PolarText: Single-stage Scene Text Detection with Polar Representation

Qiran Kong
College of Computer and Information
Hohai University
Nanjing, China
Email: kds809545917@gmail.com

Yirui Wu*
College of Computer and Information
Hohai University
Nanjing, China
Email: wuyirui@hhu.edu.cn

Shaohua Wan
School of Information and Safety Engineering
Zhongnan University of Ecnomics and Law
Wuhan, China
Email: shaohua.wan@ieee.org

2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC) | 978-1-6654-0036-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/EUC53437.2021.00030

***Abstract*—Although deep learning has achieved great success in object detection recently, scene text detection is still a challenging task, due to inherent difficulties of locating texts in complex scenes. Many approaches adopt inspirations from segmentation to detect arbitrary shaped scene text. However, most segmentation based methods have high computation cost and generally needs a lot of refinements to get accurate results. To ease this problem, we propose a novel single-stage method, i.e., PolarText network, which detects text regions by generating contour points in polar coordinates. PolarText not only relieves the burden of high computation cost by directly regressing contour points instead of pixels, but also fits with intrinsic characteristics of text instances by centers and contours, thus suppressing mislabeling boundary pixels caused by pixel-level labeling. To cope with polar representation, PolarText utilizes Polar IoU loss and polar centerness to generalize effective paradigms from box representation for polar representation. In addition, we add a dedicated bounding box branch to work with text detection since most text instances are approximately rectangular in shape. Compared with the existing methods, the proposed method achieves superior results in both accuracy and efficiency by testing on CTW 1500 and ICDAR 2015 datasets.**

I. Introduction

The goal of scene text detection is to spot text regions in images of natural scenarios, which is of great importance for task of scene understanding. Even though deep learning has made great progress in understanding images and videos, it's still challenging to know texts from scene images. The difficulty comes in several ways. Firstly, the texture of text regions and backgrounds in natural scenarios are complicated, leading to quantity of miscalculations. Secondly, shapes of text regions are arbitrary, resulting in hardness to accurately detect, especially for curved and rotated text.

Facing these two difficulties, researchers have proposed many methods to detect texts in the wild, which can be classified into two categories, i.e., regression and segmentation based methods. The former one aims to detect text instance as a common object, meanwhile the latter one obtain masks and bounding boxes according to text instances progressively. Since regression based methods could detect horizontal objects and non-horizontal objects with accurate performance, they generally require additional algorithm design and computing power to solve the problem of rotated texts. Segmentation based methods are capable to deal with situations of oriented and curved texts by generating pixel-level segmentation map and mask, which makes segmentation based methods one of the most progressing trend in solving challenges of locating texts in scene.

Based on directly generating pixel-level labels or not, we classify current segmentation based methods into two categories, i.e., bottom-up and top-down. The former ones regard text detection as a semantic segmentation problem by directly assigning pixel-level labels to text or non-text regions. For example, Wang et al. [1] proposed a novel Progressive Scale Expansion Network (PSENet), which is designed as a segmentation-based detector with multiple predictions for each text instance. However, bottom-up methods could easily result in mislabeling in boundary pixels due to sticky texts or low distinguish ability of generated feature map.

Top-down methods transform text detection to task of instance segmentation by detecting rectangular bounding boxes containing texts at first, and then perform pixel-level segmentation inside boxes. For example, Huang et al. [2] present a new Mask R-CNN based text detection approach, which could robustly detect multi-oriented and curved text from natural scene images in a unified manner. However, bounding boxes couldn't work well when rotated text instances are close to each other. Moreover, top-down methods have to use dense anchors to refine bounding boxes, which brings quantity of parameters to determine, thus greatly slowing computation speed.

Facing disadvantages brought by both categories of segmentation-based methods, this paper proposes a novel idea to detect texts by directly generating centers and contour points of text instances with polar coordinates representation. By focusing on generating centers and contours of text regions, the proposed method not only relieves the burden of generating quantity of pixel-level labels in bottom-up methods, but also involves efficient representation of text regions with center and several contours rather than pixel-level labeling, fitting with intrinsic characteristics of text instances. Compared with top-down methods, we adopt one-stage structure without steps to refine bounding boxes, which saves large computation resource. To sum up, we believe the proposed polar coordinates representation offers text-specified representation for text instances with less target points to regress.

