Accurate Arbitrary-Shaped Scene Text Detection via Iterative Polynomial Parameter Regression

Jiahao Shi, Long Chen, and Feng Su[0000−0002−8426−9634]∗

State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
suf@nju.edu.cn

Abstract. A number of scene text in natural images have irregular shapes which often cause significant difficulties for a text detector. In this paper, we propose a robust scene text detection method based on a parameterized shape modeling and regression scheme for text with arbitrary shapes. The shape model geometrically depicts a text region with a polynomial centerline and a series of width cues to capture global shape characteristics (e.g. smoothness) and local shapes of the text respectively for accurate text localization, which differs from previous text region modeling schemes based on discrete boundary points or pixels. We further propose a text detection network PolyPRNet equipped with an iterative regression module for text's shape parameters, which effectively enhances the detection accuracy of arbitrary-shaped text. Our method achieves state-of-the-art text detection results on several standard benchmarks.

1 Introduction

Scene text carries useful semantic information for various content-based image applications such as image parsing, classification, and retrieval. Due to the complexity and wide variation of scene text's appearance and various contextual interferences such as complicated background and low contrast, to reliably detect scene text in natural images remains a challenging task.

Traditional scene text detection methods [1–3] usually work in a bottom-up manner which first localizes candidate character regions in the image with manually designed features and some classifiers and then combines them into text. The multi-stage detection pipeline often keeps these methods from achieving overall optimized performance.

More recent scene text detection methods [4–8] often employ deep neural networks such as convolutional neural network (CNN) and recurrent neural network (RNN) to automatically learn effective representations of text and predict text candidates in an end-to-end manner, which significantly enhance the detection performance compared to traditional methods.

Compared to the detection performance on regular scene text which has been continuously improved to a rather high level, there is still large improvement

space in the detection of arbitrarily shaped scene text such as multi-oriented and curved ones due to their largely varied irregular appearances. Accordingly, the focus of increasing researches [5, 6, 8, 7, 9] has turned to it and a number of promising results have been attained. On the other hand, most existing methods employ discrete boundary points or a pixel mask to depict a text region, and few efforts have been made on devising more effective shape models of text to capture its distinctive geometric characteristics beyond a general connected object, which limits potential performance improvements of existing text detection methods.

In this paper, we propose a robust scene text detection method based on a novel parameterized geometric shape modeling and iterative regression scheme for arbitrary-shaped text. The key contributions of our work are summarized as follows:

- We propose a geometric, parameterized shape model for text with arbitrary shapes. The model depicts one text region with a polynomial centerline that captures global shape characteristics such as smoothness of the text as an artificial object and a series of width cues capturing local text shape, which provides effective shape constraints and sufficient flexibility for accurate localization of the text region. The model essentially differs from the pixels- or boundary points-based representations of text region employed by most previous text detection methods.
- Based on the parameterized text shape model, we take the text detection task as a conditional shape parameter regression problem. Accordingly, we propose an end-to-end trainable detection network PolyPRNet that introduces an iterative shape parameter regression module on the basis of backbone networks, which iteratively refines the shape parameters of a potential text candidate for enhanced detection accuracy. We also devise an effective labeling scheme and corresponding loss functions for training the text detection network on the basis of the boundary points based annotations of text provided in most datasets.
- The proposed text detection method is evaluated on several challenging benchmark datasets and achieves state-of-the-art text detection results.

2 Related Work

Generally, existing scene text detection methods can be divided into two main categories: traditional methods employing multi-stage detection pipelines, and recent methods based on end-to-end deep neural networks.

Most traditional scene text detection methods [1, 2, 10, 3, 11] first employed connected component analysis or sliding windows to extract character candidates from the image, and then classified them to either text or non-text using some classifiers and finally grouped characters into text. Due to the bottom-up step-wise detection pipelines employed, however, these methods are usually difficult to be optimized holistically to attain state-of-the-art detection performance.

Recently, with deep learning techniques being extensively employed in diverse computer vision problems, a number of scene text detectors based on various

deep neural network models such as CNN and RNN have emerged, which can be roughly classified into two categories: segmentation-based and regression-based.

Segmentation-based methods localize text regions in an image by inferring the text/non-text label of every pixel using some fully convolutional networks (FCN) [12]. For example, in [13], a multiresolution FCN was proposed for text detection, which classified pixels into three categories — non-text, text border, and text to help separate adjacent text. TextSnake [5] depicted one text instance by a sequence of overlapping disks centered at symmetric axes with variable radiuses and orientations, and then an FCN-based network was used to predict score maps of text center line, text region, and geometry attributes. PSENet [6] depicted text instances with kernels of different scales and, starting from the minimal scale, gradually expanded the kernel to separate and detect adjacent text instances utilizing multi-scale segmentation maps. LOMO [7] iteratively refined detected text proposals to handle long text and introduced a shape expression module to generate more accurate representation of text for detection. CRAFT [8] first localized individual character regions by inferring both character region probability and affinity probability between adjacent characters and then linked the detected characters belonging to one word as final detection results.

Regression-based methods first employ some object detection frameworks such as Faster R-CNN [14] and SSD [15] to generate a set of region proposals, and then predict text candidates by regressing text region parameters based on the proposals. For example, TextBoxes [16] extended SSD with text-box layers, in which the anchor scales and convolution kernel shapes were modified to better adapt to the text detection task. EAST [4] and Direct Regression [17] exploited fully convolutional networks to regress candidate text boxes based on the predicted offsets from each pixel to the box boundaries. In [9], RNN was exploited to predict a pair of boundary points of one potential text region at each time step until the stop label, which allowed the method to handle text regions with arbitrary shapes and adaptive number of boundary points.

