisy cues. To enrich the feature representation of

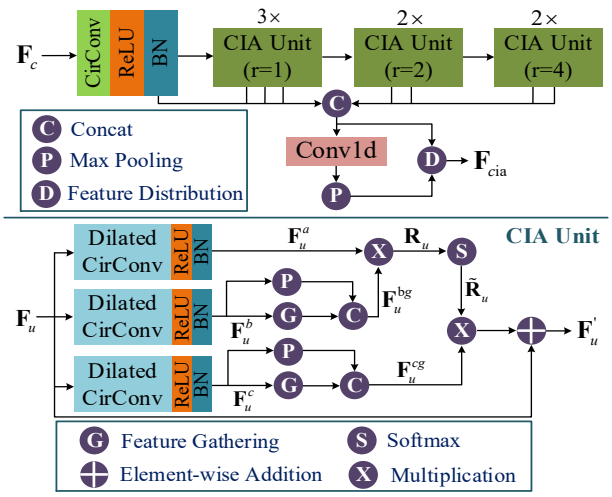


Figure 3: Illustration of the contour information aggregation. r means the dilation rate of convolution kernels in the CIA unit.

contours, we propose a Contour Information Aggregation (CIA) technique. As shown in Fig. 3, the raw contour feature $\mathbf{F}_c$ is first fed into a 9×9 circular convolution layer [32] with 128 filters, followed by the ReLU and the batch normalization layer. Then, seven CIA units with three different kinds of dilation rates are employed to enhance the contour information using a multi-scale strategy, as different dilation rates have different receptive field sizes. After that, the outputs of the first batch normalization layer and all CIA units are concatenated and fused by a $1\times$ 1 1D convolution layer with 256 filters, followed by a max pooling operation. Finally, the global pooling feature is distributed to each contour point by concatenating their features.

In each CIA unit, the input feature $\mathbf{F}_u$ is first fed into the dilated circular convolution (dilation rate is r) to encode the cyclicity of points along the closed contour, as illustrated in Fig. 3. It would generate the features $\mathbf{F}_u^a \in \mathbb{R}^{N\times D_u}$, $\mathbf{F}_u^b \in \mathbb{R}^{N\times D_u}$, and $\mathbf{F}_u^c \in \mathbb{R}^{N\times D_u}$, where N is the number of contour points and D_u denotes the feature dimension. For the contour feature $\mathbf{F}_u^b$, we only employ N_g local sink nodes to gather the information along the contour, due to the redundancy and noise of the contour points. Actually, the global context of the contour also can be regarded as a global semantic sink node. It is concatenated with the features of local sink nodes to formulate the feature representation $\mathbf{F}_u^{bg} \in \mathbb{R}^{(N_g+1)\times D_u}$, which can be expressed as,

$$\mathbf{F}_u^{bg} = [\mathcal{P}_{max}(\mathbf{F}_u^b); \phi(\mathbf{F}_u^b, N_g)], \tag{6}$$

where $\mathcal{P}_{max}$ means the max pooling operation; ϕ denotes the feature gathering operation, which is achieved by a parameter-free strategy, *e.g.*, the adaptive average pooling. In the same way, we could obtain the aggregated feature $\mathbf{F}_u^{cg} \in \mathbb{R}^{(N_g+1)\times D_u}$ from $\mathbf{F}_u^c$. The relevance $\widetilde{\mathbf{R}}_u \in \mathbb{R}^{N\times(N_g+1)}$ between the contour points and the sink nodes is calculated as,

$$\mathbf{R}_u = \frac{1}{\sqrt{D_u}}\mathbf{F}_u^a \cdot (\mathbf{F}_u^{bg})^\top, \quad \widetilde{\mathbf{R}}_u^{ij} = \frac{\mathbf{R}_u^{ij}}{\sum_{i=1}^{N}\mathbf{R}_u^{ij}}, \tag{7}$$

where $\widetilde{\mathbf{R}}_u^{ij}$ denotes the relationship between the i-th contour point and the j-th sink node. Thus, the sink node features are distributed to the contour points for generating the aggregated feature $\mathbf{F}_u^{'} \in \mathbb{R}^{N\times D_u}$, which is expressed as,

$$\mathbf{F}_u^{'} = \mathbf{F}_u \oplus \widetilde{\mathbf{R}}_u \cdot \mathbf{F}_u^{cg}, \tag{8}$$

where $\oplus$ means the element-wise addition.

