ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network*

Yuliang Liu‡†,   Hao Chen†,   Chunhua Shen†,   Tong He†,   Lianwen Jin‡,   Liangwei Wang◇

‡South China University of Technology   †University of Adelaide, Australia   ◇Huawei Noah's Ark Lab

Abstract

Scene text detection and recognition has received increasing research attention. Existing methods can be roughly categorized into two groups: character-based and segmentation-based. These methods either are costly for character annotation or need to maintain a complex pipeline, which is often not suitable for real-time applications. Here we address the problem by proposing the Adaptive Bezier-Curve Network (ABCNet). Our contributions are three-fold: 1) For the first time, we adaptively fit oriented or curved text by a parameterized Bezier curve. 2) We design a novel BezierAlign layer for extracting accurate convolution features of a text instance with arbitrary shapes, significantly improving the precision compared with previous methods. 3) Compared with standard bounding box detection, our Bezier curve detection introduces negligible computation overhead, resulting in superiority of our method in both efficiency and accuracy.

Experiments on oriented or curved benchmark datasets, namely Total-Text and CTW1500, demonstrate that ABCNet achieves state-of-the-art accuracy, meanwhile significantly improving the speed. In particular, on Total-Text, our real-time version is over 10 times faster than recent state-of-the-art methods with a competitive recognition accuracy.

Code is available at https://git.io/AdelaiDet.

(a) Segmentation-based method.   (b) Our proposed ABCNet.

Figure 1. Segmentation-based results are easily affected by nearby text. The nonparametric non-structured segmentation results make them very difficult to align features for the subsequent recognition branch. Segmentation-based results usually need complex post-processing, hampering efficiency. Benefiting from the parameterized Bezier curve representation, our ABCNet can produce structured detection regions and thus the BezierAlign sampling process can be used for naturally connecting the recognition branch.

1. Introduction

Scene text detection and recognition has received increasing attention due to its numerous applications in computer vision. Despite tremendous progress has been made recently [10, 42, 28, 36, 27, 43, 45, 41, 46, 14], detecting and recognizing text in the wild remains largely unsolved due to its diversity patterns in sizes, aspect ratios, font styles, perspective distortion, and shapes. Although the emergence of deep learning has significantly improved the performance of the task of scene text spotting, a considerable gap still exists in current methods for real-world

applications, especially in terms of efficiency.

Recently, many end-to-end methods [31, 37, 34, 24, 44, 21] have significantly improved the performance of oriented or curved scene text spotting. However, these methods either use segmentation-based approaches that maintain a complex pipeline or require a large amount of expensive character-level annotations. In addition, almost all of these methods are slow in inference, hampering the deployment to real-time applications. Thus, our motivation is to design a *simple yet effective* end-to-end framework for spotting oriented or curved scene text in images [4, 27], which ensures fast inference time while achieving an *on par* or even better performance compared with state-of-the-art methods.
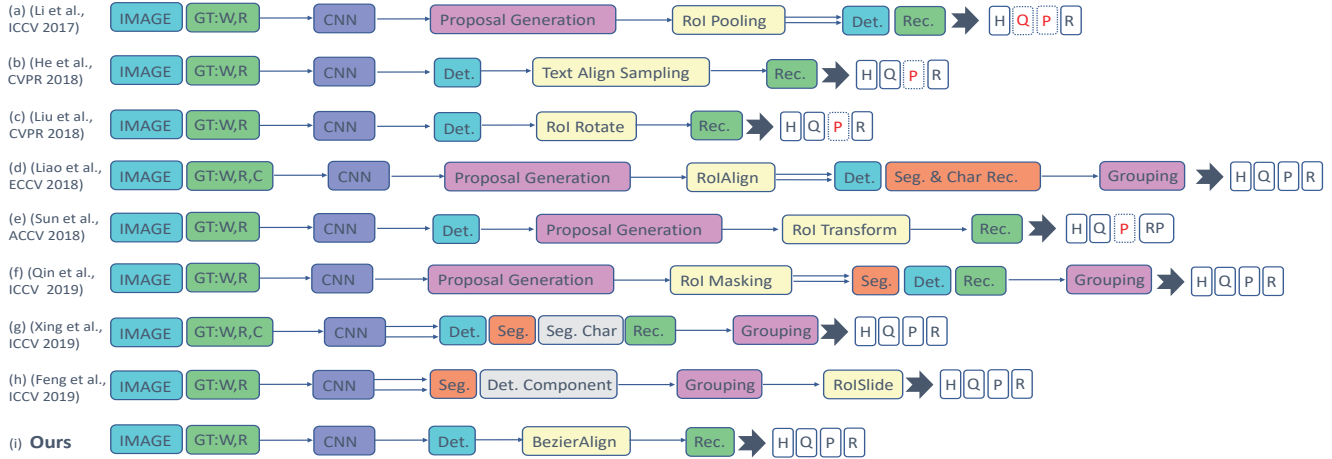

