Reading Arbitrary-Shaped Scene Text from Images Through Spline Regression and Rectification

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

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

Abstract. Scene text in natural images contains a wealth of valuable semantic information. To read scene text from the image, various text spotting techniques that jointly detect and recognize scene text have been proposed in recent years. In this paper, we present a novel end-to-end text spotting network SPRNet for arbitrary-shaped scene text. We propose a parametric B-spline centerline-based representation model to describe the distinctive global shape characteristics of the text, which helps to effectively deal with interferences such as local connection and tight spacing of text and other object, and a text is detected by regressing its shape parameters. Further, exploiting the text's shape cues, we employ adaptive projection transformations to rectify the feature representation of an irregular text, which improves the accuracy of the subsequent text recognition network. Our method achieves competitive text spotting performance on standard benchmarks through a simple architecture equipped with the proposed text representation and rectification mechanism, which demonstrates the effectiveness of the method in detecting and recognizing scene text with arbitrary shapes.

Keywords: Scene text spotting · Spline · Regression · Rectification

1 Introduction

Scene text in natural images carries a wealth of semantic information, which is of great importance in various real-world applications. To read the scene text from the image, text spotting methods first localize text regions in the image and then recognize the character sequences contained in them. Due to the complex and varied appearance of text, scene text spotting has been a challenging task and attracted increasing research attention in recent years.

Most of recent scene text spotting methods [46, 5, 31, 18, 23, 42] integrated text detection and recognition into an end-to-end framework to exploit the complementarity of these two tasks to effectively improve the performance of the whole spotting model. Meanwhile, to alleviate the difficulties that the irregular shape of a scene text causes to a text recognition network, variant techniques like

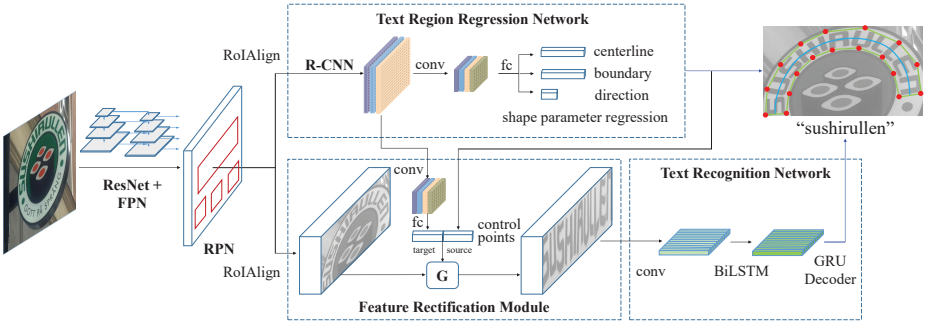

Fig. 1. Illustration of the architecture of the proposed text spotting network SPRNet. The network first detects text with arbitrary shapes in the image by a spline-based text region representation and regression model. An adaptive spatial rectification module is then employed to transform the text's feature representation to a regular shape to facilitate the subsequent text recognition. 'G' denotes the grid sampling operation for feature deformation based on predicted control points.

shape rectification [23] and spatial attention [31] have been employed in recent text spotting methods for generating appropriate features for text recognition.

Despite the great progress in enhancing scene text spotting performance, most of existing text spotting methods employed a text region localization mechanism based on either segmentation [29, 46, 31] or regression of discrete boundary points [39], which did not capture the text's shape characteristics as a whole (e.g., via a global shape model) and sometimes required some post-processing like grouping or fitting to obtain the final text region.

In this paper, we propose a novel end-to-end scene text spotting network SPRNet, which integrates a spline-based parametric shape regression network for localizing arbitrary-shaped text region, an adaptive text feature rectification module, and a light-weight text recognition network. Figure 1 shows the overall architecture of the proposed text spotting network. The key contributions of our work are summarized as follows:

– We propose a spline-based representation and regression model for detecting arbitrary-shaped text. The model geometrically describes the global shape of a text and its intrinsic smoothness and regularity with a parametric B-spline centerline and associated boundary cues. Compared with the segmentation- or boundary-based text representations employed in previous text spotting methods, our parametric, centerline-based representation of text is less susceptible to interferences such as local connection and tight spacing of text and other object due to its modeling and constraints on the overall shape and regularity of the text. Moreover, the model obtains directly the complete boundary of the text as the localization result, eliminating the need for post-processing that segmentation-based methods usually rely on to obtain the final text boundary.

- We integrate a shape rectification mechanism with the text spotting model for recognizing text with arbitrary shapes. The rectification module exploits adaptive projection transformations and the shape cues of an irregular text obtained by the detection module to regularize the text's feature representation, which effectively improves the accuracy of the subsequent text recognition network.
- Our text spotting method achieves competitive performance on several scene text benchmarks.

2 Related Work

Scene Text Detection. Most of recent scene text detection methods can be generally categorized into two schemes: segmentation-based and regression-based. Segmentation-based methods [45, 27, 41, 49] localize text regions by predicting a text/non-text label for every image pixel using some fully convolutional networks [26]. Accordingly, a text region is usually modeled as a connected set of text pixels in these methods, and some of them [27, 5, 49] further model a text's centerline region as a shrunk mask of the whole text area consisting of a set of points on the text's central axis associated with local geometric attributes such as centerline/character orientations and boundary offsets, and certain post-processing is often required to generate the final boundary of the text. Regression-based methods [20, 50, 44] predict text candidates by regressing their bounding box parameters based on generated proposals or from dense features directly, while a text region is usually depicted by its polygonal boundary with discrete vertices.