Different from most state-of-the-art segmentation-based detection models for arbitrary-shaped text [5, 18, 6, 8, 7, 19], we geometrically depict and regress one text region with a parametric shape model which effectively enhances the detection performance for text with arbitrary shapes.

3 Approach

In this work, we propose a novel polynomial-based parameterized shape modeling and iterative regression scheme for arbitrary-shaped text and, on the basis of it, a robust end-to-end scene text detection network PolyPRNet.

3.1 Polynomial-Based Parameterized Shape Model of Text Region

Existing scene text detection methods usually employ one of two different schemes to model a text region — a quadrangular or polygonal boundary depicted by

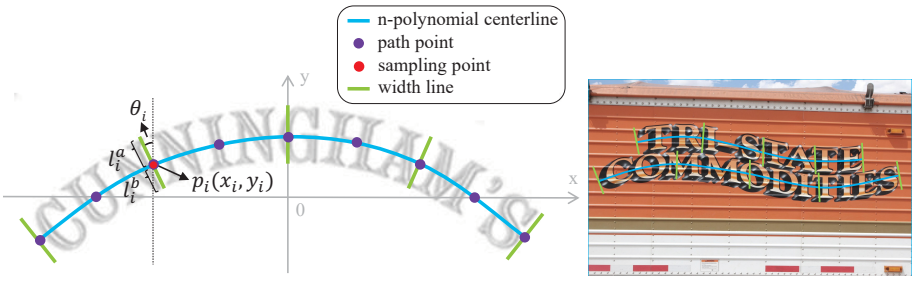

Fig. 1. Illustration of the proposed polynomial-based shape model of text region (left) and an example of text region (right) depicted by the shape model.

discrete vertices, or a set of pixels constituting the text region in a segmentation manner. Both schemes encode only local or general (e.g., connectedness) constraints between vertices or pixels and do not precisely capture distinctive holistic shape characteristics of text as one specific class of man-made objects.

In this work, we propose a parameterized text region modeling scheme that geometrically depicts the shape of one text with a polynomial centerline curve and a series of width cues along the centerline as shown in Fig. 1. Specifically, the n-polynomial centerline of a text region (n denoting the degree of the polynomial), which depicts the global layout and smoothness of the text, is formulated as:

$$y = a_n \times x^n + a_{n-1} \times x^{n-1} + \cdots + a_0 \tag{1}$$

where $a_n, a_{n-1}, \ldots, a_0$ are coefficients of respective polynomial terms, and x and y denote the coordinates of a point on the centerline.

Different from [20] that similarly employs a polynomial text centerline (for generating control points used to rectify text shape), as shown in Fig. 1, we further introduce a series of k *path points* located on the medial axis of the text region as the explicit constraints for the polynomial centerline — it should fit the path points as precisely as possible (as described in Section 3.2 and 3.5), which help attain more accurate prediction of centerline parameters.

Besides the centerline capturing the global shape characteristics of the text, we further depict the width and orientation of each local part of the text along the centerline with a series of m *width lines* as shown in Fig. 1. A width line is depicted by its intersection point p_i with the centerline, which is termed a *sampling point* with coordinates (x_i, y_i), and a pair of parameters l_i^a and l_i^b indicating its length above and below the centerline respectively and a parameter θ_i indicating its angle relative to y-axis. Unlike [20] employing a single width parameter, the two separate width parameters (l_i^a and l_i^b) allows the shape model to depict text with different-sized parts on the two sides of the centerline such as those comprising mixed upper and lower case characters and helps keep the smoothness of centerline. Moreover, the explicit depiction of the sampling points saves the computation of intersection points between the polynomial centerline and width lines to facilitate constructing a differentiable network with easier gradi-

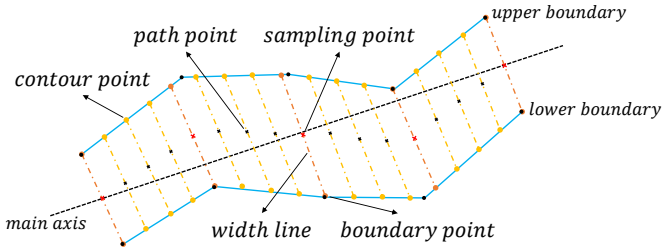

Fig. 2. Illustration of label generation for the text shape representation model.

ent computation. The polynomial parameters $\{a_n, a_{n-1}, \ldots, a_0\}$ of the centerline and the parameters $\{x_i, y_i, l_i^a, l_i^b, \theta_i\}$ of the set of width lines together depict the geometric model of the text region.

Note the above parameterization scheme of text region applies to mainly horizontal text, for vertical ones, we exchange the roles of y and x in the scheme for effective representation of the text region. Accordingly, we employ two separate sets of shape model parameters for horizontal and vertical text respectively, which more accurately capture distinct characteristics of text in two different orientations for enhanced detection accuracy.

3.2 Label Generation

To derive training labels for the parameters of the proposed polynomial-based text shape model from the common polygonal annotations of text regions provided by most text datasets, we propose an effective labeling scheme for arbitrary snake-shaped text instances, i.e., text not forking into multiple branches.