Figure 2. Overview of some end-to-end scene text spotting methods that are most relevant to ours. Inside the GT (ground-truth) box, 'W', 'R', and 'C' represent word-level annotation, text content, and character-level annotation, respectively. 'H', 'Q', and 'P' represent that the method is able to detect horizontal, quadrilateral, and oriented or curved text, respectively. 'RP' means that the method can recognize the curved text inside a quadrilateral box. 'R': recognition; 'BBox': bounding box. Dashed box represents the shape of the text which the method is unable to detect.

To achieve this goal, we propose the Adaptive Bezier Curve Network (ABCNet), an end-to-end trainable framework, for oriented or curved scene text spotting. ABCNet enables oriented or curved scene text detection with Bezier curve adaptation, which introduces negligible computation overhead compared with standard rectangle bounding box detection. In addition, we design a novel feature alignment layer—BezierAlign—to precisely calculate convolutional features of text instances in curved shapes, and thus high recognition accuracy can be achieved without introducing much computation cost. For the first time, we represent the oriented or curved text with parameterized Bezier curves, and the results show the effectiveness of our method. Examples of our spotting results are shown in Figure 1.

Note that previous methods such as TextAlign [11] and FOTS [25] can be viewed as a special case of ABCNet because a quadrilateral bounding box can be seen as the simplest oriented or curved bounding box with 4 straight boundaries. In addition, ABCNet can avoid complicated transformation such as 2D attention [20], making the design of the recognition branch considerably simpler.

We summarize our main contributions as follows.

- In order to accurately localize oriented and curved scene text in images, for the first time, we introduce a new concise parametric representation of curved scene text using Bezier curves. It introduces negligible computation overhead compared with the standard bounding box representation.

- We propose a sampling method, a.k.a. BezierAlign, for accurate feature alignment, and thus the recognition branch can be naturally connected to the overall struc-

ture. By sharing backbone features, the recognition branch can be designed with a light-weight structure.

- The simplicity of our method allows it to perform inference in real time. ABCNet achieves state-of-the-art performance on two challenging datasets, Total-Text and CTW1500, demonstrating advantages in both effectiveness and efficiency.

1.1. Related Work

Scene text spotting requires detecting and recognizing text simultaneously instead of concerning only one task. Recently, the emergence of deep-learning-based methods have significantly advanced the performance of text spotting. Both the detection and recognition have been dramatically improved in performance. We summarized several representative deep-learning-based scene text spotting methods into the following two categories. Figure 2 shows an overview of typical works.

Regular End-to-end Scene Text Spotting Li et al. [19] propose the first deep-learning based end-to-end trainable scene text spotting method. The method successfully uses a RoI Pooling [35] to joint detection and recognition features via a two-stage framework, but it can only spot horizontal and focused text. Its improved version [20] significantly improves the performance, but the speed is limited. He et al. [11] and Liu et al. [25] adopt an anchor-free mechanism to improve both the training and inference speed. They use a similar sampling strategy, i.e., Text-Align-Sampling and RoI-Rotate, respectively, to enable extracting feature from quadrilateral detection results. Note that both of these two methods are not capable of spotting oriented or curved scene text.
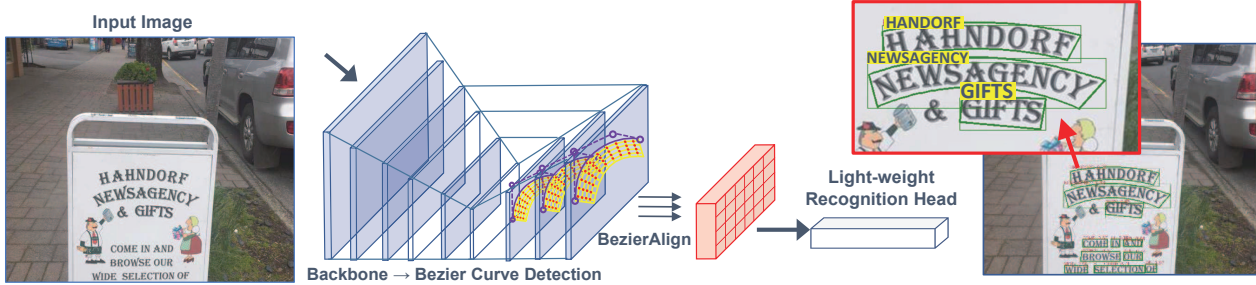
Figure 3. The framework of the proposed ABCNet. We use cubic Bezier curves and BezierAlign to extract curved sequence features using the Bezier curve detection results. The overall framework is end-to-end trainable with high efficiency. Purple dots represent the control points of the cubic Bezier curve.