Note both pixel-based and boundary-based text representations employed in most previous work capture only local constraints such as connectedness or offset between individual pixels or boundary points, lacking accurate description of a text's global shape characteristics. Comparatively, our method geometrically and holistically depicts the text shape with a parametric representation based on B-spline.

Scene Text Recognition. Recent text recognition methods usually employ some sequence models like RNN to recognize the character sequence in an image region as a whole, avoiding error-prone segmentation of individual characters. Particularly, the encoder-decoder framework has often been employed in text recognition, with the encoder encoding the text region into a feature sequence and the decoder predicting a sequence of most probable character labels corresponding to the features with connectionist temporal classification (CTC) [6, 33] or attention mechanisms [16, 3]. To cope with text in irregular shapes, some recent methods further proposed rectification [34, 28, 35, 48, 47] and 2D attention [17] techniques for obtaining appropriate text features for recognition. For example, in [47], a text's shape was characterized by a point-based centerline associated with local geometric attributes similar to [27], which was used to gen-

erate the fiducial points of a TPS transformation for rectifying the feature maps of an irregular text.

Scene Text Spotting. Earlier scene text spotting methods [40, 12, 19] often employed a two-stage pipeline that performed text detection and recognition in separate steps. Due to the complementarity of text detection and recognition tasks, however, it is difficult for these two-stage spotting methods to attain holistically optimal performance.

Most of recent scene text spotting methods [46, 5, 31, 18, 23, 42] employed an end-to-end detection and recognition pipeline for improved spotting performance. Particularly, to handle arbitrary-shaped scene text, some methods introduced spatial rectification measures [5, 23] to help obtain regularized representations of the text or spatial attention mechanisms [31, 18] to adaptively align features with characters for recognition. For example, in [31], a Mask R-CNN based instance segmentation model was combined with a seq2seq recognition model and a spatial attention mechanism for text spotting. On the other hand, ABCNet [23] first localized the text boundary depicted by two Bezier curves, and then exploited the BezierAlign operation to generate rectified features of the text for recognition. Our method differs from previous work in two main aspects — the text region representation and regression model and the text feature rectification mechanism, which are described in detail in following respective sections.

3 Methodology

We propose an effective scene text spotting network SPRNet. As shown in Fig. 1, the network localizes arbitrary-shaped text regions in the image with a spline-based text shape representation and regression model, and then adaptively rectifies the feature representation of an irregular text for subsequent text recognition.

3.1 Text Localization via Spline-Based Shape Regression

Different from most previous segmentation-based and boundary point-based text region representation schemes used for scene text detection, which lack precise description and effective constraint for the global shape of one text, we propose a parametric, geometric text region modeling and regression scheme, which captures the holistic shape characteristics of a text to improve the text region localization accuracy. Specifically, as shown in Fig. 2, a text region is modeled by a n-order B-spline centerline describing the global layout of the text and a series of boundary cues capturing its local shape details.

The B-spline centerline is formulated as:

$$\boldsymbol{B}(t) = \sum_{i=0}^{m} \boldsymbol{P}_i \mathcal{N}_{i,n}(t) \tag{1}$$

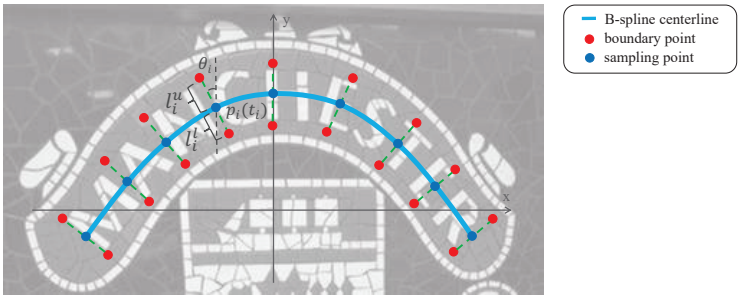

Fig. 2. Illustration of the spline-based representation of a text region.

where $\boldsymbol{P}_{i=0,\cdots,m}$ denote the $m+1$ control points of the B-spline, and $\mathcal{N}_{i,n}(t)$ is the basis function associated with $\boldsymbol{P}_i$, which is defined recursively with a predesignated knot vector $[\bar{t}_0, \bar{t}_1, \cdots, \bar{t}_{m+n}]$ as follows:

$$\mathcal{N}_{i,1}(t) = \begin{cases} 1, \bar{t}_i \leq t < \bar{t}_{i+1} \\ 0, \textit{otherwise} \end{cases}$$
$$\mathcal{N}_{i,n}(t) = \frac{t-\bar{t}_i}{\bar{t}_{i+n-1}-\bar{t}_i}\mathcal{N}_{i,n-1}(t) + \frac{\bar{t}_{i+n}-t}{\bar{t}_{i+n}-\bar{t}_{i+1}}\mathcal{N}_{i+1,n-1}(t) \quad (2)$$

In addition to the centerline, we further depict the contour of a text region with two sets of boundary points $\{v_i\}^u_{i=1..w}$ and $\{v_i\}^l_{i=1..w}$ on the upper and lower boundaries of the text region respectively as shown in Fig. 2. Each pair of two corresponding boundary points (v_i^u, v_i^l) are connected by a line segment L_i, and its length above and below the centerline are described by a pair of parameters l_i^u and l_i^l, the angle between L_i and the coordinate axis is described by a parameter θ_i, and the intersection point between L_i and the centerline (called a *sampling point*) is represented by its corresponding spline variable value t_i. Accordingly, a text region is geometrically described by the control points $\boldsymbol{P}_{i=0,\cdots,m}$ of the B-spline centerline and the parameters $\{t_i, l_i^u, l_i^l, \theta_i\}$ of the boundary points.

