All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting

Hao Wang,[1*] Pu Lu,[1*] Hui Zhang,[1*] Mingkun Yang,[1] Xiang Bai,[1†]
Yongchao Xu,[1] Mengchao He,[2] Yongpan Wang,[2] Wenyu Liu[1†]
[1]Huazhong University of Science and Technology, [2]Alibaba Group
{wanghao4659, lupu, huizhang0110, yangmingkun, xbai, yongchaoxu, liuwy}@hust.edu.cn
mengchao.hmc@alibaba-inc.com, yongpan@taobao.com

Abstract

Recently, end-to-end text spotting that aims to detect and recognize text from cluttered images simultaneously has received particularly growing interest in computer vision. Different from the existing approaches that formulate text detection as bounding box extraction or instance segmentation, we localize a set of points on the boundary of each text instance. With the representation of such boundary points, we establish a simple yet effective scheme for end-to-end text spotting, which can read the text of arbitrary shapes. Experiments on three challenging datasets, including ICDAR2015, Total-Text and COCO-Text demonstrate that the proposed method consistently surpasses the state-of-the-art in both scene text detection and end-to-end text recognition tasks.

Introduction

Automatic reading text from natural images has attracted great attention due to its wide practical applications such as office automation, network content security, intelligent transportation system (Zhu et al. 2018; Rong, Yi, and Tian 2016), geo-location, and visual search (Bai et al. 2018).

In the past decade, scene text detection and recognition are extensively studied as two separated sub-tasks of a reading system, but in fact, text detection and recognition are highly relevant and complementary to each other. This assumption is confirmed by the recent end-to-end text spotting methods (Jaderberg et al. 2016; Liao et al. 2017; Liu et al. 2018; Li, Wang, and Shen 2017; He et al. 2018; Busta et al. 2017; Lyu et al. 2018a) that combine the detection and recognition stages with an end-to-end trainable neural network. These spotting methods follow a similar pipeline. First, the horizontal/oriented bounding box of each text instance is detected. Then, the image patches or CNN features inside the detected bounding boxes are cropped and fed to a sequence recognition model. Benefiting from feature sharing and joint optimization, the performances of detection and end-to-end recognition can be enhanced at the same time.

Figure 1: Illustrations of two kinds of methods for text region representation. (a) An oriented rectangle box is used to represent the text region and cropped as (b); (c) A set of boundary points are used to represent the text region, and it can be transformed into a horizontal region like (d).

Despite promising progress, most existing spotting methods (Liao et al. 2017; Liu et al. 2018; Li, Wang, and Shen 2017; He et al. 2018; Busta et al. 2017; Jaderberg et al. 2016) suffer from dealing with text of irregular shapes, such as curve text. For a general end-to-end OCR system, it is inevitable to handle the text with arbitrary shapes, as curve text and other types of irregular text are very common in our real-world. In (Liu et al. 2018; Li, Wang, and Shen 2017; He et al. 2018; Busta et al. 2017; Liu and Jin 2017; He et al. 2017), the detected bounding box of each text instance is represented with a rectangle, which only can tightly cover straight text instances. Rectangular boxes have high limitations in describing irregular text for an end-to-end text spotter since it more or less contains background information which brings difficulties to the text recognition stage, as shown in Fig. 1(a). Recently, an end-to-end OCR model (Lyu et al. 2018a) for spotting arbitrary-shaped text is presented based on Mask RCNN, which tackles text detection and recognition via instance segmentation and achieved state-of-the-art result. However, this method needs extra character-level annotations for training, and its processes of instance segmentation brings more computational burden.

In this paper, we propose an end-to-end trainable network for spotting arbitrary-shaped text without character-level annotations. Instead of detecting a rectangle bounding box, our detection is performed by localizing the boundary of a text

[*]Authors contribute equally.
[†]Corresponding authors.