978-1-6654-0036-7/21/$31.00 ©2021 IEEE
DOI 10.1109/EUC53437.2021.00030

We thus propose a novel single-stage method, i.e., PolarText network, which is capable to detect text regions by generating contour points in polar coordinates. By involving idea of polar coordinates representation, PolarText not only relieves the burden of high computation cost by directly regressing contour points instead of pixels, but also fits with intrinsic characteristics of text instances by centers and contours, thus suppressing mislabeling boundary pixels caused by pixel-level labeling. To cope with polar representation, we further propose Polar IoU loss and polar centerness to generalize effective paradigms from box representation for polar representation. Since most text instances are shaped in rectangular, we additionally add a bounding box branch. To show the effectiveness of our proposed method, we conduct extensive experiments on several challenging benchmark datasets including CTW1500, Total-Text, ICDAR 2015, ICDAR2017 and MSRA-TD500. Our main contribution could be concluded as follows:

- The proposed PolarText accurately detects text regions by generating their centers and counter points under polar coordinates, which not only relieves burden of high computation cost brought by pixel-level classification, but also fits with intrinsic characteristics of text instances to eliminate mislabeled boundary pixels.
- We specially design polar IoU loss function and polar centerness, which helps fully take advantage of our polar representation and enables us to generalize these effective methods in polar representation, making our PolarText easier and faster to train.
- We specially add a bounding box branch and design cIoU loss function, which is very suitable for text detection. Our auxiliary bounding box branch not only helps PolarText converge faster by considering aspect ratios as factors, but also better locates the predicted bounding boxes that have no overlap regions with ground-truth.

II. Related work

In this section we give an introduction to the related works that inspire us. We divide them into two categories, i.e., scene text detection and single-stage detection.

A. Scene Text Detection

To detect text in arbitrary shape, the mainstream methods are based on segmentation. As mentioned in the above section, we divide segmentation based methods into two categories, namely, bottom-up and top-down. Both categories have their advantages and disadvantages. Top-down methods mainly borrow the idea from object detections and instance segmentation, an obvious example is Mask R-CNN[3]. Early, Mask R-CNN firstly modify the step of ROI pooling to ROI align on the basis of faster R-CNN, and then add a mask module for accurate instance segmentation. Many methods built on top of Mask R-CNN achieved good results. For example, Huang et al. [2] proposed a new Mask R-CNN based text detection approach, which could detect multi-oriented and curved text robustly from images in natural scenarios in a unified manner. Xie et al. [4] added spatial and channel attention mechanism to Mask R-CNN in order to deal with complex scenarios. Although the high accuracy of top down methods, the greatest disadvantage is that they have a lot of computations on dense generated predefined anchors, which greatly slows their speed. Further more, top-down methods depend completely on the detection boxes, which will affect the accuracy when rotated boxes are close to each other [5].

Bottom-up methods regard text detection as a semantic segmentation problem by directly assigning pixel-level labels to text or non-text regions. In order to distinguish text instances, these methods often use text center line. TextSnake[6] uses ordered disks and text center lines to represent text instances, which is able to model text in arbitrary shapes. PSENet [1] uses FCN to predict text instances directly with multiple scales, to reconstruct the whole text instance, a progressive method is adopted to determine which text instance a pixel belongs to. The accuracy of bottom up methods are mainly affected by two folds, the accuracy of the output of the semantic segmentation, the text instances reconstruction. Usually, it's challenging to accurately segment text instances directly, so the effectiveness of bottom-up methods is not as high as top-down methods [7].

Most recently, TextFuseNet [8] obtains richer text features by fusing three different categories of features, i.e., character level, word level and global level. Rich features enhance the detection ability and environmental adaptability of their proposed network. Owing to the guidance of semantic information, SPCNet [9] propose to involve more context information, resulting in stronger detection capabilities in complex natural scenes. Afterwards, ContourNet [10] generates more accurate anchors through Adaptive-RPN, and uses Local Orthogonal Texture-aware Module model the local texture information in two orthogonal directions, which successfully reduces false positive results.