Our spline-based, geometric text region representation model differs essentially from the segmentation-based representations employed by previous scene text detection and spotting methods [41, 49, 5, 43]. The explicit parametric modeling of the global centerline provides effective shape constraints for robustly and accurately localizing text in cluttered scenes such as closely spaced or partially overlapping text instances, which are often challenging for segmentation-based detection methods. Moreover, compared to previous text representations that modeled a text region by its upper and lower boundaries like in [23], our centerline-based representation of the text region is usually less affected by variations of text geometry and style such as nonuniform sizes of characters in a text which often cause more significant changes to the boundary of the text region than to its centerline, and better captures the smoothness of the overall shape of a text.

We generate training labels for the parameters of the text region representation model in a similar manner to that adopted in [36] on the basis of the common

polygonal annotations of text region provided in most scene text benchmarks. Particularly, different from ABCNet which requires generating ground-truth labels for the control points of Bezier curve boundaries, we do not generate annotations for the B-spline centerline's control points. Instead, we generate ground-truth labels for a series of k *path points* located on the text centerline, which act as more direct constraints on the B-spline centerline and are easier to be inferred from text region features than the control points.

Text Region Regression Network. To infer the shape parameters of a text candidate in an image, as shown in Fig. 1, the text region regression network takes a text region proposal's feature maps generated by the ResNet50 [10], FPN [21], and RPN [32] backbone as input, and employs a Cascade R-CNN [2] to refine the proposal's position and assign it a text/non-text score. Next, the network employs three branches, each comprising several convolution, pooling, and full-connected layers, to predict the parameters of the B-spline centerline, the boundary points, and the text direction respectively. The detailed configuration of the network is given in the supplementary material.

Localization Loss. We employ a multitask text region localization loss $\mathcal{L}_{loc}$ on each text region proposal, which integrates a RPN loss L_{rpn} [32], a Cascade R-CNN loss L_{rcnn} [2], and a text region regression loss L_{reg}:

$$\mathcal{L}_{loc} = \lambda_1 L_{rpn} + \lambda_2 L_{rcnn} + \lambda_3 L_{reg} \tag{3}$$

where λ_1, λ_2, and λ_3 are set to 1.0.

The text region regression loss L_{reg} measures the approximation accuracy of the predicted text region relative to the ground-truth, which is formulated as the combination of a centerline loss L_{spline}, a boundary loss L_{bound}, and a text direction loss L_{dir}:

$$\begin{aligned} L_{reg}(\boldsymbol{P},\boldsymbol{T}_c,\boldsymbol{T}_b,\boldsymbol{\Theta},\boldsymbol{l},\boldsymbol{Q}^*,\boldsymbol{V}^*,\boldsymbol{\Theta}^*,\boldsymbol{l}^*,\boldsymbol{d},\boldsymbol{d}^*) &= \lambda_4 L_{spline}(\boldsymbol{P},\boldsymbol{T}_c,\boldsymbol{Q}^*) \\ &+ \lambda_5 L_{bound}(\boldsymbol{P},\boldsymbol{T}_b,\boldsymbol{\Theta},\boldsymbol{l},\boldsymbol{V}^*,\boldsymbol{\Theta}^*,\boldsymbol{l}^*) + \lambda_6 L_{dir}(\boldsymbol{d},\boldsymbol{d}^*) \end{aligned} \tag{4}$$

where $\boldsymbol{P} = \{\boldsymbol{P}_0, \cdots, \boldsymbol{P}_m\}$ are predicted control points of the B-spline centerline defined by Eq. (1). $\boldsymbol{T}_c = \{t_1^c, \cdots, t_k^c\}$ and $\boldsymbol{T}_b = \{t_1^b, \cdots, t_w^b\}$ are predicted spline variable values for the path points and the sampling points on the centerline respectively, while $\boldsymbol{Q}^*$ and $\boldsymbol{V}^*$ are the ground-truth coordinates of path points and boundary points respectively. $\boldsymbol{\Theta}, \boldsymbol{l} = [\boldsymbol{l}^u, \boldsymbol{l}^l]$ and $\boldsymbol{\Theta}^*, \boldsymbol{l}^*$ are the predicted and ground-truth angles and lengths of the lines connecting sampling points to corresponding boundary points respectively. $L_{dir}(\boldsymbol{d}, \boldsymbol{d}^*)$ is the binary cross-entropy loss between the predicted text direction probability vector $\boldsymbol{d}$ and the ground-truth one-hot direction label vector $\boldsymbol{d}^*$ which is generated for a text region based on the angle θ_t between the text's main axis (i.e. the line connecting the first and last path points) and the x axis to categorize it to horizontal if $\theta_t < 50°$ and vertical otherwise. The weights λ_4, λ_5, and λ_6 are experimentally set to 5.0, 5.0, and 0.5 respectively in this work.