Specifically, as shown in Fig. 2, given the polygon boundary of one text region provided by the dataset, we first divide it into four connected boundary segments: two line segments marking the head and tail positions of the text, and two polylines marking the upper and lower boundaries of the text. Next, we evenly sample a series of k *contour points* on the upper and lower boundaries of the text region respectively, which cover the total length of each boundary with equal spacing. Then, we connect every pair of two corresponding contour points on the upper and lower boundaries with a line segment denoted by $s_{i=1..k}$, and take its midpoint as one path point which is supposed to be located on the centerline of the text region. We further sample a set of m width lines from the line sequence $\{s_1, \cdots, s_k\}$, taking s_1 and s_k as the first and the last width lines respectively. We then take the midpoint (also a path point) of each width line as one sampling point and label its two endpoints as *boundary points*.

Moreover, we assign each text region a direction label represented as a 2-dim one-hot vector $\boldsymbol{d}$, which is set to horizontal ($d_0 = 1$) if the angle between the text's main axis (i.e., the line connecting the first and the last path points) and x-axis is less than 50 degrees, otherwise it is set to vertical ($d_1 = 1$). To accommodate text regions of varied sizes, we normalize the coordinates of all

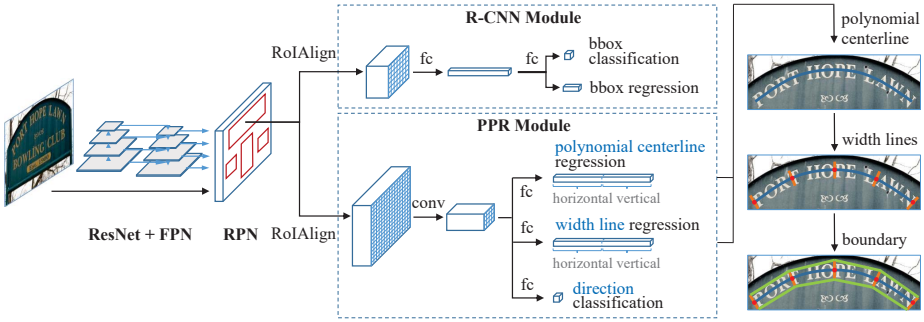

Fig. 3. Illustration of the architecture of the proposed text detection network.

points to the range $[-0.5, 0.5]$ after generating the labels. Note that no information other than standard annotations provided by the dataset is exploited in the label generation process.

3.3 Network Architecture

We propose an end-to-end text detection network PolyPRNet on the basis of the polynomial-based text shape model, which adopts a two-stage R-CNN based framework as illustrated in Fig. 3.

In the first stage, the ResNet50 [21] and a Feature Pyramid Network (FPN) [22] with a four-level feature pyramid are employed to extract multi-level feature maps from the input image, which are then used as the shared input to subsequent network modules. Next, we employ the RPN network [14] to generate a set of text region proposals, and an RoIAlign operation [23], which evenly splits input RoI feature maps into 16×16 blocks, is applied on each proposal to generate feature maps of fixed size which preserve the proposal's exact spatial location information.

In the second stage, we employ an R-CNN module with a bounding box regression branch and a classification branch to refine the bounding box of a text region proposal generated by RPN with more accurate location information and assign it a text/non-text score. Specifically, in this work, we employ a Cascade R-CNN [24] as the R-CNN module, which comprises three stages with IoU thresholds $\{0.5, 0.55, 0.6\}$ and loss weights $\{1, 0.5, 0.25\}$ for each stage respectively.

Given the text region proposals generated by RPN, we introduce a *polynomial-based shape parameter regression (PPR)* module to infer the shape parameters and direction of a potential text candidate based on the proposed parameterized text shape model. Specifically, the feature maps of one text region proposal first undergo a 3×3 convolutional layer followed by two groups of 3×3 convolutional layers and 2×2 max-pooling layers. Finally, two full-connected layers are employed to predict the shape parameters of the candidate text region.

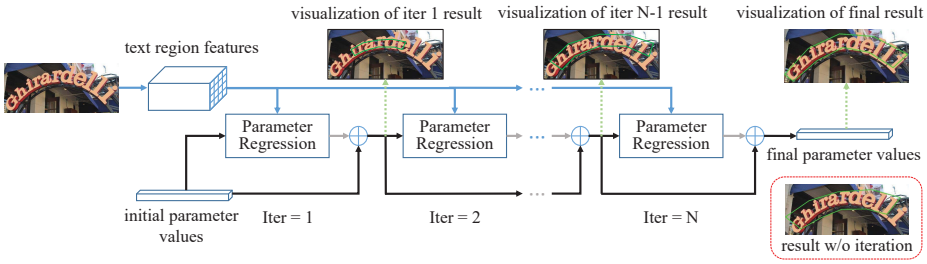

Fig. 4. Illustration of the iterative shape parameter regression pipeline. The 'Parameter Regression' block is composed of the last two full-connected layers of the PPR module for text shape parameter prediction. $\oplus$ denotes the addition operation. The black input to $\oplus$ is the shape parameter values obtained in the previous iteration, and the grey input is the predicted refinements to the parameter values. The output of $\oplus$ is the updated parameter values of the current iteration.