3.3. Reliable Contour Localization Mechanism

To increase the confidences of the detected contours of arbitrary-shape scene texts, we exploit a contour scoring mechanism in parallel with the contour localization to formulate a Reliable Contour Localization Mechanism (RCLM). Specifically, RCLM first feeds the evolved contour points into the contour feature extractor following the contour information aggregation module to generate the contour feature representation $\mathbf{F}_{cia}$, which is then fed into the contour localization branch to generate the final contour location $\widehat{\mathbf{X}}^{final}$. Meanwhile, the contour feature $\mathbf{F}_{cia}$ is also fed into the contour scoring mechanism to generate the contour score $\mathbf{s}$, which is denoted as,

$$\mathbf{s} = \varphi(\mathbf{F}_{cia}; \Theta_{csm}), \tag{9}$$

where φ means the contour scoring network, and Θ_{csm} is the corresponding network parameters. Specifically, in φ, the input $\mathbf{F}_{cia}$ is first fed into a 1×1 convolution layer with 256 filters, obtaining the feature representation $\mathbf{F}_{csm}$. Then the average pooling operation $\mathcal{P}_{avg}$ and the max pooling operation $\mathcal{P}_{max}$ are utilized to generate global feature representation $\mathbf{F}'_{csm}$, denoted as $\mathbf{F}'_{csm} = [\mathcal{P}_{avg}(\mathbf{F}_{csm}); \mathcal{P}_{max}(\mathbf{F}_{csm})]$. After that, three fully connected layers (the hidden sizes are 512, 256 and 2) and a softmax layer are stacked to generate final contour scores for text/non-text. Note that the former two fully connected layers are equipped with the LeakyReLU-BN-Dropout operation, where the slope of the Leaky ReLU is 0.2 and the dropout probability is 0.5.

To learn a robust contour scoring network, it requires positive samples and negative samples to train this network for distinguishing the contours of scene texts from those of backgrounds. Specifically, we treat the minimum bounding boxes of arbitrary-shape scene texts as positive samples. Furthermore, we exploit a shape-preserving negative sample mining technique to generate negative training samples. This negative sample mining technique first places the contour of each arbitrary-shape scene text on the image in a copy-move manner. Then, we calculate the overlaps between the generated contours and all positive contours. Af-

ter that, the generated contours are assigned to different bins based on overlaps, and we randomly choose a contour from the bin with the lowest interval of overlaps. Finally, the minimum bounding box of the selected contour is regarded as the negative sample.

3.4. Training and Inference

The proposed network is trained in an end-to-end manner, using the following total loss function,

$$\mathcal{L} = \mathcal{L}_{center} + \lambda_1 \mathcal{L}_{size} + \lambda_2 \mathcal{L}_{corner} + \lambda_3 \mathcal{L}_{evolution} + \lambda_4 \mathcal{L}_{csm}, \tag{10}$$

where λ_1, λ_2, λ_3 and λ_4 indicate the balance factors among the loss functions, which are set to 0.1, 1.0, 1.0 and 1.0 in our experiments. The ADAM optimizer [12] is utilized to train the proposed model.

In the inference stage, we first use a threshold τ_c to filter the center points with low scores on $\widehat{\mathbf{P}}$, like that in [61]. After we obtain the final localized contours, a threshold τ_a is employed to suppress the detected contours with low scores, before using the polygonal NMS [5] to reduce the overlapped contours.

4. Experiments

4.1. Datasets

CTW1500 [24] is an arbitrary-shape scene text dataset that consists of 1,000 images for training and 500 images for testing. In this dataset, the annotations of text instances are line-level and labeled by a polygon with 14 key points.

Total-Text [2] is another arbitrary-shape scene text benchmark that contains 1,255 training images and 300 testing images. All the text instances are annotated by the word-level polygon with adaptive number of key points.

ArT [3] is a large-scale multi-lingual arbitrary-shape scene text detection dataset. It includes 5,603 training images and 4,563 testing images. The text regions are annotated by the polygons with adaptive number of key points.

TD500 [55] is a scene text dataset for detecting arbitrary-oriented long texts. It consists of 300 training images and 200 testing images. The texts in these images are either English or Chinese scripts, annotated by rotated rectangles. The annotations of these texts are line-level.