The centerline loss L_{spline} measures how accurately the predicted B-spline centerline approximates the ground-truth path points $\boldsymbol{Q}^*$ and is formulated as:

$$L_{spline}(\boldsymbol{P}, \boldsymbol{T}_c, \boldsymbol{Q}^*) = smooth_{L1}(|\mathcal{F}(\boldsymbol{P}, \boldsymbol{T}_c) - \boldsymbol{Q}^*|) \tag{5}$$

where the function $\mathcal{F}(\boldsymbol{P}, \boldsymbol{T})$ computes a set of s output points corresponding to a set of spline variable values $\boldsymbol{T} = \{t_1, \cdots, t_s\}$, which are located on the B-spline defined by the control points $\boldsymbol{P} = \{\boldsymbol{P}_0, \cdots, \boldsymbol{P}_m\}$:

$$\mathcal{F}(\boldsymbol{P}, \boldsymbol{T}) = \begin{bmatrix} \boldsymbol{T}_1 \\ .. \\ \boldsymbol{T}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_{0,n} \ \boldsymbol{N}_{1,n} \ .. \ \boldsymbol{N}_{m,n} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_0 \\ \boldsymbol{P}_1 \\ .. \\ \boldsymbol{P}_m \end{bmatrix} \tag{6}$$

where $\boldsymbol{N}_{i,n}$ denotes the coefficient vector of the ith basis function of B-spline, and $\boldsymbol{T}_j = [t_j^{n-1}, t_j^{n-2}, \cdots, t_j^0]$ with t_j being the spline variable value for the jth output point. Therefore, $\mathcal{F}(\boldsymbol{P}, \boldsymbol{T}_c)$ yields the set of predicted path points.

The function $smooth_{L1}(\cdot)$ is defined as:

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

The boundary loss L_{bound} measures the accuracy of the predicted boundary points of text region relative to the ground-truth $\boldsymbol{V}^*$ and is formulated as:

$$\begin{aligned} L_{bound}(\boldsymbol{P}, \boldsymbol{T}_b, \boldsymbol{\Theta}, \boldsymbol{l}, \boldsymbol{V}^*, \boldsymbol{\Theta}^*, \boldsymbol{l}^*) &= smooth_{L1}(|\mathcal{G}(\boldsymbol{P}, \boldsymbol{T}_b, \boldsymbol{\Theta}, \boldsymbol{l}) - \boldsymbol{V}^*|) \\ &+ smooth_{L1}(sum(|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*|)) + smooth_{L1}(sum(|\boldsymbol{l} - \boldsymbol{l}^*|)) \end{aligned} \tag{8}$$

where the function $\mathcal{G}(\boldsymbol{P}, \boldsymbol{T}_b, \boldsymbol{\Theta}, \boldsymbol{l})$ computes w pairs of boundary points based on the set of sampling points computed by $\mathcal{F}(\boldsymbol{P}, \boldsymbol{T}_b)$ and the predicted parameters $\boldsymbol{\Theta}, \boldsymbol{l}$ of lines connecting sampling and boundary points. Moreover, we maintain two separate sets of $\boldsymbol{\Theta}, \boldsymbol{l}$ parameters to better capture shape characteristics of horizontal and vertical text respectively, and compute L_{bound} on the parameter set corresponding to the direction label $\boldsymbol{d}^*$.

3.2 Spatial Rectification of Text Features

To alleviate the difficulties caused by irregular text shapes (e.g., curved or perspectively distorted) to a text recognizer, we introduce an adaptive shape rectification module to spatially regularize the text's feature representation before feeding it to the recognizer for improved recognition accuracy. Different from most previous text rectification methods [34, 35, 30] which used spatial transform network (STN) [13] with thin-plate-spline (TPS) transformation to deform the text's shape, we employ a piecewise linear deformation model based on projection transformation for feature sampling and mapping to reduce non-linear distortion to the text's shape during rectification, while keeping sufficient deformation flexibility for widely varied shapes of scene text.

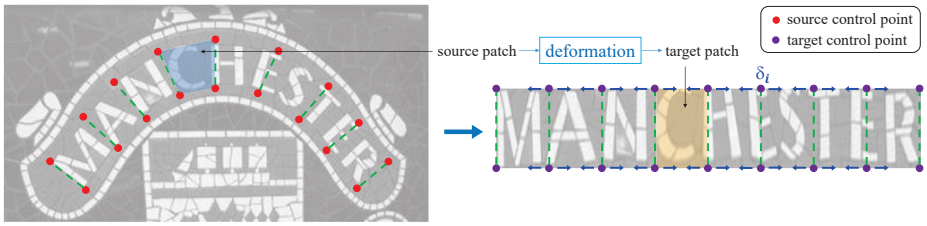

Fig. 3. Illustration of spatial rectification of irregular text. Note the deformation actually occurs on the feature maps rather than the image itself.

Specifically, given the predicted boundary points of a text region, as shown in Fig. 3, we first use the line connecting each pair of boundary points on the upper and lower text boundaries respectively to divide the feature map of the text region into a strip of adjacent quadrilateral patches (called source patches), each of which will be deformed individually.