Oriented or curved End-to-end Scene Text Spotting To detect oriented or curved scene text, Liao et al. [31] propose a Mask TextSpotter which subtly refines Mask R-CNN and uses character-level supervision to simultaneously detect and recognize characters and instance masks. The method significantly improves the performance of spotting oriented or curved scene text. However, the character-level ground truths are expensive, and using free synthesized data is hard to produce character-level ground truth for real data in practice. Its improved version [21] significantly alleviated the reliance for the character-level ground truth. The method relies on a region proposal network, which restricts the speed to some extent. Sun et al. [37] propose the TextNet which produces quadrilateral detection bounding boxes in advance, and then use a region proposal network to feed the detection features for recognition. Although the method can directly recognize the oriented or curved text from a quadrilateral detection, the performance is still limited.

Recently, Qin et al. [34] propose to use a RoI Masking to focus on the oriented or curved text region. However, the results may easily be affected by outlier pixels. In addition, the segmentation branch increases the computation burden; the fitting polygon process also introduces extra time consumption; and the grouping result is usually jagged and not smooth. The work in [24] is the first one-stage oriented or curved scene text spotting method, requiring character-level ground truth data for training. Authors of [44] propose a novel sampling method, RoISlide, which uses fused features from the predicting segments of the text instances, and thus it is robust to long oriented or curved text.

2. Adaptive Bezier Curve Network (ABCNet)

ABCNet is an end-to-end trainable framework for spotting oriented or curved scene text. An intuitive pipeline can be seen in Figure 3. Inspired by [49, 38, 12], we adopt a single-shot, anchor-free convolutional neural network as the detection framework. Removal of anchor boxes significantly simplifies the detection for our task. Here the detection

is densely predicted on the output feature maps of the detection head, which is constructed by 4 stacked convolution layers with stride of 1, padding of 1, and 3×3 kernels. Next, we present the key components of the proposed ABCNet in two parts: 1) Bezier curve detection; and 2) BezierAlign and recognition branch.

2.1. Bezier Curve Detection

Compared to segmentation-based methods [41, 46, 1, 39, 47, 29], regression-based methods are more direct solutions to oriented or curved text detection, e.g., [27, 43]. However, previous regression-based methods require complicated prediction to fit the text boundary, which is not very efficient and robust for the various text shapes in practice.

To simplify the oriented or curved scene text detection, following the regression method, we find that Bezier curve, a most fundamental concept of curve representation, is suitable for parameterization of curved text. The Bezier curve represents a parametric curve $c(t)$ that uses the Bernstein Polynomials [30] as its basis. The definition is shown in Equation (1).

$$c(t) = \sum_{i=0}^{n} b_i B_{i,n}(t), 0 \leq t \leq 1, \tag{1}$$

where, n represents the degree, b_i represents the i-th control points, and $B_{i,n}(t)$ represents the Bernstein basis polynomials, as shown in Equation (2):

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, ..., n, \tag{2}$$

where $\binom{n}{i}$ is a binomial coefficient. To fit arbitrary shapes of the text with Bezier curves, we comprehensively observe oriented or curved scene text from the existing datasets and the real world, and we empirically show that a cubic Bezier curve (i.e., n is 3) is sufficient to fit different kinds of the oriented or curved scene text in practice. An illustration of cubic Bezier curve is shown in Figure 4.
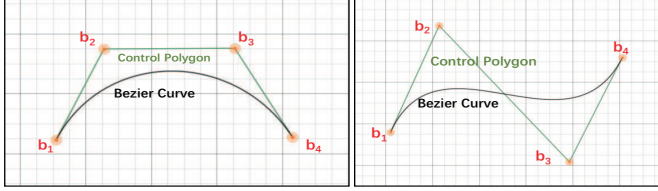
Figure 4. Cubic Bezier curves. b_i represents the control points. The green lines forms a control polygon, and the black curve is the cubic Bezier curve. Note that with only two end-points b_1 and b_4 the Bezier curve degenerates to a straight line.

Based on the cubic Bezier curve, we can simplify the oriented or curved scene text detection to a bounding box regression with eight control points in total. Note that a straight text that has four control points (four vertexes) is a typical case of oriented or curved scene text. For consistency, we interpolate additional two control points in the tripartite points of each long side.