3.4 Iterative Shape Parameter Regression

To help attain optimal regression of shape parameters of a text region, we employ an iterative parameter regression pipeline as shown in Fig. 4. Specifically, with the values of shape parameters being initialized to zero, in each iteration, the parameter regression block takes the concatenation of the flattened feature maps of a text region and the vector of current shape parameter values as input, and predicts a refinement to be added to each current parameter value to generate its updated value for the next iteration. This iterative shape parameter regression process repeats until a predefined number (3 in this work) of refinement iterations is reached, which yields the final shape parameter values of the text candidate. Comparing the detection results shown in Fig. 4 with and without the iterative refinement process, the iteration mechanism effectively improves the accuracy of shape parameter regression.

3.5 Loss Functions

We define a multitask loss on each text region proposal as the sum of the loss L_{rpn} for the RPN subnetwork [14], the loss L_{rcnn} for the Cascade R-CNN subnetwork [24], and the loss L_{ppr} for the PPR module in the proposed PolyPRNet:

$$L = \lambda_1 L_{rpn} + \lambda_2 L_{rcnn} + \lambda_3 L_{ppr} \tag{2}$$

where the weights λ_1, λ_2, and λ_3 are set to 1.0 in this work.

Specifically, the loss L_{ppr} is composed of the text region approximation loss L_{reg} and the text direction classification loss L_{dir}:

$$\begin{aligned} L_{ppr} = {} & \lambda_4 I_{\mathbf{d}_0^*=1} L_{reg}(\mathbf{a}_x, \mathbf{c}_x, \mathbf{\Theta}_x, \mathbf{l}_x, \mathbf{P}^*, \mathbf{c}_x^*, \mathbf{\Theta}_x^*, \mathbf{l}_x^*, \mathbf{T}^*) \\ & + \lambda_4 I_{\mathbf{d}_1^*=1} L_{reg}(\mathbf{a}_y, \mathbf{c}_y, \mathbf{\Theta}_y, \mathbf{l}_y, \mathbf{P}^*, \mathbf{c}_y^*, \mathbf{\Theta}_y^*, \mathbf{l}_y^*, \mathbf{T}^*) \\ & + \lambda_5 L_{dir}(\mathbf{d}, \mathbf{d}^*) \end{aligned} \tag{3}$$

where, $\mathbf{a}$ denotes the vector of the predicted coefficients of the polynomial centerline function defined by Eq. (1). $\mathbf{c}$ denotes the vector of predicted x/y coordinates of the sampling points of a horizontal/vertical text, and $\mathbf{\Theta}$ and $\mathbf{l}$ denote the vectors of predicted angles and length of the width lines respectively. $\mathbf{c}^*$, $\mathbf{\Theta}^*$, and $\mathbf{l}^*$ are corresponding ground-truth. The subscript x and y indicate the associated terms applying to horizontal and vertical text regions respectively. $\mathbf{P}^*$ denotes the vector of ground-truth path points that the predicted polynomial centerline is supposed to pass through. $\mathbf{T}^*$ denotes the vector of ground-truth boundary points of the text region. $L_{dir}(\mathbf{d}, \mathbf{d}^*)$ is the binary cross-entropy loss between the predicted text direction probability vector $\mathbf{d}$ and the ground-truth one-hot direction vector $\mathbf{d}^*$. I denotes the indicator function for the text direction. The balancing weights λ_4 and λ_5 are set to 5.0 and 0.5 respectively.

The text region approximation loss L_{reg} measures the approximation accuracy of the predicted text region relative to the ground-truth annotation, which is formulated as a combination of the approximation loss L_{reg}^{line} on the polynomial centerline and the approximation loss L_{reg}^{width} on the width lines:

$$L_{reg}(\mathbf{a}, \mathbf{c}, \mathbf{\Theta}, \mathbf{l}, \mathbf{P}^*, \mathbf{c}^*, \mathbf{\Theta}^*, \mathbf{l}^*, \mathbf{T}^*) = L_{reg}^{line}(\mathbf{a}, \mathbf{P}^*) + L_{reg}^{width}(\mathbf{a}, \mathbf{c}, \mathbf{\Theta}, \mathbf{l}, \mathbf{c}^*, \mathbf{\Theta}^*, \mathbf{l}^*, \mathbf{T}^*) \tag{4}$$

The centerline approximation loss L_{reg}^{line} measures the fitting accuracy of the predicted polynomial centerline (parameterized by $\mathbf{a}$) against the ground-truth path points $\mathbf{P}^*$, which is formulated as:

$$L_{reg}^{line}(\mathbf{a}, \mathbf{P}^*) = smooth_{L1}(sum(|f(\mathbf{a}, \mathbf{P}^*)|)) \tag{5}$$

$$f(\mathbf{a}, \mathbf{P}^*) = \begin{bmatrix} a_n & a_{n-1} & .. & a_0 & -1 \end{bmatrix} \begin{bmatrix} u_0^n & u_1^n & .. & u_k^n \\ u_0^{n-1} & u_1^{n-1} & .. & u_k^{n-1} \\ .. & .. & .. & .. \\ u_0^0 & u_1^0 & .. & u_k^0 \\ v_0 & v_1 & .. & v_k \end{bmatrix} \tag{6}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{7}$$

where u_i and v_i correspond to the coordinates x_i and y_i of the ith path point in $\mathbf{P}^*$ respectively if $\mathbf{d}_0^* \geq \mathbf{d}_1^*$, otherwise they correspond to y_i and x_i.