Next, we map each source patch to a target patch in the output (rectified) feature map as shown in Fig. 3. Different from predefining a set of fixed-size target patches on the output feature map using a uniform grid as employed in previous methods [35, 48], we propose a variable target grid by predicting an offset δ for each grid point to allow a target patch's boundary to deviate adaptively from the uniform grid position, which increases the model's flexibility for rectifying non-uniform distortions of text. Note the offsets of the target grid points are end-to-end learned with the recognition task without any extra supervision.

Finally, we compute the feature values in a target patch by grid-sampling features in the corresponding source patch to obtain a regular feature representation of the text region for recognition.

Feature Patch Deformation. We employ projection transformation as the mapping function between the source and target patches because of its linearity which helps keep shape characteristics of character and the fact that most scene text has a certain degree of perspective distortion resulting from the viewing process.

Using the four boundary points of a source patch as four *source* control points and the four corner points of the corresponding target patch as *target* control points, the homogeneous deformation matrix $\mathcal{H}$ of a projection transformation for the patch can be formulated as:

$$\mathcal{H} = reshp([\boldsymbol{b} \quad 1])_{3\times 3} \tag{9}$$

where function $reshp(\cdot)_{3\times 3}$ reshapes the input tensor to a 3×3 view, and $\boldsymbol{b}$ is a 1×8 vector computed as:

$$\boldsymbol{b} = \mathcal{A}^{-1}\boldsymbol{x} \tag{10}$$

where $\boldsymbol{x}$ is an 8×1 vector containing the coordinates of the four target control points. $\mathcal{A}$ is an 8×8 matrix formulated as follows based on the Direct Linear

Transformation (DLT) algorithm [8]:

$$\mathcal{A} = \begin{bmatrix} r_x^{(0)} & r_y^{(0)} & 1 & 0 & 0 & 0 & -r_x^{(0)} * t_x^{(0)} & -r_y^{(0)} * t_x^{(0)} \\ 0 & 0 & 0 & r_x^{(0)} & r_y^{(0)} & 1 & -r_x^{(0)} * t_y^{(0)} & -r_y^{(0)} * t_y^{(0)} \\ \cdots & & & & & & & \\ r_x^{(3)} & r_y^{(3)} & 1 & 0 & 0 & 0 & -r_x^{(3)} * t_x^{(3)} & -r_y^{(3)} * t_x^{(3)} \\ 0 & 0 & 0 & r_x^{(3)} & r_y^{(3)} & 1 & -r_x^{(3)} * t_y^{(3)} & -r_y^{(3)} * t_y^{(3)} \end{bmatrix} \tag{11}$$

where $(r_x^{(i)}, r_y^{(i)})$ and $(t_x^{(i)}, t_y^{(i)})$ are the (x, y) coordinates of the ith source and target control points respectively.

Given the deformation matrix $\mathcal{H}$, a position $\boldsymbol{p}_t$ in the target patch is mapped back to the position $\boldsymbol{p}_r = \mathcal{H}^{-1}\boldsymbol{p}_t$ in the source patch. Accordingly, we compute the feature value in the position $\boldsymbol{p}_t$ in the target patch's feature map by bilinear interpolation of feature values neighbouring to $\boldsymbol{p}_r$ in the source feature map. This grid sampling operation is represented by the symbol 'G' in Fig. 1.

3.3 Text Recognition

Given the rectified feature maps of one text region, we employ a light-weight attention-based sequence-to-sequence recognition network to recognize the text. As shown in Fig. 1, the network first employs several convolutional layers to produce a feature map of height 1, and then uses a bidirectional LSTM to encode long-range forward and backward dependencies between the column feature vectors of the feature map and outputs a sequence of features. A gated recurrent unit (GRU) decoder with Bahdanau attention is finally employed to decode the feature sequence into a character label sequence. More details about character sequence prediction with GRU can be found in [28], and the configuration of the recognition network is presented in the supplementary material.

Recognition Loss. The text recognition loss $\mathcal{L}_{rec}$ is formulated as:

$$\mathcal{L}_{rec} = -\sum_{i=1}^{N}\sum_{j=1}^{NC} \mathbb{I}(\hat{\mathbf{y}}_i^j = 1) log(\mathbf{y}_i^j) \tag{12}$$

where N is the length of the predicted character label distribution sequence $\{\mathbf{y}_i\}$, NC is the total number of different characters, $\{\hat{\mathbf{y}}_i\}$ is the ground-truth one-hot label distribution sequence, and $\mathbb{I}(\cdot)$ is a binary function that returns 1 if its input is evaluated as true and returns 0 otherwise.

3.4 Text Spotting Loss

The total loss of the text spotting model is a combination of the text region localization loss $\mathcal{L}_{loc}$ and the text recognition loss $\mathcal{L}_{rec}$:

$$\mathcal{L} = \lambda_l \mathcal{L}_{loc} + \lambda_r \mathcal{L}_{rec} \tag{13}$$

where the weights λ_l and λ_r are set to 1.0 and 0.2 respectively in this work.

4 Experiments

4.1 Datasets

We evaluate our scene text spotting method on three challenging benchmarks: TotalText, CTW1500, and ICDAR2015. **TotalText** [4] is composed of 1255 and 300 images for training and testing respectively and contains large numbers of curved text instances, each annotated by a polygonal boundary of 10 vertices. **CTW1500** [24] contains 1000 training images and 500 testing images with many challenging long curved text, each annotated by a polygonal boundary of 14 vertices. **ICDAR2015** [14] consists of 1000 training images and 500 testing images with multi-oriented accidental scene text instances, each annotated by a quadrilateral bounding box. We employ precision P, recall R, and f-measure F to evaluate text spotting performance.