B. Single-Stage Detector

To relieve the high computation and optimization burden brought by multiple stage detector, researchers are interested in developing fast and accurate one-stage detector. Unlike two-stage detectors like Mask R-CNN that can gradually refine the predictions, directly generating detections in one stage is a great challenge We introduce single-stage detectors for predicting bounding boxes and masks respectively [11].

We first introduce single-stage methods that are used to generate rectangular boxes. YOLO [12] is short for You Only Look once, It divides an image into multiple grids and each grid is responsible for predicting the boxes whose center is located in that gird. SSD [13] stands for Single Shot MultiBox Detector that does a dense prediction on the entire image without the need of region proposals. FCOS[14] adopts a divide and conquer strategy which makes different levels of feature maps responsible for different sizes of boxes. However, these methods could only be used to predict bounding boxes in one-stage. Our proposed PolarText could generate both bounding boxes and masks in one stage [15].

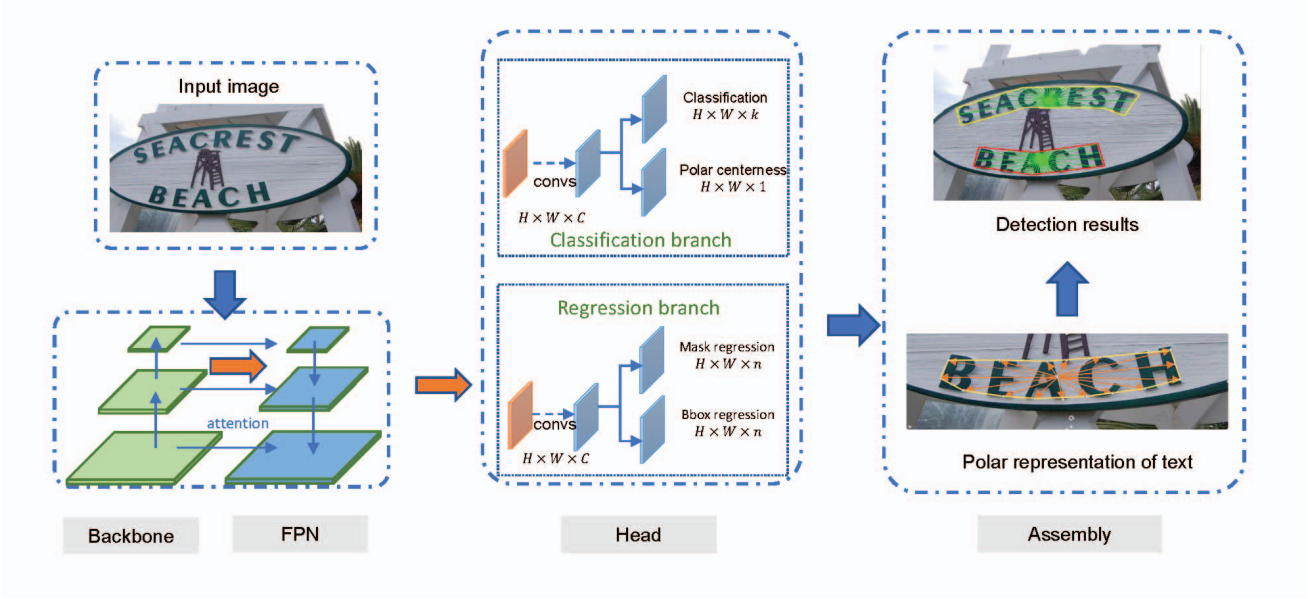

Fig. 1. The overall network structure of PolarText, which consists of backbone, head, polar representation and results.