To learn the coordinates of the control points, we first generate the Bezier curve ground truths described in 2.1.1 and follow a similar regression method as in [26] to regress the targets. For each text instance, we use

$$\Delta_x = b_{ix} - x_{min}, \Delta_y = b_{iy} - y_{min}, \quad (3)$$

where x_{min} and y_{min} represent the minimum x and y values of the 4 vertexes, respectively. The advantage of predicting the relative distance is that it is irrelevant to whether the Bezier curve control points are beyond the image boundary. Inside the detection head, we only need one convolution layer with 16 outputted channels to learn the Δ_x and Δ_y, which is nearly cost-free while the results can still be accurate, which will be discussed in Section 3.

2.1.1 Bezier Ground Truth Generation

In this section, we briefly introduce how to generate Bezier curve ground truth based on the original annotations. The oriented or curved datasets, e.g., Total-text [4] and CTW1500 [27], use polygonal annotations for the text regions. Given the annotated points $\{p_i\}_{i=1}^n$ from the curved boundary, where p_i represents the i-th annotating point, the main goal is to obtain the optimal parameters for cubic Bezier curves $c(t)$ in Equation (1). To achieve this, we can simply apply standard least square method, as shown in Equation (4):

$$\begin{bmatrix} B_{0,3}(t_0) & \cdots & B_{3,3}(t_0) \\ B_{0,3}(t_1) & \cdots & B_{3,3}(t_1) \\ \vdots & \ddots & \vdots \\ B_{0,3}(t_m) & \cdots & B_{3,3}(t_m) \end{bmatrix} \begin{bmatrix} b_{x_0} & b_{y_0} \\ b_{x_1} & b_{y_1} \\ b_{x_2} & b_{y_2} \\ b_{x_3} & b_{y_3} \end{bmatrix} = \begin{bmatrix} p_{x_0} & p_{y_0} \\ p_{x_1} & p_{y_1} \\ \vdots & \vdots \\ p_{x_m} & p_{y_m} \end{bmatrix}$$

$$(4)$$

Here m represents the number of annotated points for a curved boundary. For Total-Text and CTW1500, m is 5 and

(a) Original ground truth.   (b) Generated results.

Figure 5. Comparison of Bezier curve generation. In Figure (b), for each curve boundary, the red dash lines form a control polygon, and the red dots represent the control points. Warping results are showed below. In Figure (a), we utilize TPS [2] and STN [15] to warp the original ground truth into rectangular shape. In Figure (b), we use generated Bezier curves and our BezierAlign to warp the results.
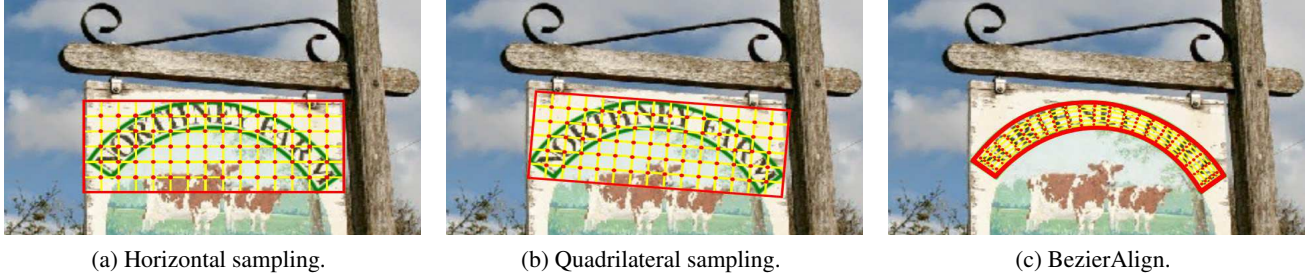
7, respectively. t is calculated by using the ratio of the cumulative length to the perimeter of the polyline. According to Equation (1) and Equation (4), we convert the original polyline annotation to a parameterized Bezier curve. Note that we directly use the first and the last annotating points as the first (b_0) and the last (b_4) control points, respectively. A visualization comparison is shown in the Figure 5, which shows that the generating results can be even visually better than the original ground truth. In addition, based on the structured Bezier curve bounding box, we can easily using our BezierAlign described in Section 2.2 to warp the curved text into a horizontal format without dramatic deformation. More examples of the Bezier curve generation results are shown in Figure 6. The simplicity of our method allows it generalize to different kinds of text in practice.