4.2 Implementation Details

We implement the proposed text spotting network on the basis of the PyTorch framework and conduct the experiments on a NVIDIA Tesla V100 GPU. We depict the text centerline by a cubic B-spline (order $n = 4$) with 5 control points ($m = 4$) and an open uniform knot vector, and approximate the centerline with $k = 17$ path points. We employ $w = 9$ pairs of boundary points on the upper and lower boundaries of a text region.

The spotting network is optimized using stochastic gradient descent with a weight decay of 0.0001 and a momentum of 0.9. The network is first pre-trained on a combined dataset similar to that used in [29] for 90K iterations with the learning rate starting from 0.01 and reduced to 0.001 for the last 20K iterations. The combined dataset contains training samples of SynthText [7], ICDAR 2013 [15], ICDAR 2015 [14], COCO-Text [38], and Total-Text [4] datasets, with a sampling ratio $2 : 2 : 2 : 2 : 1$ among these datasets for generating a mini-batch of 10. Next, we fine-tune separate spotting models for different test datasets using their own training sets. For TotalText and CTW1500 curved text datasets, the learning rate is initialized to 0.001 for the first 40K training iterations and is reduced to 0.0001 for further 20K iterations. For ICDAR2015 dataset, a learning rate of 0.001 is used during 40K training iterations of the network.

4.3 Ablation Study

Effectiveness of Spline-Based Text Region Regression. We verify the effectiveness of the proposed spline-based text region representation and regression model by comparing the text detection performance of some variants of the text region regression network in Table 1. The model 'Baseline' uses the Cascade R-CNN backbone to predict the bounding boxes of text instances in the image. The model 'Mask' replaces the shape parameter regression branches with the mask branch in Mask R-CNN [9] for text detection.

Table 1. Text detection performance of the proposed spline-based text region regression model and two variant models

Model	TotalText			ICDAR2015		
	P	R	F	P	R	F
Baseline	73.3	72.7	73.0	79.5	74.8	77.1
Mask	85.0	82.2	83.6	89.7	80.9	85.1
Proposed	**85.7**	**85.1**	**85.4**	**91.1**	**85.4**	**88.1**

Table 2. Text detection performance using variant number of control points for the B-spline centerline of a text region

Num	TotalText			ICDAR2015		
	P	R	F	P	R	F
4	**85.8**	84.5	85.1	89.0	**86.9**	87.9
5	85.7	**85.1**	**85.4**	**91.1**	85.4	**88.1**
6	85.5	85.0	85.3	90.2	85.6	87.8
7	85.1	84.6	84.8	88.4	86.5	87.4

Fig. 4. Text detection results obtained by variant models in Table 1.

Fig. 5. Text spotting results obtained by the baseline model (left) and our model (right) in Table 3. Detected text instances are marked with green boxes. Incorrect recognition results are shown with red text.

Compared to the baseline, the proposed spline-based text region regression model substantially improves the text detection performance through more accurate and flexible modeling of the text region. It also achieves higher detection f-measure than the mask mechanism [9], showing the advantages of the proposed parameterized, geometric representation of the text over the pixel-level representation in accurately describing the shape of the text. Figure 4 presents some text detection results obtained by variant models in Table 1. The proposed model yields more accurate text region boundaries than the others.

We further inspect the impact of using different numbers of control points for the B-spline centerline of a text region on the detection performance. As shown in Table 2, a B-spline with 5 control points is usually sufficient to cope with the different shape complexities of most scene text.

Effectiveness of Text Feature Rectification for Text Spotting. We verify the effectiveness of our text feature rectification mechanism in scene text spotting. Table 3 compares the spotting performance with our rectification model, a

Table 4. Scene text spotting results on TotalText and CTW1500. 'None' and 'Full' are f-measure of spotting using no lexicon and the full lexicon in recognition respectively. 'Det' is the f-measure of text detection results. 'FPS' is the inference speed on TotalText. In each column, the best result is shown in bold and the second best result is shown with underline. Methods marked with * exploited additional character-level labels besides the common word-level labels in training and are not included in ranking.

Method	TotalText			CTW1500			*FPS*
	Det	*None*	*Full*	*Det*	*None*	*Full*	
TextNet [37]	63.5	54.0	-	-	-	-	-
FOTS [22]	-	-	-	62.8	21.1	39.7	-
Qin *et al.*[31]	83.3	67.8	-	-	-	-	4.8
TextDragon [5]	80.3	48.8	74.8	83.6	39.7	72.4	-
ABCNet [23]	-	64.2	75.7	-	45.2	74.1	17.9
Text Perceptron [30]	85.2	<u>69.7</u>	78.3	84.6	57.0	-	-
PAN++ [42]	86.0	68.6	<u>78.6</u>	-	-	-	21.1
ABCNet v2 [25]	**87.0**	**70.4**	78.1	<u>84.7</u>	<u>57.5</u>	**77.2**	10
Mask TextSpotter [29] *	83.9	52.9	71.8	-	-	-	4.8
CharNet [46] *	85.6	66.6	-	-	-	-	-
Mask TextSpotter v3 [18] *	-	71.2	78.4	-	-	-	-
Ours	<u>86.6</u>	67.8	**80.0**	**84.9**	**59.6**	<u>75.0</u>	8.6

STN-based rectification model similar to [34] for adaptive text shape deformation, and a baseline model that removes the rectification module from the spotting network (i.e., feeding features of a text region directly to the text recognition module).