To generate instance masks in arbitrary shape, early, deep Watershed Transform [16] first uses fully convolutional network to predict the energy map of the entire image, and then use the watershed algorithm to generate connected object instances from the energy map. The recent YOLACT[17] introduces a new concept called prototype masks that do no depend on any individual instances. The resulting instances are generated by the linear combination of these prototype masks in real time. TensorMask [18] investigates the paradigm of dense sliding window instance segmentation instead of sliding windows bounding boxes, which makes the output at every spatial location is itself a geometric structure with its own spatial dimension. CenterMask [19] first predicts bounding boxes together with box centerness on each location, instances segmentation is based on these predicted bounding boxes. To further improve the performance, a novel backbone named vovNet was also proposed in this paper. SOLO [20] reformulate the instance segmentation as a combination of category prediction and instance mask generation, it generates pixel segmentation masks instead of bounding boxes. Densely predicting masks requires large computation, so the inference speed of SOLO is very slow. Unlike existing methods that generate instance masks with pixel level labeling, our PolarText directly outputs masks with polar representation, thus our proposed method reformulate mask generation as a regression task instead.

III. Proposed Method

In this section we introduce our proposed method. We firstly introduce our light-scale network architecture for the task of text detection. Then, we introduce a task-specified module to perform computation for representation in polar coordinates. After that, we introduce our proposed assembly module to generate text instances, using polar centerness and polar distance regression. Finally, we describe the detail information of the loss function design, including Polar loss and cIoU loss design.

A. Network Architecture Design

Our proposed Polar Text reformulate text detection as two sub tasks, locating the center of a text instance and predicting the distance of the contour points from the text center. The overall structure of the proposed PolarText is illustrated in Fig.1, where we can notice the input image is firstly fed into the backbone, typically ResNet[21]. After processing of the attention module, we generate the feature maps on different levels via FPN (Feature pyramid network) [22]. All these processes can be defined as

$$F = Backbone(I), where\ F = \{F_i, i = 1, ..., n\} \quad (1)$$

Where I is the input image, i refers to the level of the output feature map via FPN, $backbone$ refers to the backbone operation with multiple levels.

Different sizes of instances are separated on different levels of feature maps. Therefore, large feature maps that have more global context information will be used to predict small text instances while smaller feature maps that have coarse spatial information will be used to predict large text instances. After these results are fed into the head, we made predictions on every pixel of the feature maps. There are 4 parallel branches in our head, including classification, polar centerness, mask and box regression:

For a feature map whose dimension is $H \times W \times C$, we get a $H \times W \times nrays$ dimension O_{polar} and a $H \times W \times 4$ dimension O_{box}, where $nrays$ refers to the total number of rays emitted from every pixel on the feature map, each of

$nrays$ dimensions correspond to a specific angle, the $nrays$ distances can be used to reconstruct the mask of a text instance. Since most of the text instances are shaped like rectangles, we still have a dedicated box regression branch. The horizontal and vertical rays will be averaged with the edges in the corresponding direction on the box. With this help, our network will pay attention on specific angles instead of treating all directions equally.

While in the classification branch, our output of classification logits has the dimension of $H \times W \times k$, where k is the number of classes and equals to 2 in case of text detection. In addition to the classification logits, our network can has a parallel branch named polar centerness, whose dimension is $H \times W \times 1$. Polar centerness is used to determine whether a pixel is near the center of a text instance. We use polar centerness and classification logits together to filter low quality predictions, this will be discussed later. Our final classification score which will be used for post-processing like NMS is the multiplication of polar centerness and the classification logits:

$$O_{final} = O_{cls} * O_{center} \tag{2}$$

where, O_{final} is our final classification score, $*$ denotes the element wise product. To cope with polar representation, PolarText utilizes Polar IoU loss and polar centerness to generalize effective paradigms from box representation for polar representation. After that, we assemble these results together for post-processing, to get the final detection results:

$$R = A(O_{final}, O_{polar}, O_{box}) \tag{3}$$

where A denotes our Assembly module, $R = \{D_i, i = 1, 2..., N\}$ is our detection results.

Supposing a training set with N pairs of Network predictions and corresponding ground truth labels represented byD_i and G_i respectively, the overall network Loss function could be defined as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(D_i, G_i) + L_{centerness}(D_i, G_i) + L_{polar}(D_i, G_i) + L_{box}(D_i, G_i) \tag{4}$$

where, L_{cls}, $L_{centerness}$, L_{polar}, L_{box} are the loss functions of classification logits, polar centerness, polar regression, bounding box regression respectively, θ denotes the set of network parameters.