2.1.2 Bezier Curve Synthetic Dataset

For the end-to-end scene text spotting methods, a massive amount of free synthesized data is always necessary, as shown in Table 2. However, the existing 800k SynText dataset [7] only provides quadrilateral bounding box for a majority of straight text. To diversify and enrich the oriented or curved scene text, we make some effort to synthesize 150k synthesized dataset (94,723 images contain a majority of straight text, and 54,327 images contain mostly curved text) with the VGG synthetic method [7]. Specially, we filter out 40k text-free background images from COCO-Text [40] and then prepare the segmentation mask and scene depth of each background image with [33] and [18] for the following text rendering. To enlarge the shape diversity of synthetic texts, we modify the VGG synthetic method by synthesizing scene text with various art fonts and corpus and generate the polygonal annotation for all the text instances. The annotations are then used for producing Bezier curve ground truth by the generating method described in

Figure 6. Example results of Bezier curve generation. Green lines are the final Bezier curve results. Red dash lines represent the control polygon, and the 4 red end points represent the control points. Zoom in for better visualization.

(a) Horizontal sampling.   (b) Quadrilateral sampling.   (c) BezierAlign.

Figure 7. Comparison between previous sampling methods and BezierAlign. The proposed BezierAlign can accurately sample features of the text region, which is essential for recognition training. Note that the align procedure is processed in intermediate convolutional features.

Section 2.1.1. Examples of our synthesized data are shown in Figure 8.

Figure 8. Examples of cubic Bezier curve synthesized data.

2.2. BezierAlign

To enable end-to-end training, most of the previous methods adopt various sampling (feature alignment) methods to connect the recognition branch. Typically a sampling method represents an in-network region cropping procedure. In other words, given a feature map and Region-of-Interest (RoI), using the sampling method to select the features of RoI and efficiently output a feature map of a fixed size. However, sampling methods of previous non-segmentation based methods, e.g., RoI Pooling [19], RoI-Rotate [25], Text-Align-Sampling [11], or RoI Transform [37] cannot properly align features of oriented or curved text (RoISlide [44] numerous predicting segments). By exploiting the parameterization nature of a compact Bezier curve bounding box, we propose BezierAlign for feature sampling. BezierAlign is extended from RoIAlign [8]. Un-

like RoIAlign, the shape of sampling grid of BezierAlign is not rectangular. Instead, each column of the oriented or curved grid is orthogonal to the Bezier curve boundary of the text. The sampling points have equidistant interval in width and height, respectively, which are bilinear interpolated with respect to the coordinates.

Formally given an input feature map and Bezier curve control points, we concurrently process all the output pixels of the rectangular output feature map with size $h_{out} \times w_{out}$. Taking pixel g_i with position (g_{iw}, g_{ih}) (from output feature map) as an example, we calculate t by Equation (5):

$$t = \frac{g_{iw}}{w_{out}}. \tag{5}$$

We then use t and Equation (1) to calculate the point of upper Bezier curve boundary tp and lower Bezier curve boundary bp. Using tp and bp, we can linearly index the sampling point op by Equation (6):

$$op = bp \cdot \frac{g_{ih}}{h_{out}} + tp \cdot (1 - \frac{g_{ih}}{h_{out}}). \tag{6}$$

With the position of op, we can easily apply bilinear interpolation to calculate the result. Comparisons among previous sampling methods and BezierAlign are shown in Figure 7.

Recognition branch. Benefiting from the shared backbone feature and BezierAlign, we design a light-weight recognition branch as shown in Table 1, for faster execution. It consists of 6 convolutional layers, 1 bidirectional

Layers (CNN - RNN)	Parameters (kernel size, stride)	Output Size (n, c, h, w)
conv layers $\times 4$	(3, 1)	$(n, 256, h, w)$
conv layers $\times 2$	(3, (2,1))	$(n, 256, h, w)$
average pool for h	-	$(n, 256, 1, w)$
Channels-Permute	-	$(w, n, 256)$
BLSTM	-	$(w, n, 512)$
FC	-	(w, n, n_{class})

Table 1: Structure of the recognition branch, which is a simplified version of CRNN [36]. For all convolutional layers, the padding size is restricted to 1. n represents batch size. c represents the channel size. h and w represent the height and width of the outputted feature map, and n_{class} represents the number of the predicted class, which is set to 97 in this paper, including upper and lower cases of English characters, digits, symbols, one category representing all other symbols, and an "EOF" of the last category.

LSTM [13] layer, and 1 fully connected layer. Based on the output classification scores, we use a classic CTC Loss [6] for text string (GT) alignment. Note that during training, we directly use the generated Bezier curve GT to extract the RoI features. Therefore the detection branch does not affect the recognition branch. In the inference phase, the RoI region is replaced by the detecting Bezier curve described in Section 2.1. Ablation studies in Experimental Section 3 demonstrate that the proposed BezierAlign can significantly improve the recognition performance.