As shown in Table 3, introducing adaptive rectification of text features ahead of recognition significantly enhances the text spotting performance owing to the rectified, more regular representation of the text, especially on benchmarks with curved/irregular text instances like TotalText as expected. Figure 5 shows some examples of text spotting results obtained by the baseline model and our rectification-based model respectively. The improved spotting accuracy achieved by our model shows its effectiveness for arbitrary-shaped scene text spotting.

Table 3. Text spotting performance with variant rectification models

Model	TotalText			ICDAR2015		
	P	*R*	*F*	*P*	*R*	*F*
Baseline	75.8	73.0	74.4	69.9	69.0	69.5
STN	80.8	72.8	76.6	71.7	69.0	70.3
Ours	**81.2**	**78.9**	**80.0**	**72.2**	**69.3**	**70.7**

4.4 Comparison with State-of-the-Arts

We compare the performance of our text spotting method with some state-of-the-art methods on both curved and multi-oriented text benchmarks in Ta-

Table 5. Scene text spotting results on ICDAR2015. 'S', 'W', and 'G' are f-measure of spotting using the strong (100 words), weak (1000+ words), and generic (90K words) lexicons respectively. Methods marked with * exploited additional character-level labels besides the common word-level labels in training and are not included in ranking.

Method	Word Spotting			End-to-End Recognition			FPS
	S	W	G	S	W	G	
Deep TextSpotter [1]	58.0	53.0	51.0	54.0	51.0	47.0	9.0
TextBoxes++ [19]	76.5	69.0	54.4	73.3	65.9	51.9	-
FOTS [22]	84.7	79.3	63.3	81.1	75.9	60.8	7.5
He *et al.*[11]	85.0	80.0	65.0	82.0	77.0	63.0	-
TextDragon [5]	**86.2**	**81.6**	68.0	82.5	78.3	65.2	2.6
Text Perceptron [30]	84.1	79.4	67.9	80.5	76.6	65.1	-
PAN++ [42]	-	-	-	**82.7**	78.2	69.2	13.8
ABCNet v2 [25]	-	-	-	**82.7**	**78.5**	**73.0**	10
Mask TextSpotter [29] *	79.3	74.5	64.2	79.3	73.0	62.4	2.6
CharNet [46] *	-	-	-	83.1	79.2	69.1	-
Mask TextSpotter v3 [18] *	83.1	79.1	75.1	83.3	78.1	74.2	2.5
Ours	82.7	77.0	**70.7**	**82.7**	76.6	70.6	6.2

bles 4 and 5. Note that, besides the word-level annotations of text, some methods (marked with *) further exploited external character-level annotations as extra supervision information, which are not available in the benchmark datasets.

Curved Text Spotting. Table 4 shows that our method achieves the best results in two text spotting and one text detection tasks on TotalText and CTW1500 curved text datasets and comparable results in the rest of detection/spotting tasks, which demonstrate the method's capability to accurately localize and recognize various curved text in natural images.

Particularly, compared to Text Perceptron which combined a TPS-based feature rectification module with a focusing attention recognizer [3] and ABCNet which employed a Bezier curve-based feature sampling mechanism for recognizing irregular text, our rectification and spotting model achieves higher performance on most evaluation metrics on the two curved text benchmarks. ABCNet v2 further extended the ABCNet's backbone (e.g. introducing the BiFPN and CoordConv modules) and its training mechanism for enhanced performance. When the ResNet+FPN backbone of ABCNet is used, which is similar to that employed in our model, it achieves a text spotting f-measure of 67.4 on TotalText and 54.6 on CTW1500 using no lexicon [25]. On the other hand, unlike our method employing common word-level annotations of text as supervision information, Mask TextSpotter v3 exploited both word-level and character-level annotations (e.g. bounding boxes and category indices of characters) for training the model and employed a combinatory text recognition strategy integrating character-level pixel voting and spatial attention mechanisms.

Multi-Oriented Text Spotting. On ICDAR2015 which consists of multi-oriented but mostly straight text instances, as shown in Table 5, our method also achieves comparable text spotting performance among the methods that similarly exploit only word-level annotations of text and common training datasets. The good results of our method on the curved and multi-oriented text benchmarks demonstrate its effectiveness in spotting scene text in arbitrary shapes.

4.5 Qualitative Results

Figure 6 shows some text spotting results of our method. The proposed spotting network robustly detects and recognizes various scene text with largely varied appearances and qualities. More examples of scene text spotting results and discussions of limitations can be found in the supplementary material.

Fig. 6. Examples of text spotting results. Detected text instances are marked with green boxes, with corresponding recognition results shown nearby.

5 Conclusions

We present a method for accurately spotting arbitrary-shaped scene text in natural images. A parametric text representation and regression model based on the spline centerline is proposed to capture the distinctive global shape characteristics of text for robustly localizing text instances with varied appearances. The method further spatially rectifies the feature representation of an irregularly shaped text with an adaptive deformation model before feeding it to the text recognition network, which effectively improves the text spotting accuracy. In the future work, we will explore integrating effective language models with the recognition network and further improving the collaboration between detection and recognition modules for enhancing the performance of the method.