4.2. Implementation Details

We employ DLA-34 [57] and ResNet-50 [9] pre-trained on ImageNet as the backbone. The parameters of our introduced layers are then initialized randomly. Alternatively, all the layers are initialized by the weights pre-trained on the training set and validation set of MLT-2017 [31] (9,000 images in total) for 20 epochs. σ, N_o, N_a, N_c, D_e, and D_u are set to 4, 64, 128, 4, 64, and 128, respectively. The initialized learning rate is fixed to 0.0001. For CTW1500, we train 250 epochs. For Total-Text, ArT and TD500, the model is trained for 300 epochs. The learning rate is decayed by 0.1 at 80, 120, 160, 180, and 260 epochs, respectively. In the training stage, we randomly crop original images into subimages with a size of 640 $\times$ 640. The cropped images are randomly rotated in $\{\ 0°, 90°, 180°, 270°\}$. Then, they are randomly flipped, blurred, and changed in color. The batch size is fixed to 6 and 3 for the model with the backbone DLA-34 and ResNet-50, respectively. 400 images in HUST400 [54] are added into the training set for TD500. In the testing stage, the batch size is fixed to 1. The shorter side of the test image is set to 416, 512, 640, and 640 for CTW1500, Total-Text, ArT, and TD500, respectively. The longer side is resized to keep the original aspect ratio. The threshold τ_c is set to 0.35 for CTW1500, ArT, and TD500, and 0.3 for Total-Text. Besides, τ_a is fixed to 0.9 for all the datasets. The proposed model is implemented based on Pytorch. All experiments are carried out on a workstation with a 4.00GHz Intel(R) Xeon(R) W-2125 CPU, a single NVIDIA GTX 2080Ti GPU, and 15G RAM.

4.3. Ablation Study

In this section, we conduct the ablation study on CTW1500. The model employs DLA-34 pre-trained on ImageNet as the backbone, and is directly trained on the training set of CTW1500. To verify the advantage of our model, we introduce a *baseline* that first employs the horizontal text proposal generation module to generate horizontal bounding boxes, and then evolves these boxes by the contour localization mechanism once.

As shown in Table 1, when the baseline model integrates the Oriented Text Proposal Generation (OTPG) network, it can improve 2.6% in terms of *F-measure*. The improvement could be ascribed that oriented text proposals have smaller offset variations for arbitrary-shape contours, which can facilitate the learning of contour regression. Then, the *F-measure* of the model has also increased by 2.6% and 0.7% after employing Contour Information Aggregation (CIA) and Reliable Contour Localization Mechanism (RCLM). If both CIA and RCLM are utilized, our model can achieve *Recall* of 81.3%, *Precision* of 86.1% and *F-measure* of 83.7%, which promotes 6.5% in *F-measure* compared with the baseline. These gains can be ascribed that CIA enriches the feature representations of text contours and RCLM filters some false positives by the predicted contour scores.

In the arbitrary-shape text generation, we employ the Contour Localization Mechanism (CLM) to deform the oriented text proposal contours to arbitrary-shape contours. CLM is similar to RCLM except that the former lacks the contour score mechanism. As the number K of CLM modules increases, the performance of our model boosts from 83.0% to 84.0% in *F-measure*. Table 2 demonstrates the ef-

Table 1: Ablation study on dataset *CTW1500*.

Module			R (%)	P (%)	F (%)
OTPG	CIA	RCLM			
×	×	×	73.9	80.9	77.2
√	×	×	78.2	81.5	79.8
√	√	×	80.1	84.7	82.4
√	×	√	77.8	83.4	80.5
√	√	√	81.3	86.1	83.7

Table 2: Influence of the number of CLM.

#CLM	R (%)	P (%)	F (%)	FPS
K = 0	80.3	86.0	83.0	16.3
K = 1	81.3	86.1	83.7	12.8
K = 2	81.1	87.1	84.0	11.8
K = 3	81.3	86.5	83.8	10.7

Table 3: Explorations of CIA. ▷ denotes 'replaced by'.