B. Representation in Polar Coordinates

Inspired by PolarMask [23] to enhance feature representation for better object detection results, we believe it's essential to offer abundant and enhanced representation in polar coordinates for accurate text detection. We thus design a task-specified computation module to perform such task.

We represent text instances as a set of contour points in polar coordinates. The contour points can be determined by the distance and angle from polar center and we can easily reconstruct text instances via these contour points. Starting from the polar center, we emit n rays uniformly. The number of rays is a hyper parameter (we set to 36) so these rays have a fixed angle interval. Only distances are needed to generate the contour points. With a polar center whose coordinate is (c_x, c_y) and n different lengths, the coordinates of the contour points (x_i, y_i) $i = 1, 2, ..., n$ could be easily calculated:

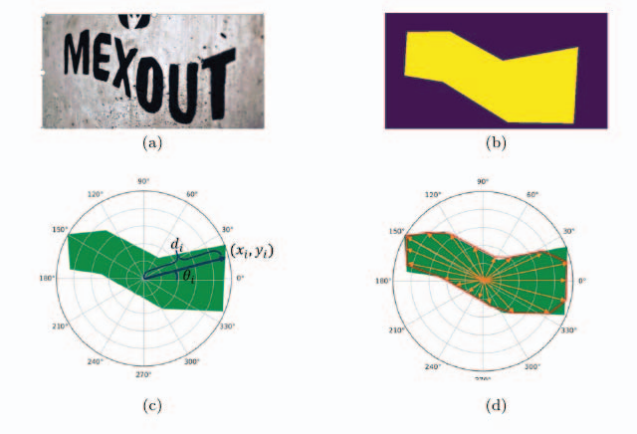

Fig. 2. Representation in polar Coordinates: (a) Picture with text instance. (b) The generated Text mask. (c) Calculate the coordinate of the contour point with distance predicted. (d) Mask segmentation with polar representation

$$x_i = \cos\theta_i \times d_i + x_c \tag{5}$$

$$y_i = \sin\theta_i \times d_i + y_c \tag{6}$$

where i denotes the ith contour point with the corresponding angle θ. Figure 2 illustrates the process to represent a text instance in polar coordinates. Where we first locate the polar center, then we calculate the contour points corresponding to a specific angle via the distance, then we repeat this n times to get all the contour points for generating text mask in arbitrary shape. To find regression targets for these rays, for each ground truth, we sample a point and the emit rays from that center, with these lines, we could get the crossover points with the contour of the ground truth. The distance between these crossover points and the sample point is the regression target. If there are more than one point that intersect with the ray, we simply consider the point that has the largest distance from the sample point as the contour point for distance regression. If the center point is located outside the mask, which is rarely the case, we set the regression target as the minimum value. A location is considered as a sample point only if it is near the mass center of a text instance so we can make sure that we always sample from the polar center, for those points away from the mass center, we simply ignore them.

C. Assembly

We assembly the outputs of the heads to get the detection results for post-processing. Thus, a total of $H * W$ text instances with corresponding masks and bounding boxes will be generated, where (H, W) denotes the shape of the corresponding feature map. The polar representation of text masks

is introduced above, we emit rays to get contour points and use these contour points to reconstruct our text masks. Bounding box generation is similar, with a box center and the lengths of different edges, the corresponding bounding boxes could be easily determined. When locating a set of center points of text instances, we can easily get a set of detection results $D_i, i = 1, 2, ...N$.

To determined the center points of text instances, our network also outputs classification and centerness on each pixel. To make the predicted points more consistent with the regression targets, we consider only the pixels near the mass center of text instances as positive examples, which is approximately $1.5\times$ the stride of the corresponding feature map. The output classification logit is the probability whether the corresponding pixel is a positive sample, ie. located near the mass center. Polar centerness is used to suppress a lot of low-quality outputs which is defined as:

$$Polar\ Centerness = \sqrt{d_{min}/d_{max}} \tag{7}$$

where d_{min} is the minimum length of the n rays and d_{max} is the maximum, after that, we use square root to normalize. We multiply the classification logits with the corresponding centerness score to get the final scores. Then we use these results to filter low-quality outputs by only keeping 1 k top-scoring predictions at most on each feature map, any output that is lower than 0.05 is also directly filtered. NMS then is used to further filter outputs.