The width line approximation loss L_{reg}^{width} is formulated as:

$$\begin{aligned} L_{reg}^{width}(\mathbf{a}, \mathbf{c}, \mathbf{\Theta}, \mathbf{l}, \mathbf{c}^*, \mathbf{\Theta}^*, \mathbf{l}^*, \mathbf{T}^*) = smooth_{L1}(sum(|\mathbf{c} - \mathbf{c}^*|)) \\ + smooth_{L1}(sum(|\mathbf{\Theta} - \mathbf{\Theta}^*|)) + smooth_{L1}(sum(|\mathbf{l} - \mathbf{l}^*|)) \\ + smooth_{L1}(sum(|\mathbf{T} - \mathbf{T}^*|)) \end{aligned} \tag{8}$$

where $\mathbf{T}$ denotes the vector of predicted boundary points, whose coordinates are computed based on the predicted width line parameters $\mathbf{\Theta}$ and $\mathbf{l}$ and the predicted sampling point coordinates computed based on $\mathbf{a}$ and $\mathbf{c}$.

3.6 Inference

Given an input text image, each text region proposal generated by the RPN network is first fed to the R-CNN module to obtain its accurate bounding box and classification score. Then, proposals whose scores fall below 0.7 (0.65 in multi-scale testing — see Section 4.4) are discarded, and non-maximum suppression [25] with IoU threshold 0.4 is applied on the proposals to obtain a set of at most 200 detection boxes with highest scores, which are fed to the PPR module for text region parameter prediction.

With the shape parameters predicted by the PPR module, we first determine the direction of the text to be the one corresponding to the greater probability in the predicted 2-dim direction vector. We then calculate the y coordinate of each sampling point on the polynomial centerline of a horizontal text region according to Eq. (1) given its predicted x coordinate — for a vertical text region, the roles of y and x are exchanged. Finally, given the predicted parameters θ_i, l_i^a, and l_i^b of a width line crossing the sampling point (x_i, y_i), we calculate the coordinates of its two endpoints and further obtain the polygonal boundary of the text region by sequentially connecting the endpoints of all width lines. Note that, for datasets adopting quadrangular annotations of text, we calculate the minimal area bounding rectangle of the predicted polygonal boundary of the text as the final detection result.

4 Experiments

4.1 Dataset

We evaluate our scene text detection method on four challenging benchmark datasets: TotalText, CTW1500, ICDAR2015, and ICDAR2017-MLT.

TotalText dataset [26] consists of 1255 and 300 images for training and testing respectively, which contain multi-oriented and curved text instances, each with a polygonal annotation comprising 10 vertices. **CTW1500** dataset [27] comprises 1000 training images and 500 testing images with a large number of challenging long curved text. Each text is annotated by a polygon with 14 vertices. **ICDAR2015** dataset [28] is composed of 1000 training images and 500 testing images, which contain accidental scene text instances with quadrangular annotations. **ICDAR2017-MLT** dataset [29] consists of 7200, 1800, and 9000 images for training, validation, and testing respectively, which contain multi-oriented, multi-scripting, and multi-lingual scene text instances with quadrangular annotations.

We adopt the standard evaluation protocol for text detection, which measures the detection performance by precision P, recall R, and f-measure F (i.e., the harmonic mean $\frac{2*P*R}{P+R}$ of P and R).

4.2 Implementation Details

We implement the proposed PolyPRNet on the basis of the PyTorch framework and conduct the experiments on one NVIDIA Tesla V100 GPU.

With the ResNet [21] backbone pre-trained on ImageNet dataset [30], we train the whole detection network end-to-end using stochastic gradient descent (SGD) with 0.0001 weight decay and 0.9 momentum and mini-batch of 10.

The training process comprises two stages: pre-training on a combined dataset and fine-tuning on each dataset. The pre-training dataset is the same as that employed in [6], which is composed of 10K images from ICDAR2015 dataset's training set and the training and validation sets of ICDAR2017-MLT dataset. We train the detection model on the pre-training dataset for 60K iterations with the learning rate starting from 0.01 and reduced to 0.001 for the last 20K iterations. In the fine-tuning stage, we train separate detection models for different test datasets using their own training sets on the basis of the pre-trained model. For curved text datasets TotalText and CTW1500, the learning rate is initialized to 0.01 for first 40K training iterations and is reduced to 0.001 for further 20K iterations. For ICDAR2015 and ICDAR2017-MLT datasets, the learning rate is set to 0.001 during 40K training iterations of the model.

We use a polynomial centerline of degree 3 and 5 width lines as default for depicting a text region, and the number of path points is set to 13 in the experiments.

4.3 Ablation Study

Effectiveness of Polynomial-Based Text Shape Modeling and Regression. We validate the effectiveness of the proposed polynomial-based text shape modeling and regression mechanism for scene text detection by comparing the performance of some variants of PolyPRNet in Table 1. Specifically, the model 'Cas. R-CNN' removes the PPR module from PolyPRNet and uses the Cascade R-CNN for text detection, which employs rectangular bounding boxes to depict text regions. The model 'Cas. R-CNN + QuadPR' employs a quadrangle to depict the text region and replaces the PPR module in PolyPRNet with a regression branch to predict the parameters of quadrangles. The model 'Cas. R-CNN + Mask' replaces the PPR module with the mask branch proposed in Mask R-CNN [23] for text candidate prediction. The model 'Cas. R-CNN + PPR' denotes the proposed PolyPRNet with the iterative shape parameter regression mechanism being removed for fair comparison with other variants in which this mechanism cannot be similarly employed.