3. Experiments

We evaluate our method on two recently introduced oriented or curved scene text benchmarks, Total-Text [3] and CTW1500 [27], which also contain a large amount of straight text. We also conduct ablation studies on Total-Text to verify the effectiveness of our proposed method.

3.1. Implemented details

The backbone of this paper follows a common setting as most of the previous papers, i.e., ResNet-50 [9] together with a Feature Pyramid Network (FPN) [23]. For detection branch, we utilize RoIAlign on 5 feature maps with 1/8, 1/16, 1/32, 1/64, and 1/128 resolution of the input image while for recognition branch, BezierAlign is conducted on three feature maps with 1/4, 1/8, and 1/16 sizes. The pretrained data is collected from publicly available English word-level-based datasets, including 150k synthesized data described in Section 2.1.2, 15k images filtered from the original COCO-Text [40], and 7k ICDAR-MLT data [32]. The pretrained model is then finetuned on the training set of the target datasets. In addition, we also adopt data augmentation strategies, e.g., random scale training, with the short size randomly being chosen from 560 to 800 and the long size being less than 1333; and random crop, which we make

sure that the crop size is larger than half of the original size and without any text being cut (for some special cases that hard to meet the condition, we do not apply random crop).

We train our model using 4 Tesla V100 GPUs with the image batch size of 32. The maximum iteration is 150K; and the initialized learning rate is 0.01, which reduces to 0.001 at the 70K$^{\text{th}}$ iteration and 0.0001 at 120K$^{\text{th}}$ iteration. The whole training process takes about 3 days.

3.2. Experimental results on Total-Text

Dataset. Total-text dataset [3] is one of the most important oriented or curved scene text benchmark proposed in 2017, which was collected form various scenes, including text-like scene complexity and low-contrast background. It contains 1,555 images, with 1,255 for training and 300 for testing. To resemble the real-world scenarios, most of the images of this dataset contain a large amount of regular text while guarantee that each image has at least one curved text. The text instance is annotated with polygon based on word-level. Its extended version [4] improves its annotation of training set by annotating each text instance with a fixed ten points following text recognition sequence. The dataset contains English text only. To evaluate the end-to-end results, we follow the same metric as previous methods, which use F-measure to measure the word-accuracy.

Ablation studies: BezierAlign. To evaluate the effectiveness of the proposed components, we conduct ablation studies on this dataset. We first conduct sensitivity analysis of how the number of the sampling points may affect the end-to-end results, which is shown in Table 4. From the results we can see that the number of sampling points can significantly affect the final performance and efficiency. We find (7,32) achieves the best trade-off between F-measure and FPS, which is used as the final setting in the following experiments. We further evaluate BezierAlign by comparing it with previous sampling method shown in Figure 7. The results shown in Table 3 demonstrate that the BezierAlign can dramatically improve the end-to-end results. Qualitative examples are shown in Figure 9.

Ablation studies: Bezier curve detection. Another important component is Bezier curve detection, which enables oriented or curved scene text detection. Therefore, we also conduct experiments to evaluate the time consumption of Bezier curve detection. The result in Table 5 shows that the Bezier curve detection does not introduce extra computation compared with standard bounding box detection.

Comparison with state-of-the-art. We further compare our method to previous methods. From the Table 2, we can see that our single scale result (short size being 800) can achieve a competitive performance meanwhile achieving a real time inference speed, resulting in a better trade-off between speed and word-accuracy. With multi-scale inference, ABCNet achieves state-of-the-art performance, sig-

Method	Data	Backbone	F-measure		FPS
			None	Full	
TextBoxes [22]	SynText800k, IC13, IC15, TT	ResNet-50-FPN	36.3	48.9	1.4
Mask TextSpotter'18 [31]	SynText800k, IC13, IC15, TT	ResNet-50-FPN	52.9	71.8	4.8
Two-stage [37]	SynText800k, IC13, IC15, TT	ResNet-50-SAM	45.0	-	-
TextNet [37]	SynText800k, IC13, IC15, TT	ResNet-50-SAM	54.0	-	2.7
Li et al. [20]	SynText840k, IC13, IC15, TT, MLT, AddF2k	ResNet-101-FPN	57.80	-	1.4
Mask TextSpotter'19 [21]	SynText800k, IC13, IC15, TT, AddF2k	ResNet-50-FPN	65.3	77.4	2.0
Qin et al. [34]	SynText200k, IC15, COCO-Text, TT, MLT Private: 30k (manual label), 1m (partial label)	ResNet-50-MSF	67.8	-	4.8
CharNet [24]	SynText800k, IC15, MLT, TT	ResNet-50-Hourglass57	66.2	-	1.2
TextDragon [44]	SynText800k, IC15, TT	VGG16	48.8	74.8	-
ABCNet-F	SynText150k, COCO-Text, TT, MLT	ResNet-50-FPN	61.9	74.1	**22.8**
ABCNet			64.2	75.7	17.9
ABCNet-MS			**69.5**	**78.4**	6.9