D. Loss Function Design with Polar Loss and cIoU Loss

Specifically, we use polar IoU loss for mask regression in polar coordinates and cIoU loss for Bbox regression. Directly applying mask loss on pixel level is not optimal. Since our masks are represented in polar coordinates, it needs a lot of computation to reconstruct the mask instances in pixel level and calculate losses per pixel level. Smooth-l_1 doesn't consider the prediction as a whole because it takes different rays separately and overlooks the correlations among them. Thus, we use a specially designed Polar IoU loss that fits in our polar representation best. Given two sets $d = \{d_1, d_2, d_3, ..., d_n\}$ as the lengths of n predicted rays and $d^* = \{d_1^*, d_2^*, d_3^*, ..., d_n^*\}$ as the n ground truth lengths. Polar IoU loss is an approximation of IoU loss in polar coordinates, it is defined as:

$$\mathcal{L}_{PolarIoU} = \log \frac{\sum_{i=1}^{n} \max(d_i, d_i^*)}{\sum_{i=1}^{n} \min(d_i, d_i^*)} \tag{8}$$

Since the shape of most text instances is close to a rectangle, we also add a bounding box branch. Our Bbox branch is parallel to the mask branch, we use cIoU loss [24] as our Bbox loss. CIoU could consider the aspect ratios of bounding boxes and help predictions with low overlap with ground truth find targets to regress. The cIoU loss is defined as:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v \tag{9}$$

where $\mathbf{b}$, $\mathbf{b}^{gt}$ denote the center points of predicted boxes and ground truth boxes respectively. $\rho(\cdot)$ is Euclidean distance, and

TABLE I
PERFORMANCE COMPARISONS WITH DIFFERENT STRUCTURE DESIGNS ON CTW-1500 AND ICDAR2015 DATASET.

Dataset	Method	Precision	Recall	F	FPS
CTW-1500	ResNet50[a] + IoU	81.3	73.1	77.8	13.3
	ResNet101 + IoU	82.5	76.7	79.4	7.7
	Attention + IoU	83.2	78.7	80.8	8.9
	Attention + cIoU	**83.5**	**78.8**	**81**	8.6
ICDAR-15	resnet50[a] + iou	81.3	73.1	77.8	13.3
	resnet101 + iou	82.5	76.7	79.4	7.7
	Attention + iou	83.2	78.7	80.8	8.9
	Attention + ciou	83.5	78.8	81	8.6

c denotes the length of diagonal of the smallest enclosing box which could cover the two boxes. α is a trade-off parameter and v measures the aspect ratio consistency of the two boxes. They are defined respectively as:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{10}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{11}$$

where w and w^{gt} denote the width of predicted boxes and ground truth boxes respectively. h represents the height.

IV. EXPERIMENT

In this section, we show the effectiveness and efficiency of the proposed PolarText for Scene Text detection. We first introduce dataset and measurements. Then, we conduct ablation and parameter setting experiments to show designs of PolarText is highly effective. Afterwards, two groups of comparative studies on several public dataset are conducted to demonstrate PolarText is effective in text detection. Finally, we describe implementation details for readers' convenience.

A. Datasets

Among all datasets for scene text detection, we choose five datasets, including ICDAR15, ICDAR17-MLT, MSRA, Total Text, SCUT CTW-1500. We also used Image-Net and MS-COCO for pre-training. Annotations of ICDAR 15 are labeled as 4 vertices at word level and annotations of CTW1500 and Total text are labeled with boundary points, at text level. Annotations of MSRA are labeled as rectangle boxes and the corresponding angle, we convert it to vertices of quadrilaterals. The evaluation metric used is similar to Pascal Voc, any text instance that has an IoU larger than 0.5 with any ground truth will be considered as positive, and any ground truth could have only one positive example. We use precision, recall and the F-value to evaluate the performance.