Strategy	R (%)	P (%)	F (%)
CIAU (N_g=0)	81.1	85.3	83.2
CIAU (N_g=4)	81.5	85.3	83.3
CIAU (N_g=8)	81.1	86.4	83.7
CIAU (N_g=16)	81.1	87.1	84.0
CIAU (N_g=32)	81.3	85.9	83.6
CIAU ▷ CirConv	80.0	85.4	82.6
CirConv ▷ StdConv (N_g=16)	81.3	86.0	83.6

Table 4: Comparisons with related works on *CTW1500*. 'Ext' means using the external dataset to pretrain the model.'Hybrid' denotes integrating the regression and segmentation in a framework.

Type	method	Venue	Backbone	Ext	R (%)	P (%)	F (%)
Segmentation-based	PAN [44]	ICCV'19	Res18	×	77.7	84.6	81.0
	TextSnake [27]	ECCV'18	VGG16	√	**85.3**	67.9	75.6
	MSR [53]	IJCAI'19	Res50	√	78.3	85.0	81.5
	PSENet [43]	CVPR'19	Res50	√	79.7	84.8	82.2
	CRAFT [1]	CVPR'19	VGG16	√	81.1	86.0	83.5
	LOMO [58]	CVPR'19	Res50	√	69.6	**89.2**	78.4
	SAE [38]	CVPR'19	Res50	√	77.8	82.7	80.1
	PAN [44]	ICCV'19	Res18	√	81.2	86.4	83.7
	SAST [42]	MM'19	Res50	√	77.1	85.3	81.0
	TextField [51]	TIP'19	VGG16	√	79.8	83.0	81.4
	DB [17]	AAAI'20	Res50-DCN	√	80.2	86.9	83.4
	DRRGN [59]	CVPR'20	VGG16	√	83.0	85.9	84.5
	CRNet [63]	MM'20	Res50	√	80.9	87.0	83.8
Hybrid	CSE [25]	CVPR'19	Res34	×	76.0	81.1	78.4
	ContourNet [48]	CVPR'20	Res50	×	84.1	83.7	83.9
	Mask-TTD [23]	TIP'20	Res50	×	79.0	79.7	79.4
	SD [49]	ECCV'20	Res50	√	82.3	85.8	84.0
Regression-based	SLPR [64]	ICPR'18	Res50	×	70.1	80.1	74.8
	CTD-CLOC [24]	PR'19	Res50	×	69.8	77.4	73.4
	ATRR [45]	CVPR'19	SE-VGG16	×	80.2	80.1	80.1
	TextRay [39]	MM'20	Res50	×	80.4	82.8	81.6
	ICG [36]	PR'19	VGG16	√	79.8	82.8	81.3
	Our PCR	—	Res50	×	79.8	85.3	82.4
	Our PCR	—	DLA34	×	81.1	87.1	84.0
	Our PCR	—	DLA34	√	82.3	87.2	**84.7**

fectiveness of the contour regression in a progressive manner. Meanwhile, the number of CLM modules also burdens the speed of the model. When adding three CLM modules, the runtime of the whole model drops from 16.3 FPS to 10.7 FPS. To make a trade-off between the performance and the speed, we set K = 2 in the following experiments.

In CIA, the change of the sink node number N_g in each CIA Unit (CIAU) would affect the performance of our model, as listed in Table 3. Specifically, when N_g increases to 16, the model achieves the optical *F-measure*. The improvement could be ascribed that more sink nodes can gather more representative features to some extent. When each CIAU is replaced by one circular convolution (termed as 'CirConv') layer, the performance of the model would decrease by 1.1%, 1.7%, and 1.4% in terms of *Recall*, *Precision*, and *F-measure*, respectively. It reveals that the effectiveness of the semantic sink nodes in capturing valid contexts of contours. If we utilize the standard convolution (termed as 'StdConv') layer to replace all 'CirConv' layers in CIA, the performance of our model drops from 87.1% to 86.0% in *F-measure*. The reason is that the standard convolution can not encode the circular topology of text contours.

4.4. Comparisons with State-of-the-art Methods

Evaluation on CTW1500: As shown in Table 4, our model achieves the best performance, compared with all previous state-of-the-art methods. Specifically, our method outperforms the regression-based methods (*e.g.*, SLPR [64], ATRR [45], CTD-CLOC [24] and TextRay [39], ICG [36]) by large margins. When comparing with PSENet [43] that segments text regions using a progressive scale expansion, our progressive contour regression mechanism has promoted *Recall* of 2.6%, *Precision* of 2.4% and *F-measure* of 2.5%. Besides, our performance also increases by 12.7% and 6.3% in *Recall* and *F-measure*, compared with LOMO [58] that also localizes texts progressively. The qualitative detection results are displayed in Fig. 4 (a).