Table 2: Scene text spotting results on Total-Text. ABCNet-F is faster as the short size of input image is 600. MS: multi-scale testing. "None" represents recognition without any lexicon. "Full" lexicon contains all words in test set. Datasets: AddF2k [48]; IC13 [17]; IC15 [16]; TT [5]; MLT [32]; COCO-Text [40].

Methods	Sampling method	F-measure (%)
ABCNet	with Horizontal Sampling	38.4
ABCNet	with Quadrilateral Sampling	44.7
ABCNet	with BezierAlign	61.9

Table 3: Ablation study for BezierAlign. Horizontal sampling follows [19], and quadrilateral sampling follows [11].

Method	Sampling points (n_h, n_w)	F-measure (%)	FPS
ABCNet	+ (6, 32)	59.6	**23.2**
	+ (7, 32)	**61.9**	22.8
	+ (14, 64)	58.1	19.9
	+ (21, 96)	54.8	18.0
	+ (28, 128)	53.4	15.1
	+ (30, 30)	59.9	21.4

Table 4: Ablation study of the number of sampling points of BezierAlign.

Methods	Inference time
without Bezier curve detection	22.8 fps
with Bezier curve detection	22.5 fps

Table 5: Ablation study for time consumption of the Bezier curve detection.

Figure 9. Qualitative recognition results of the quadrilateral sampling method and BezierAlign. Left: original image. Top right: results by using quadrilateral sampling. Bottom right: results by using BezierAlign.

nificantly outperforming all previous methods especially in the running time. It is worth mentioning that our faster version can be more than 11 times faster than previous best method [21] with *on par* accuracy.

Qualitative Results. Some qualitative results of ABC-Net are shown in Figure 10. The results show that our method can accurately detect and recognize most of the oriented or curved text. In addition, our method can also well handle straight text, with nearly quadrilateral compact bounding box and correct recognize results. Some errors are also visualized in the figure, which are mainly caused by mistakenly recognizing one of the characters.

3.3. Experimental Results on CTW1500

Dataset. CTW1500 [27] is another important oriented or curved scene text benchmark proposed in 2017. Compared to Total-Text, this dataset contains both English and Chinese text. In addition, the annotation is based on text-line level, and it also includes some document-like text, i.e., numerous small text may stack together. CTW1500 contains 1k training images, and 500 testing images.

Experiments. Because the occupation of Chinese text in this dataset is very small, we directly regard all the Chi-

Figure 10. Qualitative results of ABCNet on the Total-text. The detection results are shown with red bounding boxes. The float number is the predicted confidence. Zoom in for better visualization.

Figure 11. Qualitative end-to-end spotting results of CTW1500. Better viewed on screen.

Methods	Data	F-measure	
		None	Strong Full
FOTS [25]	SynText800k, CTW1500	21.1	39.7
Two-Stage* [44]	SynText800k, CTW1500	37.2	69.9
RoIRotate* [44]	SynText800k, CTW1500	38.6	70.9
LSTM* [44]	SynText800k, CTW1500	39.2	71.5
TextDragon [44]	SynText800k, CTW1500	39.7	72.4
ABCNet	SynText150k, CTW1500	**45.2**	**74.1**

Table 6: End-to-end scene text spotting results on CTW1500. * represents the results are from [44]. "None" represents lexicon-free. "Strong Full" represents that we use all the words appeared in the test set.

nese text as "unseen" class during training, i.e., the 96-th class. Note that the last class, i.e., the 97-th class is "EOF" in our implementation. We follow the same evaluation metric as [44]. The experimental results are reported in Table 6, which demonstrate that in terms of end-to-end scene text spotting, the ABCNet can significantly surpass previous state-of-the-art methods. Examples results of this dataset are showed in Figure 11. From the figure, we can see that some long text-line instances contain many words, which make a full-match word-accuracy extremely difficult. In other words incorrectly recognizing one character will result in zero scores for the whole text.

4. Conclusion

We have proposed ABCNet—a real-time end-to-end method that uses Bezier curves for oriented or curved scene text spotting. By reformulating oriented or curved scene text using parameterized Bezier curves, ABCNet can detect oriented or curved scene text with Bezier curves which introduces negligible computation cost compared with standard bounding box detection. With such regular Bezier curve bounding boxes, we can naturally connect a lightweight recognition branch via a new BezierAlign layer.