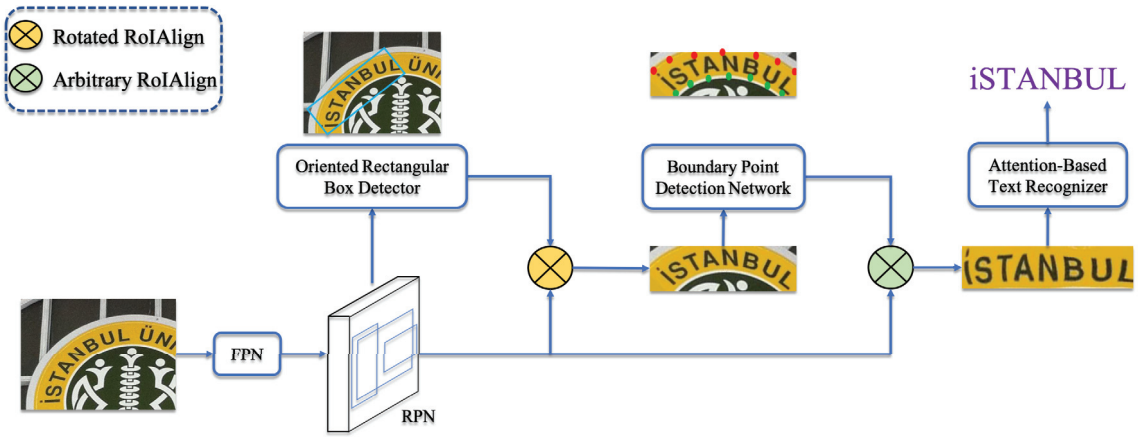

Figure 2: An overview of the proposed method. The Oriented Rectangular Box Detector is used to predict oriented rectangular box. The Boundary Point Detection Network inputs each rotated Region of Interest (RoI) feature and predicts a set of boundary points. Finally, the Arbitrary RoIAlign is used to get rectified features for attention-based text recognizer to predict text labels. Note that, for all figures, we use the input image for illustration, but operations are actually conducted on feature maps.

instance. More specifically, the aim of our detection is to predict a set of boundary points, which are more flexible for describing various shapes of scene text, often embodied in two dimensional space, as shown in Fig. 1(c). The usage of boundary points has three advantages for building an end-to-end OCR system: 1) CNN features of irregular text region can be accurately acquired with boundary points, resulting in effectively eliminating the disturb of background noise to the subsequent recognition; 2) With boundary points, irregular text can be easily transformed or rectified into a regular one (i.e. horizontal text), as described in Fig. 1(d), which is realistic input for a sequence recognition model. Similar to a recent model for irregular text recognition (Shi et al. 2019), such a transformation operation can be simply implemented and differentiable in CNN; 3) The position of boundary points can be easily refined through back propagation when training a recognition model, fully enjoying the improvement of detection performance from recognition stage. Therefore, boundary points appear to be a reasonable representation that can smoothly and effectively bridge text detection and recognition modules.

However, directly detecting boundary points of text is challenging due to the diversity of text shape and scale. To effectively extract text boundary points, we adopt a coarse-to-fine strategy: First, the minimum oriented rectangular box of each text instance is detected with a two-stage CNN detector, as shown in in Fig. 1(a); Then, the boundary point prediction is performed in the oriented rectangular box. Our experiments have validated the effectiveness of the proposed boundary point detection. Additionally, benefiting from the representation of boundary points, the proposed method achieves state-of-the-art performance in both text detection and text spotting on several benchmark datasets.

The contributions in this work are two-fold: 1) We recommend the representation of boundary points for end-to-end text spotting, which is more suitable than a rectangular box or segmentation mask for connecting detection and recognition modules. 2) We design a novel end-to-end trainable network for joint optimizing boundary point detection and text recognition, which can read both straight and curve text.

Related Work

Scene text reading has attracted great attention in computer vision. Plenty of excellent works have appeared in the past decade. Due to the page limit, we can not detail the works about scene text detection and recognition which are referred to (Long, He, and Yao 2018; Zhu, Yao, and Bai 2016). Here, related works about scene text spotting are introduced.

Methods on text spotting could be roughly divided into two categories according to the representation of text region: rectangular box based methods and segmentation based methods. In the first category, previous methods (Liao et al. 2017; Jaderberg et al. 2016) train text detector and recognizer separately. And image patches are cropped based on rectangular boxes for recognition. The separate pipeline results in unsatisfactory performance on both tasks, since the relationship within them is ignored. (Liu et al. 2018; Li, Wang, and Shen 2017; He et al. 2018; Busta et al. 2017) share a common idea: text region features are extracted for subsequent recognizer. In (Li, Wang, and Shen 2017), text region is formulated as horizontal rectangular box, which can only support horizontal text instance. Methods in (Liu et al. 2018; He et al. 2018; Busta et al. 2017; Sun et al. 2018) extract text features in the minimum oriented rectangular box, which can tackle oriented text. But those works suffer from handling curved text. As the only segmentation based method (Lyu et al. 2018a), each text instance and corresponding characters are segmented. Although curved text could be represented by the segmentation map, extra character-level annotations are required. In addition, the trained model ignores contextual information between characters of a word, which will affect the recognition performance.