B. Parameter Setting Experiment

We conduct experiments to study the effectiveness of different components. Experiments on CTW-1500 and ICDAR-15 are shown in Table I. On CTW1500 dataset, Attention improves the performance by a large margin, it even outperforms heavier backbones such as resnet101 with fewer speed influence. Attention module works especially well on

TABLE II
PERFORMANCE COMPARISON IN CONVERGENCE SPEED ON CTW1500 DATASET.

Method	Epoch50	Epoch100	Epoch200	Epoch300
IoU	34.3	56.1	68.7	78.5
cIoU	34.6	60.2	76.2	80.7

TABLE III
PERFORMANCE COMPARISONS WITH THE EXISTING METHODS ON CTW-1500 AND ICDAR2015 DATASET.

Datasets	Method	Precision	Recall	F	FPS
CTW-1500	CTPN[25]	60.4	53.8	56.9	7.1
	lSegLink[26]	42.3	40.0	40.8	**10.7**
	EAST[27]	78.7	49.1	60.4	-
	CTD [28]	74.3	65.2	69.5	-
	CTD+TLOC[28]	77.4	69.8	73.4	13.3
	DMPNet [29]	69.9	56.0	62.2	-
	TextSnake[6]	67.9	85.3	75.6	8.2
	PSENet[1]	80.6	75.6	78.0	3.9
	LOMO[30]	85.7	69.6	76.8	4.4
	Mask R-CNN [a] [3]	80.8	**83.1**	**81.9**	1.8
	Ours	**83.5**	78.8	81.0	8.6
ICDAR-15	CTPN[25]	74.2	51.6	60.9	7.1
	Zhang et al.[30]	70.8	43.0	53.6	0.5
	PixelLink[26]	82.9	81.7	82.3	7.3
	MSR[31]	86.6	78.4	82.3	-
	EAST[27]	83.6	73.5	78.2	**13.2**
	TextDragon[32]	84.8	81.8	83.1	7.5
	PSENet[1]	81.5	79.7	80.6	1.6
	PAN[33]	77.8	**82.9**	80.3	-
	Mask R-CNN [a] [3]	86.3	81.5	83.8	1.9
	Ours	**88.1**	80.2	**84**	8.7

CTW1500, we believe that it is because we need much more information to get the context of complex shapes. On the other hand, cIoU has little impact on the overall performance but it could help the network train faster, comparisons of the convergence speed between different IoU types are shown in Table II, and we can conclude that cIoU converges much faster and make the network easier to train. After epoch 300, the whole network became over-saturated.

C. Comparative Experiment and Analysis

Comparison of results on different datasets are shown in TableIII, IV respectively. On datasets like ICDAR15, where most text instances are quadrilateral in shape, our proposed method outperforms existing methods. Our method performs particularly well and has a large edge on MSRA dataset where most text instances are rotated. This indicates that our method is rotation invariant. ICDAR17 dataset contains multiple languages, which means its scenes are more complex and diverse than any other dataset. Experiments show that our method works well on this challenging dataset. On curved text dataset like ICDAR15 and CTW1500 our method gained

TABLE IV
PERFORMANCE COMPARISONS WITH THE EXISTING METHODS ON ICDAR2017, MSRA, TOTAL-TEXT DATASETS.

Dataset	Method	Precision	Recall	F	FPS
ICDAR-17	He et al.[34]	76.7	57.9	66.0	-
	Lyu et al.[35]	**83.8**	55.6	66.8	-
	Pixellink[26]	70.9	61.7	65.4	7.3
	Mask R-CNN [a] [3]	74.8	61.1	67.2	2.1
	Ours	75.6	**62.8**	**68.6**	**9.7**
MSRA	SegLink[36]	86.0	70.0	77.0	-
	East [27]	81.7	61.6	70.2	6.5
	TextSnake[6]	83.2	73.9	78.3	1.1
	Zhang et al.[30]	83.0	67.0	74.0	0.48
	He et al. [34]	77.0	70.0	74.0	1.1
	Pixellink[26]	83.0	73.2	77.8	3.0
	Mask R-CNN [a] [3]	84.6	80.5	82.5	1.9
	Ours	**87.0**	**81.2**	**83.9**	9.5
TotalText	SegLink[26]	30.3	23.8	26.7	7.7
	EAST[27]	50.0	36.2	42.0	-
	MSR[31]	83.8	74.8	79.0	4.3
	TextSnake[6]	82.7	74.5	78.4	3.6
	PSENet[1]	81.8	75.1	78.3	3.9
	Mask R-CNN [a] [3]	82.3	**84.5**	**83.3**	1.5
	Ours	**82.4**	76.6	79.3	**7.9**

comparable performance with Mask R-CNN. Our method achieved an FPS of 8.8, which is at least 4 times faster than Mask R-CNN. We think it's because that Mask R-CNN is a two stage method and needs to generate dense predefined anchors and has a lot of computations, so it is slower than our single stage method.