In addition, by using our Bezier curve synthesized dataset and publicly available data, experiments on two oriented or curved scene text benchmarks (Total-Text and CTW1500) demonstrate that our ABCNet can achieve state-of-the-art performance, which is also significantly faster than previous methods.

Acknowledgements

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character Region Awareness for Text Detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9365–9374, 2019.

[2] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[3] C.-K Chng and C.-S Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *Proc. IAPR Int. Conf. Document Analysis Recog.*, pages 935–942, 2017.

[4] Chee-Kheng Chng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: toward orientation robustness in scene text detection. *Int. J. Document Analysis Recogn.*, pages 1–22, 2019.

[5] Chee-Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. ICDAR2019 Robust Reading Challenge on Arbitrary-Shaped Text (RRC-ArT). *Proc. IAPR Int. Conf. Document Analysis Recog.*, 2019.

[6] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. Int. Conf. Mach. Learn.*, pages 369–376. ACM, 2006.

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2315–2324, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask R-CNN. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016.

[10] Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. Image Process.*, 25(6):2529–2541, 2016.

[11] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5020–5029, 2018.

[12] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[13] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. In *Neural Computation*, volume 9, pages 1735–1780, 1997.

[14] Zhida Huang, Zhuoyao Zhong, Lei Sun, and Qiang Huo. Mask r-cnn with pyramid attention network for scene text detection. In *Winter Conf. Appl. Comp. Vision*, pages 764–772. IEEE, 2019.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2017–2025, 2015.

[16] D. Karatzas, L. Gomez-Bigorda, et al. ICDAR 2015 competition on robust reading. In *Proc. IAPR Int. Conf. Document Analysis Recog.*, pages 1156–1160, 2015.

[17] D. Karatzas, F. Shafait, S. Uchida, et al. ICDAR 2013 Robust Reading Competition. In *Proc. IAPR Int. Conf. Document Analysis Recog.*, pages 1484–1493, 2013.

[18] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. Int. Conf. 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[19] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 5238–5246, 2017.

[20] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting in natural scenes. *arXiv: Comp. Res. Repository*, 2019.

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

[22] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proc. AAAI Conf. Artificial Intell.*, 2017.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2117–2125, 2017.

[24] Xing Linjie, Tian Zhi, Huang Weilin, and R. Scott Matthew. Convolutional Character Networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[25] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5676–5685, 2018.

[26] Yuliang Liu and Lianwen Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[27] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, 2019.

[28] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Yaqiang Wu, and Zhepeng Wang. Omnidirectional scene text detection with sequential-free box discretization. *Proc. Int. Joint Conf. Artificial Intell.*, 2019.

[29] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proc. Eur. Conf. Comp. Vis.*, pages 20–36, 2018.

[30] George G. Lorentz. *Bernstein polynomials*. American Mathematical Soc., 2013.

[31] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proc. Eur. Conf. Comp. Vis.*, pages 67–83, 2018.

[32] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition–RRC-MLT-2019. *Proc. IAPR Int. Conf. Document Analysis Recog.*, 2019.

[33] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(1):128–140, 2016.

[34] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 91–99, 2015.

[36] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11):2298–2304, 2016.

[37] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. TextNet: Irregular Text Reading from Images with an End-to-End Trainable Network. In *Proc. Asian Conf. Comp. Vis.*, pages 83–99. Springer, 2018.

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[39] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning Shape-Aware Embedding for Scene Text Detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4234–4243, 2019.

[40] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv: Comp. Res. Repository*, 2016.

[41] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape Robust Text Detection with Progressive Scale Expansion Network. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[42] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network. *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[43] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6449–6458, 2019.

[44] Feng Wei, He Wenhao, Yin Fei, Zhang Xu-Yao, and Cheng-Liu Liu. TextDragon: An end-to-end framework for arbitrary shaped text spotting. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[45] Zecheng Xie, Yaoxiong Huang, Yuanzhi Zhu, Lianwen Jin, Yuliang Liu, and Lele Xie. Aggregation cross-entropy for sequence recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2019.

[46] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Trans. Image Process.*, 2019.

[47] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[48] Zhuoyao Zhong, Lianwen Jin, Shuye Zhang, and Ziyong Feng. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv: Comp. Res. Repository*, 2016.

[49] Zhuoyao Zhong, Lei Sun, and Qiang Huo. An anchor-free region proposal network for faster r-cnn-based text detection approaches. *Int. J. Document Analysis Recogn.*, 22(3):315–327, 2019.