As shown in Table 1, compared to the Cascade R-CNN backbone, introducing the PPR module significantly enhances the detection f-measure by $9 - 19\%$, which clearly demonstrates the effectiveness of the proposed text shape modeling and regression mechanism. Moreover, the proposed PPR module is more effective than the mask mechanism [23] as it effectively captures and exploits distinctive shape characteristics of text rather than low-level segmentation information. Figure 5 shows some examples of detection results by variant models in Table 1. The PPR-based detection model yields more accurate text region boundaries than other models.

To further evaluate the effectiveness of the PPR module, we combine it with the more general Faster R-CNN backbone. Compared to the detection f-measure

Table 1. Effectiveness of the proposed PPR module for enhancing text detection performance (%)

Model	TotalText			CTW1500			ICDAR2015		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Cas. R-CNN	64.6	65.0	64.8	69.0	67.1	68.0	78.1	78.7	78.4
Cas. R-CNN + QuadPR	73.3	72.7	73.0	69.2	68.4	68.8	79.5	74.8	77.1
Cas. R-CNN + Mask	**86.5**	82.0	84.1	82.2	81.0	81.6	**89.4**	81.0	85.0
Cas. R-CNN + PPR	84.5	**84.7**	**84.6**	**84.2**	**82.7**	**83.4**	89.1	**86.0**	**87.5**

62.2%, 65.3%, and 76.7% attained by Faster R-CNN on TotalText, CTW1500, and ICDAR2015 datasets respectively, introducing the PPR module achieves again the significantly enhanced f-measure 83.5%, 82.7%, and 87.1% which reveal its effectiveness.

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

Fig. 5. Examples of detection results by variant text detection models in Table 1: (a) Cas. R-CNN, (b) Cas. R-CNN + QuadPR, (c) Cas. R-CNN + Mask, and (d) Cas. R-CNN + PPR.

Influence of Degree of Polynomial. We investigate the influence of the degree of the polynomial text centerline on the detection performance in Table 2. It can be seen that, for curved text in TotalText dataset, the degree of the polynomial needs to be large enough to accommodate complex shapes of text, and the detection performance increases with the polynomial degree and reaches the peak around a degree of 5. On the other hand, for mostly straight text in ICDAR2015 dataset, as expected, different polynomial degrees do not change the detection performance much.

Influence of Number of Width Lines. We further inspect the influence of the number of width lines in the proposed text region shape model on the detection performance in Table 3. It can be seen that using more width lines to depict the text region usually yields better detection results due to the more accurate text boundary. For text instances in the experiment datasets, 5 to 9 width lines are generally sufficient for the detection task.

Table 2. Comparison of text detection performance (%) using different degrees (n) of polynomial for text region modeling

n	TotalText			ICDAR2015		
	P	R	F	P	R	F
1	82.0	82.2	82.1	89.2	85.8	87.5
2	84.1	83.5	83.8	88.4	86.0	87.1
3	84.5	84.7	84.6	89.1	86.0	87.5
4	84.6	84.7	84.7	**89.7**	85.8	**87.7**
5	84.8	**84.9**	**84.9**	88.6	**86.9**	**87.7**
6	**84.9**	84.7	84.8	89.4	85.7	87.5

Table 3. Comparison of text detection performance (%) using different numbers (m) of width lines for text region modeling

m	TotalText			ICDAR2015		
	P	R	F	P	R	F
3	83.9	83.4	83.6	88.3	86.0	87.1
5	84.5	**84.7**	84.6	89.1	86.0	87.5
7	85.1	84.1	84.6	89.1	**86.1**	87.6
9	**85.6**	84.5	**85.0**	**89.9**	86.0	**87.9**
11	83.9	83.6	83.8	89.4	85.7	87.5

Note that increasing the number (m) of width lines and the polynomial degree (n) has limited impact on the network's efficiency as the network size is enlarged little. For example, increasing n from 1 to 6 causes only a 0.4 drop in FPS on TotalText dataset, and increasing m from 3 to 11 results in a 0.2 drop in FPS.

Effectiveness of Iterative Shape Parameter Regression (IPR). We verify the effectiveness of the iterative shape parameter regression mechanism by comparing the detection performance with and without it in Table 4. It can be seen that, owing to the more accurate text boundaries predicted, iterative shape parameter regression effectively improves the detection results without introducing much computational overhead (e.g. 0.2 drop in FPS on TotalText dataset compared to no IPR).

Table 4. Comparison of text detection performance (%) with and without IPR

Model	TotalText			CTW1500			ICDAR2015		
	P	R	F	P	R	F	P	R	F
w/o IPR	84.5	84.7	84.6	84.2	82.7	83.4	**89.1**	86.0	87.5
w. IPR	**86.3**	**85.0**	**85.6**	**84.3**	**83.4**	**83.8**	89.0	**86.6**	**87.8**

4.4 Comparison with State-of-the-Art Text Detection Methods

Curved Text Detection. To demonstrate the effectiveness of PolyPRNet for detecting curved text, in Table 6 and 5, we compare both the single-scale and the multi-scale (MS) performance of PolyPRNet with other state-of-the-art text detection methods on CTW1500 and TotalText datasets respectively, using the same evaluation scheme as [26]. In the single-scale testing, the shorter sides of a test image are scaled to 720, while in the multi-scale testing, they are scaled to $\{640, 720, 800\}$ respectively. For CTW1500 dataset, the number of width lines in the text shape model is set to 7 to correspond to the 14-vertices annotations in the dataset, and it is set to 5 for TotalText and other datasets.