References

1. Busta, M., Neumann, L., Matas, J.: Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2223–2231 (2017)
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018)
3. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: Towards accurate text recognition in natural images. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5086–5094 (Oct 2017)
4. Chng, C.K., Chan, C.S.: Total-Text: A comprehensive dataset for scene text detection and recognition. In: ICDAR. pp. 935–942 (2017)
5. Feng, W., He, W., Yin, F., Zhang, X., Liu, C.: TextDragon: An end-to-end framework for arbitrary shaped text spotting. In: ICCV. pp. 9075–9084 (2019)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification. In: Proceedings of the 23rd International Conference on Machine Learning - ICML 2006. pp. 369–376 (2006)
7. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: CVPR. pp. 2315–2324 (2016)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, second edn. (2003)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988 (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An end-to-end textspotter with explicit alignment and attention. In: CVPR. pp. 5020–5029 (2018)
12. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. IJCV **116**(1), 1–20 (2016)
13. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS. pp. 2017–2025 (2015)
14. 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. pp. 1156–1160 (2015)
15. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazán, J., de las Heras, L.: ICDAR 2013 robust reading competition. In: ICDAR. pp. 1484–1493 (2013)
16. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2231–2239 (June 2016)
17. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: A simple and strong baseline for irregular text recognition. In: AAAI Conference on Artificial Intelligence. vol. 33, pp. 8610–8617 (Jul 2019)
18. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 706–722. Springer International Publishing, Cham (2020)
19. Liao, M., Shi, B., Bai, X.: Textboxes++: A single-shot oriented scene text detector. TIP **27**(8), 3676–3690 (2018)

20. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: A fast text detector with a single deep neural network. In: AAAI. pp. 4161–4167 (2017)
21. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. pp. 936–944 (2017)
22. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: fast oriented text spotting with a unified network. In: CVPR. pp. 5676–5685 (2018)
23. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In: CVPR. pp. 9806–9815 (2020)
24. Liu, Y., Jin, L., Zhang, S., Zhang, S.: Detecting curve text in the wild: New dataset and new solution. CoRR **abs/1712.02170** (2017), http://arxiv.org/abs/1712.02170
25. Liu, Y., Shen, C., Jin, L., He, T., Chen, P., Liu, C., Chen, H.: Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 8048–8064 (2022)
26. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. pp. 3431–3440 (2015)
27. 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. pp. 19–35. Springer International Publishing (2018)
28. Luo, C., Jin, L., Sun, Z.: MORAN: A multi-object rectified attention network for scene text recognition. Pattern Recognition **90**, 109–118 (2019)
29. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: ECCV. pp. 71–88 (2018)
30. Qiao, L., Tang, S., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F.: Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence. pp. 11899–11907 (2020)
31. Qin, S., Bissacco, A., Raptis, M., Fujii, Y., Xiao, Y.: Towards unconstrained end-to-end text spotting. In: ICCV. pp. 4703–4713 (2019)
32. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
33. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(11), 2298–2304 (Nov 2017)
34. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4168–4176 (Jun 2016)
35. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: An attentional scene text recognizer with flexible rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(9), 2035–2048 (Sep 2019)
36. Shi, J., Chen, L., Su, F.: Accurate arbitrary-shaped scene text detection via iterative polynomial parameter regression. In: Computer Vision – ACCV 2020. pp. 241–256 (2021)
37. Sun, Y., Zhang, C., Huang, Z., Liu, J., Han, J., Ding, E.: TextNet: Irregular text reading from images with an end-to-end trainable network. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) Computer Vision – ACCV 2018. pp. 83–99 (2019)
38. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.J.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. CoRR **abs/1601.07140** (2016)

39. Wang, H., Lu, P., Zhang, H., Yang, M., Bai, X., Xu, Y., He, M., Wang, Y., Liu, W.: All you need is boundary: Toward arbitrary-shaped text spotting. In: AAAI. pp. 12160–12167 (2020)
40. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: Proceedings of the 21st International Conference on Pattern Recognition. pp. 3304–3308 (2012)
41. 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. pp. 9336–9345 (2019)
42. Wang, W., Xie, E., Li, X., Liu, X., Liang, D., Yang, Z., Lu, T., Shen, C.: Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9), 5349–5367 (2022)
43. 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. pp. 8439–8448 (2019)
44. Wang, X., Jiang, Y., Luo, Z., Liu, C., Choi, H., Kim, S.: Arbitrary shape scene text detection with adaptive text region representation. In: CVPR. pp. 6449–6458 (2019)
45. Wu, Y., Natarajan, P.: Self-organized text detection with minimal post-processing via border learning. In: ICCV. pp. 5010–5019 (2017)
46. Xing, L., Tian, Z., Huang, W., Scott, M.R.: Convolutional character networks. In: ICCV. pp. 9125–9135 (2019)
47. Yang, M., Guan, Y., Liao, M., He, X., Bian, K., Bai, S., Yao, C., Bai, X.: Symmetry-constrained rectification network for scene text recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9146–9155 (2019)
48. Zhan, F., Lu, S.: ESIR: End-to-end scene text recognition via iterative image rectification. In: CVPR. pp. 2054–2063 (Jun 2019)
49. 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. pp. 10544–10553 (2019)
50. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: An efficient and accurate scene text detector. In: CVPR. pp. 2642–2651 (2017)