We have also noticed that, Mask R-CNN tends to have higher recall than ours, since Mask R-CNN used a lot of predefined anchors, text instances are less likely to be ignored. On all datasets, our method outperforms Bottom-up methods, like PSENet[1], TextSnake[6]. We think it's due to the difficulty to get a relatively good pixel level segmentation and text instances that are close to each other are not easy to distinguish. The overall experiments show that our method works well in most cases. We show detection samples achieved by the proposed method in Fig. 3 and 4, where we can notice detecting texts in polar representation could greatly improve performance on text detection.

D. Implementation Details

Our network is trained with stochastic gradient descent with the initial learning rate set to 0.01. Warm-up policy is adopted to prevent our method from getting stuck into local minimum. The positive and negative IOU threshold is set to 0.4 and 0.5 respectively. Input images are resized so that the longer side is no longer than 1280. During training, simple data augmentation is used such as random resize, crop and clip. We use Res-Net 50 as backbones and non-local networks as

Fig. 3. Detection results on CTW-1500 dataset.

Fig. 4. Detection results on total text dataset

our attention module. All our experiments are conducted on 4 nvidia GTX 1080 TI gpus.

V. Conclusion

This paper proposes a novel idea to detect text by directly generating contour points of text instances with polar coordinates representation. Our polar representation is specially fit for text detection with the characteristic of rotation invariance. The proposed PolarText not only relieves burden of high computation cost brought by pixel-level classification, but also fits with intrinsic characteristics of text instances to eliminate mislabeled boundary pixels.

Acknowledgment

This work was supported by National Key R&D Program of China under Grant 2018YFC0407901, the Fundamental Research Funds for the Central Universities under Grant B200202177.

References

[1] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," *arXiv preprint arXiv:1806.02559*, 2018.

[2] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask r-cnn with pyramid attention network for scene text detection," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 764–772.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[4] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9038–9045.

[5] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.

[6] 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 (ECCV)*, 2018, pp. 20–36.

[7] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Appl. Soft Comput.*, vol. 103, p. 107108, 2021.

[8] J. Ye, Z. Chen, J. Liu, and B. Du, "Textfusenet: Scene text detection with richer fused features." IJCAI, 2020.

[9] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9038–9045.

[10] Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 753–11 762.

[11] S. Wan, Y. Xia, L. Qi, Y. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Trans. Multim.*, vol. 22, no. 7, pp. 1756–1768, 2020.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[14] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9627–9636.

[15] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, "A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning," *Future Gener. Comput. Syst.*, vol. 100, pp. 316–324, 2019.

[16] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5221–5229.

[17] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9157–9166.

[18] X. Chen, R. Girshick, K. He, and P. Dollár, "Tensormask: A foundation for dense object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2061–2069.

[19] Y. Lee and J. Park, "Centermask: Real-time anchor-free instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 906–13 915.

[20] R. Zhang, Z. Tian, C. Shen, M. You, and Y. Yan, "Mask encoding for single shot instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 226–10 235.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[23] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 193–12 202.

[24] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of The AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[25] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European conference on computer vision*. Springer, 2016, pp. 56–72.

[26] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[27] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[28] 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.

[29] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1962–1969.

[30] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 552–10 561.

[31] C. Xue, S. Lu, and W. Zhang, "Msr: Multi-scale shape regression for scene text detection," *arXiv preprint arXiv:1901.02596*, 2019.

[32] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9076–9085.

[33] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8440–8449.

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

[35] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon *et al.*, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1454–1459.

[36] 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*, 2017, pp. 2550–2558.