Table 5. Comparison of text detection performance (%) on TotalText dataset

Method	P	R	F	FPS
SegLink[31]	30.3	23.8	26.7	-
DeconvNet[26]	33.0	40.0	36.0	-
EAST[4]	50.0	36.2	42.0	-
TextSnake[5]	82.7	74.5	78.4	-
Wang *et al.*[9]	80.9	76.2	78.5	-
TextDragon [32]	84.5	74.2	79.0	-
PSENet-1s[6]	84.0	78.0	80.9	3.9
SPCNet[18]	83.0	82.8	82.9	-
CRAFT [8]	87.6	79.9	83.6	-
PAN [19]	**89.3**	81.0	85.0	39.6
PolyPRNet	86.3	**85.0**	**85.6**	13.5
LOMO MS[7]	87.6	79.3	83.3	-
PolyPRNet MS	**88.1**	**85.3**	**86.7**	3.7

Table 6. Comparison of text detection performance (%) on CTW1500 dataset

Method	P	R	F	FPS
SegLink [31]	42.3	40.0	40.8	10.7
EAST [4]	78.7	49.1	60.4	21.2
CTD+TLOC [27]	77.4	69.8	73.4	13.3
TextSnake [5]	67.9	**85.3**	75.6	1.1
Wang *et al.* [9]	80.1	80.2	80.1	-
TextDragon [32]	79.5	81.0	80.2	-
PSENet-1s [6]	84.8	79.7	82.2	3.9
CRAFT [8]	86.0	81.1	83.5	-
PAN [19]	**86.4**	81.2	83.7	39.8
PolyPRNet	84.3	83.7	**84.0**	14.1
LOMO MS [7]	**85.7**	76.5	80.8	-
PolyPRNet MS	85.4	**83.9**	**84.7**	3.8

On both TotalText and CTW1500 datasets, PolyPRNet surpasses all other methods on f-measure in both single-scale and multi-scale testings, which reveal the superiority of our method in detecting various curved scene text with the polynomial-based text shape model and iterative regression mechanism.

Multi-Oriented Text Detection. We validate the effectiveness of PolyPRNet for detecting multi-oriented text in ICDAR2015 dataset. The shorter sides of a test image are scaled to 1320 in the single-scale testing and $\{720, 1320, 1920\}$ respectively in the multi-scale (MS) testing. As shown in Table 7, PolyPRNet achieves the highest f-measure in both single-scale and multi-scale testings in the comparison, demonstrating PolyPRNet's well capability to localize text with arbitrary orientations.

Multilingual Text Detection. We also evaluate PolyPRNet's performance on multilingual text in ICDAR2017-MLT dataset. The test images are scaled in the same way as for ICDAR2015 dataset in single-scale and multi-scale testings. As shown in Table 8, PolyPRNet yields the highest f-measure in the comparison, showing that the proposed polynomial-based text shape model effectively captures the largely varied shape characteristics of text in different languages.

4.5 Qualitative Results

Figure 6 shows some text detection results of our method. Notice the curved text in the images, regardless of their variant styles and tight spacing, are accurately localized. The results demonstrate our method's capability of robustly detecting text with varied shapes, orientations, sizes, and languages.

Table 7. Comparison of text detection performance (%) on ICDAR2015 dataset

Method	*P*	*R*	*F*	*FPS*
MCN [33]	72.0	80.0	76.0	-
Lyu *et al.* [34]	**94.1**	70.7	80.7	3.6
TextSnake [5]	84.9	80.4	82.6	1.1
PAN [19]	84.0	81.9	82.9	26.1
TextDragon [32]	84.8	81.8	83.1	-
PixelLink [35]	85.5	82.0	83.7	7.3
PSENet-1s [6]	86.9	84.5	85.7	1.6
IncepText [36]	89.4	84.3	86.8	-
SPCNet [18]	88.7	85.8	87.2	-
Wang *et al.* [9]	89.2	86.0	87.6	-
PolyPRNet	89.0	**86.6**	**87.8**	7.4
RRD MS [37]	88.0	80.0	83.8	-
Lyu *et al.* MS [34]	89.5	79.7	84.3	-
LOMO MS [7]	87.6	**87.8**	87.7	-
PolyPRNet MS	**91.5**	86.1	**88.7**	1.5

Table 8. Comparison of text detection performance (%) on ICDAR2017-MLT dataset

Method	*P*	*R*	*F*	*FPS*
E2E-MLT [38]	64.6	53.8	58.7	-
He *et al.* [17]	76.7	57.9	66.0	-
Lyu *et al.* [34]	**83.8**	56.6	66.8	-
AF-RPN [39]	75.0	66.0	70.0	-
SPCNet [18]	73.4	66.9	70.0	-
PSENet [6]	73.8	**68.2**	70.9	-
PolyPRNet	81.2	66.8	**73.3**	7.4
Lyu *et al.* MS [34]	74.3	**70.6**	72.4	-
LOMO MS [7]	80.2	67.2	73.1	-
SPCNet MS [18]	80.6	68.6	74.1	-
PolyPRNet MS	**82.9**	69.3	**75.5**	1.5

Fig. 6. Examples of text detection results of our method.

5 Conclusions

We present a robust scene text detection method with a polynomial-based parameterized shape modeling and regression scheme for arbitrary-shaped text, which effectively captures the shape characteristics of text and significantly enhances the performance of the R-CNN based detection backbone. The state-of-the-art text detection results our method obtains on standard benchmarks demonstrate its effectiveness.