Evaluation on Total-Text: As shown in Table 5, our method is obviously superior to the regression-based methods, *e.g.*, ATRR [45], Boundary [41], TextRay [39], and Poly-FRCNN [2]. When comparing with Boundary [41] that integrates the recognition branch to guid the detection learning, our method still has an improvement of 3.3% in *Precision*, without the labor of designing anchors. The experimental results also reveal that the performance of our model outperforms the hybrid-based methods, *e.g.*, FTSN [5], Mask-TextSpotter-v2 [13], SPCNet [50], MS-CAFA [4], Mask-TTD [23], etc. For example, our method significantly boosts the *Recall* of 6.6%, *Precision* of 6.7% and *F-measure* of 6.7%, compared with the well-known model Mask-TextSpotter-v2. The qualitative detection results of our model can be seen in Fig. 4 (b).

Evaluation on ArT: As shown in Table 6, our model can boost the *F-measure* from 66.2% to 73.1%, compared with the recent contour point regression based method TextRay [39]. After employing the external dataset to pretrain the model, it brings a 7.8% improvement in *F-measure*. Fig. 4 (c) shows the qualitative detection results of our model.

Evaluation on TD500: As shown in Table 7, our method can achieve the *F-measure* of 87.0%, outperforming al-

Table 5: Comparisons with related works on *Total-Text*. 'Ext' means using the external dataset to pretrain the model. † denotes the end-to-end scene text spotting.

Type	method	Venue	Backbone	Ext	R (%)	P (%)	F (%)
Segmentation-based	PAN [44]	ICCV'19	Res18	×	79.4	88.0	83.5
	TextSnake [27]	ECCV'18	VGG16	√	74.5	82.7	78.4
	LOMO [58]	CVPR'19	Res50	√	75.7	88.6	81.6
	PSENet [43]	CVPR'19	Res50	√	78.0	84.0	80.9
	CRAFT [1]	CVPR'19	VGG16	√	79.9	87.6	83.6
	MSR [53]	IJCAI'19	Res50	√	74.8	83.8	79.0
	PAN [44]	ICCV'19	Res18	√	81.0	**89.3**	85.0
	TextDragon [7]†	ICCV'19	VGG16	√	75.7	85.6	80.3
	SAST [42]	MM'19	Res50	√	76.9	83.8	80.2
	TextField [51]	TIP'19	VGG16	√	79.9	81.2	80.6
	DB [17]	AAAI'20	Res50-DCN	√	82.5	87.1	84.7
	CRNet [63]	MM'20	Res50	√	82.5	85.8	84.1
Hybrid	CSE [25]	CVPR'19	Res34	×	79.1	81.4	80.2
	Mask-TTD [23]	TIP'20	Res50	×	74.5	79.1	76.7
	FTSN [5]	ICPR'18	Res101	√	78.0	84.7	81.3
	Mask-TextSpotter [28] †	ECCV'18	Res50	√	55.0	69.0	61.3
	SPCNet [50]	AAAI'19	Res50	√	82.8	83.0	82.9
	Mask-TextSpotter-v2 [13] †	TPAMI'19	Res50	√	75.4	81.8	78.5
	MS-CAFA [4]	TMM'20	Res50	√	78.6	84.6	81.5
Regression-based	ATRR [45]	CVPR'19	SE-VGG16	×	76.2	80.9	78.5
	CTC-CLOC [24]	PR'19	Res50	×	71.0	74.0	73.0
	TextRay [39]	MM'20	Res50	×	77.9	83.5	80.6
	ICG [36]	PR'19	VGG16	√	80.9	82.1	81.5
	Boundary [41]†	AAAI'20	Res50	√	**83.5**	85.2	84.3
	Poly-FRCNN [2]	IJDAR'20	Inc-Res-v2	√	68.0	78.0	73.0
	MS-CAFA [4]	TMM'20	Res50	√	78.6	84.6	81.5
	Our PCR	—	Res50	×	80.2	86.1	83.1
	Our PCR	—	DLA34	×	81.5	86.4	83.9
	Our PCR	—	DLA34	√	82.0	88.5	**85.2**

Table 6: Comparisons with related works on *ArT*.

Method	Venue	Ext	R (%)	P (%)	F (%)
TextRay [44]	MM'20	√	58.6	76.0	66.2
Ours (DLA-34)	—	×	65.0	83.6	73.1
Ours (DLA-34)	—	√	**66.1**	**84.0**	**74.0**

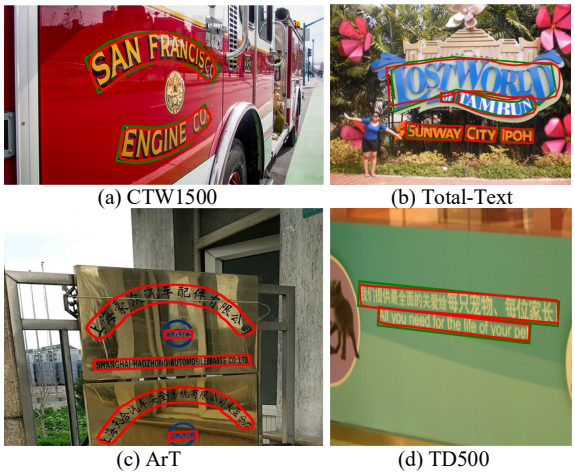
l existing well-known ones. Especially compared with the anchor-free pioneer work EAST [62] for detecting multi-oriented scene texts, our method achieves the improvement of 16.1%, 3.5%, and 10.9% in *Recall*, *Precision*, and *F-measure*, respectively. Besides, when comparing with the naive corner-based method [29] that involves heuristic rules to group the predicted corners, our model also promotes the *F-measure* from 81.5% to 87.0%. Fig. 4 (d) shows some multi-oriented scene text detection results.

5. Conclusion

In this paper, we present an end-to-end trainable contour-based regression framework to detect arbitrary-shape texts in the natural image. Our model first regresses the contour of horizontal text proposals generated by an anchor-free network to the corner points of oriented text proposals, and then evolves the contour of oriented text proposals to the contour of arbitrary-shape texts. In the progressive regression process, the contour information aggregation technique is utilized to enrich the feature representation of contours by considering the circular geometric topology and semantic sink nodes of the text contour. Meanwhile, a reliable contour localization mechanism is integrated to relieve the false positives by the predicted contour confidence. The effectiveness and superiority of our model have been validated on four public benchmarks including curved, wavy, long, oriented and multilingual scene texts. In the future, we would like to integrate a recognition network to develop an end-to-end scene text spotting system.

Table 7: Comparisons with related works on *TD500*.

Type	method	Venue	Backbone	Ext	R (%)	P (%)	F (%)
Segmentation-based	EAST [62]	CVPR'17	PVANet	×	67.4	87.3	76.1
	PixelLink [6]	AAAI'18	VGG16	×	73.2	83.0	77.8
	Border [52]	ECCV'18	DesNet121	×	77.4	83.0	80.1
	TextSnake [27]	ECCV'18	VGG16	√	73.9	83.2	78.3
	MSR [53]	IJCAI'19	Res50	√	76.7	87.4	81.7
	CRAFT [1]	CVPR'19	VGG16	√	78.2	88.2	82.9
	SAE [38]	CVPR'19	Res50	√	81.7	84.2	82.9
	PAN [44]	ICCV'19	Res18	√	**83.8**	84.4	84.1
	TextField [51]	TIP'19	VGG16	√	75.9	87.4	81.3
	DB [17]	AAAI'20	Res50-DCN	√	79.2	**91.5**	84.9
	CRNet [63]	MM'20	Res50	√	82.0	86.0	84.0
	DRRGN [59]	CVPR'20	VGG16	√	82.3	88.1	85.1
Hybrid	DSRN [47]	IJCAI'19	Res50	×	71.2	87.6	78.5
	FTSN [5]	ICPR'18	Res101	√	77.1	87.6	82.0
	Corner [29]	CVPR'18	VGG16	√	76.2	87.6	81.5
	Mask-TextSpotter-v2 [13] †	TPAMI'19	Res50	√	68.6	80.8	74.2
	Mask-TextSpotter-v3 [14] †	ECCV'20	Res50	√	77.5	90.7	83.5
Regression-based	RRPN [30]	TMM'18	VGG16	×	69.0	82.0	75.0
	ATRR [45]	CVPR'19	SE-VGG16	×	82.1	85.2	83.6
	SegLink [34]	CVPR'17	VGG16	√	70.0	86.0	77.0
	RRD [18]	CVPR'18	VGG16	√	73.0	87.0	79.0
	Our PCR	—	Res50	×	77.8	87.6	82.4
	Our PCR	—	DLA34	×	79.2	90.0	84.3
	Our PCR	—	DLA34	√	83.5	90.8	**87.0**