Acknowledgments. The research was supported by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20171345 and the National Natural Science Foundation of China under Grant Nos. 61003113, 61321491, and 61672273.

References

1. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR. (2010) 2963–2970
2. Yin, X., Yin, X., Huang, K., Hao, H.: Robust text detection in natural scene images. IEEE Trans. Pattern Anal. Mach. Intell. **36** (2014) 970–983
3. Tian, S., Pan, Y., Huang, C., Lu, S., Yu, K., Tan, C.L.: Text Flow: A unified text detection system in natural scene images. In: ICCV. (2015) 4651–4659
4. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: An efficient and accurate scene text detector. In: CVPR. (2017) 2642–2651
5. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: TextSnake: A flexible representation for detecting text of arbitrary shapes. In Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., eds.: Computer Vision – ECCV 2018, Springer International Publishing (2018) 19–35
6. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: CVPR. (2019) 9336–9345
7. Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., Ding, X.: Look more than once: An accurate detector for text of arbitrary shapes. In: CVPR. (2019) 10544–10553
8. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: CVPR. (2019) 9365–9374
9. Wang, X., Jiang, Y., Luo, Z., Liu, C., Choi, H., Kim, S.: Arbitrary shape scene text detection with adaptive text region representation. In: CVPR. (2019) 6449–6458
10. Sun, L., Huo, Q., Jia, W., Chen, K.: A robust approach for text detection from natural scene images. Pattern Recognition **48** (2015) 2906–2920
11. Zhang, Z., Shen, W., Yao, C., Bai, X.: Symmetry-based text line detection in natural scenes. In: CVPR. (2015) 2558–2567
12. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015) 3431–3440
13. Wu, Y., Natarajan, P.: Self-organized text detection with minimal post-processing via border learning. In: ICCV. (2017) 5010–5019
14. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. (2015) 91–99
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: Computer Vision – ECCV 2016, Springer International Publishing (2016) 21–37
16. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: A fast text detector with a single deep neural network. In: AAAI. (2017) 4161–4167
17. He, W., Zhang, X., Yin, F., Liu, C.: Multi-oriented and multi-lingual scene text detection with direct regression. IEEE Trans. Image Processing **27** (2018) 5406–5419
18. Xie, E., Zang, Y., Shao, S., Yu, G., Yao, C., Li, G.: Scene text detection with supervised pyramid context network. In: AAAI. (2019) 9038–9045
19. Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., Shen, C.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: ICCV. (2019) 8439–8448
20. Zhan, F., Lu, S.: ESIR: End-to-end scene text recognition via iterative image rectification. In: CVPR. (2019) 2054–2063
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 770–778

22. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. (2017) 936–944
23. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. (2017) 2980–2988
24. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into high quality object detection. In: CVPR. (2018) 6154–6162
25. Girshick, R.B., Iandola, F.N., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: CVPR. (2015) 437–446
26. Chng, C.K., Chan, C.S.: Total-Text: A comprehensive dataset for scene text detection and recognition. In: ICDAR. (2017) 935–942
27. Liu, Y., Jin, L., Zhang, S., Zhang, S.: Detecting curve text in the wild: New dataset and new solution. CoRR **abs/1712.02170** (2017)
28. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S.K., Bagdanov, A.D., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., Shafait, F., Uchida, S., Valveny, E.: ICDAR 2015 competition on robust reading. In: ICDAR. (2015) 1156–1160
29. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., Khlif, W., Luqman, M.M., Burie, J., Liu, C., Ogier, J.: ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In: ICDAR. (2017) 1454–1459
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NeurIPS. (2012) 1106–1114
31. Shi, B., Bai, X., Belongie, S.J.: Detecting oriented text in natural images by linking segments. In: CVPR. (2017) 3482–3490
32. Feng, W., He, W., Yin, F., Zhang, X., Liu, C.: TextDragon: An end-to-end framework for arbitrary shaped text spotting. In: ICCV. (2019) 9075–9084
33. Liu, Z., Lin, G., Yang, S., Feng, J., Lin, W., Goh, W.L.: Learning markov clustering networks for scene text detection. In: CVPR. (2018) 6936–6944
34. Lyu, P., Yao, C., Wu, W., Yan, S., Bai, X.: Multi-oriented scene text detection via corner localization and region segmentation. In: CPVR. (2018) 7553–7563
35. Deng, D., Liu, H., Li, X., Cai, D.: PixelLink: Detecting scene text via instance segmentation. In: AAAI. (2018) 6773–6780
36. Yang, Q., Cheng, M., Zhou, W., Chen, Y., Qiu, M., Lin, W.: IncepText: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection. In: IJCAI. (2018) 1071–1077
37. Liao, M., Zhu, Z., Shi, B., Xia, G.s., Bai, X.: Rotation-sensitive regression for oriented scene text detection. In: CVPR. (2018) 5909–5918
38. Busta, M., Patel, Y., Matas, J.: E2E-MLT - an unconstrained end-to-end method for multi-language scene text. In Carneiro, G., You, S., eds.: Computer Vision – ACCV 2018 Workshops, Springer International Publishing (2019) 127–143
39. Zhong, Z., Sun, L., Huo, Q.: An anchor-free region proposal network for Faster R-CNN-based text detection approaches. IJDAR **22** (2019) 315–327