(a) CTW1500 (b) Total-Text

(c) ArT (d) TD500

Figure 4: Qualitative detection results of our proposed method. Red denotes the detection result. Green means the ground-truth. Note that we do not access the ground-truth of ArT. More results are shown in supplementary materials.

Acknowledgements: This work was supported by the National Key R&D Program of China (Grant No. 2020YFB1406704), National Natural Science Foundation of China (No. 62025604, 61733007, 62072454, U1936210, 61971016), Beijing Natural Science Foundation (No. 4202084), Beijing Municipal Education Commission Cooperation Beijing Natural Science Foundation (No. KZ 201910005007) and Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019. 1, 2, 7, 8

[2] Chee-Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-Text: toward orientation robustness in scene text detection. *IJDAR*, 23(1):31–52, 2020. 2, 3, 6, 7, 8

[3] Chee Kheng Chng, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, and et al. ICDAR2019 robust reading challenge on arbitrary-shaped text - rrc-art. In *ICDAR*, pages 1571–1576, 2019. 6

[4] Pengwen Dai, Hua Zhang, and Xiaochun Cao. Deep multi-scale context aware feature aggregation for curved scene text detection. *IEEE Trans. Multimedia*, 22(8):1969–1984, 2020. 2, 3, 7, 8

[5] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *ICPR*, pages 3604–3609, 2018. 2, 3, 6, 7, 8

[6] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. PixelLink: Detecting scene text via instance segmentation. In *AAAI*, pages 6773–6780, 2018. 1, 2, 8

[7] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. TextDragon: An end-to-end framework for arbitrary shaped text spotting. In *ICCV*, pages 9075–9084, 2019. 1, 2, 8

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[10] Ziyang Hong, Yvan Petillot, David Lane, Yishu Miao, and Sen Wang. TextPlace: Visual place recognition and topological localization through reading scene texts. In *ICCV*, pages 2861–2870, 2019. 1

[11] Sezer Karaoglu, Ran Tao, Theo Gevers, and Arnold W. M. Smeulders. Words Matter: Scene text for image classification and retrieval. *IEEE Trans. Multimedia*, 19(5):1063–1076, 2017. 1

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[13] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, In Press, 2019. 2, 3, 7, 8

[14] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask TextSpotter v3: Segmentation proposal network for robust scene text spotting. In *ECCV*, pages 706–722, 2020. 8

[15] Minghui Liao, Baoguang Shi, and Xiang Bai. TextBoxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Process.*, 27(8):3676–3690, 2018. 3

[16] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. TextBoxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017. 3

[17] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020. 1, 2, 7, 8

[18] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *CVPR*, pages 5909–5918, 2018. 3, 8

[19] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, pages 5257–5266, 2019. 3

[20] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, pages 9806–9815, 2020. 2

[21] Yuliang Liu and Lianwen Jin. Deep Matching Prior Network: Toward tighter multi-oriented text detection. In *CVPR*, pages 3454–3461, 2017. 3

[22] Yuliang Liu, Lianwen Jin, and ChuanMing Fang. Arbitrarily shaped scene text detection with a mask tightness text detector. *IEEE Trans. Image Process.*, 29:2918–2930, 2020. 2, 3

[23] Yuliang Liu, Lianwen Jin, and ChuanMing Fang. Arbitrarily shaped scene text detection with a mask tightness text detector. *IEEE Trans. Image Process.*, 29:2918–2930, 2020. 7, 8

[24] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognit.*, 90:337–345, 2019. 2, 3, 6, 7, 8

[25] Zichuan Liu, Guosheng Lin, Sheng Yang, Fayao Liu, Weisi Lin, and Wang Ling Goh. Towards robust curve text detection with conditional spatial expansion. In *CVPR*, pages 7269–7278, 2019. 7, 8

[26] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. 129(1):161–184, 2021. 2

[27] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. TextSnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, pages 19–35, 2018. 1, 2, 7, 8

[28] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, pages 71–88, 2018. 3, 8

[29] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, pages 7553–7563, 2018. 8

[30] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia*, 20(11):3111–3122, 2018. 3, 8

[31] Nibal Nayef, Fei Yin, Imen Bizid, and et al. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In *ICDAR*, pages 1454–1459, 2017. 6

[32] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *CVPR*, pages 8530–8539, 2020. 3, 5

[33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 4

[34] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, pages 3482–3490, 2017. 2, 8

[35] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.*, 30(1):32–46, 1985. 2

[36] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. SegLink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recognit.*, 96, 2019. 2, 7, 8

[37] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72, 2016. 2

[38] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *CVPR*, pages 4234–4243, 2019. 1, 2, 7, 8

[39] Fangfang Wang, Yifeng Chen, Fei Wu, and Xi Li. TextRay: Contour-based geometric modeling for arbitrary-shaped scene text detection. In *ACM-MM*, pages 111–119, 2020. 2, 3, 7, 8

[40] Fangfang Wang, Liming Zhao, Xi Li, Xinchao Wang, and Dacheng Tao. Geometry-aware scene text detection with instance transformation network. In *CVPR*, pages 1381–1389, 2018. 3

[41] Hao Wang, Pu Lu, Hui Zhang, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. All You Need Is Boundary: Toward arbitrary-shaped text spotting. In *AAAI*, pages 12160–12167, 2020. 2, 3, 7, 8

[42] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. A single-shot arbitrarily-shaped text detector based on context attended multi-task learning. In *ACM-MM*, pages 1277–1285, 2019. 1, 2, 7, 8

[43] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, pages 9336–9345, 2019. 1, 2, 7, 8

[44] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, pages 8439–8448, 2019. 1, 2, 7, 8

[45] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary shape scene text detection with adaptive text region representation. In *CVPR*, pages 6449–6458, 2019. 2, 3, 7, 8

[46] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *CVPR*, pages 10123–10132, 2020. 1

[47] Yuxin Wang, Hongtao Xie, Zilong Fu, and Yongdong Zhang. DSRN: A deep scale relationship network for scene text detection. In *IJCAI*, pages 947–953, 2019. 8

[48] Yuxin Wang, Hongtao Xie, Zhengjun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *CVPR*, pages 11750–11759, 2020. 7

[49] Shanyu Xiao, Liangrui Peng, Ruijie Yan, Keyu An, Gang Yao, and Jaesik Min. Sequential deformation for accurate scene text detection. In *ECCV*, pages 108–124, 2020. 3, 7

[50] Enze Xie, Yuhang Zang, Shuai Shao, Gang Yu, Cong Yao, and Guangyao Li. Scene text detection with supervised pyramid context network. In *AAAI*, pages 9038–9045, 2019. 2, 3, 7, 8

[51] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. TextField: Learning A deep direction field for irregular scene text detection. *IEEE Trans. Image Process.*, 28(11):5566–5579, 2019. 1, 2, 7, 8

[52] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *ECCV*, pages 370–387, 2018. 8

[53] Chuhui Xue, Shijian Lu, and Wei Zhang. MSR: multi-scale shape regression for scene text detection. In *IJCAI*, pages 989–995, 2019. 1, 2, 7, 8

[54] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Trans. Image Process.*, 23(11):4737–4749, 2014. 6

[55] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090, 2012. 6

[56] Qixiang Ye and David S. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(7):1480–1500, 2015. 2

[57] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018. 6

[58] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look More Than Once: An accurate detector for text of arbitrary shapes. In *CVPR*, pages 10552–10561, 2019. 2, 7, 8

[59] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *CVPR*, pages 9696–9705, 2020. 1, 2, 7, 8

[60] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *CVPR*, pages 4159–4167, 2016. 2

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. 3, 4, 6

[62] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In *CVPR*, pages 2642–2651, 2017. 8

[63] Yu Zhou, Hongtao Xie, Shancheng Fang, Yan Li, and Yongdong Zhang. CRNet: A center-aware representation for detecting text of arbitrary shapes. In *ACM-MM*, pages 2571–2580, 2020. 1, 2, 7, 8

[64] Yixing Zhu and Jun Du. Sliding line point regression for shape robust scene text detection. In *ICPR*, pages 3735–3740, 2018. 2, 3, 7