In order to read arbitrary-shaped text in an end-to-end

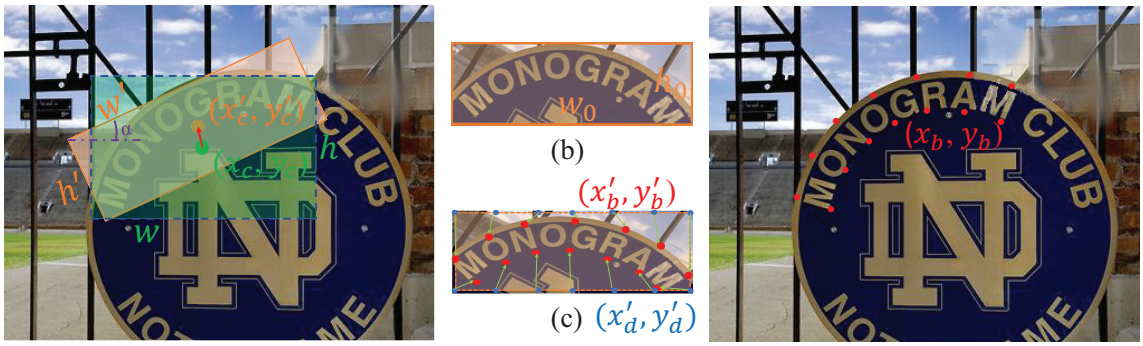

Figure 3: (a) Regression from the axis-aligned box of RPN to the oriented rectangular box. (b) The minimum rectangular box. (c) Regression from a set of default points which are uniformly distributed at the upper and lower sides of the minimum rectangular box to boundary points. (d) Aligning boundary points to the original image.

manner, a set of boundary points are proposed to describe text regions. Although boundary points are also used for text region representation in (Zhang et al. 2019; Zhu and Du 2018), a complex post processing is required to generate them, which is not differentiable for training. In spite of predicting boundary points directly without post processing in (Wang et al. 2019), a recurrent neural network based network is time consuming for end-to-end task. Compared to (Liu et al. 2018; Li, Wang, and Shen 2017; He et al. 2018), the novel representation is more flexible and accurate to represent text with arbitrary shapes than rectangle box. Compared to (Lyu et al. 2018a), such a description is free from character-level annotations. More details about the proposed method will be introduced in the following sections.

Methodology

As illustrated in Fig. 2, our pipeline is composed of three parts: the oriented rectangular box detector, the boundary point detection network, and the recognition network. As for the oriented rectangular box detector, we first apply a RPN (Ren et al. 2015), where the backbone is FPN (Lin et al. 2017) equipped with ResNet-50 (He et al. 2016), to generate horizontal text proposals. Then, an oriented rectangular box of each proposal is generated via predicting its center point, height, width, and orientation. Next, the boundary points of each oriented rectangular box are regressed by the boundary point detection network (BPDN). Finally, with the predicted boundary points, the feature maps are rectified as regular ones for the subsequent text recognizer.

As shown in Fig. 3(a), the boundary points can be predicted for each horizontal proposal by BPDN, but we observe that BPDN suffers from text instances of various directions and shapes since such cases contain more background noise and have stronger deformation. To alleviate this effect, an oriented rectangular box of each proposal is predicted, with which the feature maps are transformed into horizontal ones via RotatedRoIAlign (Huang et al. 2018), as illustrated in Fig. 3(b). Concretely, we follow the method proposed in (Ma et al. 2018) to obtain the oriented rectangular box by predicting its center point, height, width, and orientation. Here, a module composed of three stacked fully connected layers is designed. More details about detecting the oriented rectangle are referred to Fig. 3(a).

Boundary Point Detection Network

BPDN consists of four stacked 3×3 convolutional layers and one fully connected layer. Inspired by RPN where proposals are regressed based on default anchors, a set of default points are predefined for boundary points to refer, as shown in Fig. 3(c). Specifically, K points are equidistantly sampled on each long side of text instance as target boundary points. And corresponding default points are evenly placed along long sides of the minimum rectangular box. Instead of directly predicting the coordinates of boundary points, offsets to its associated default points are first generated. The module predicts a $4K$-d vector which is coordinate offsets (2-d) of $2K$ boundary points. Given the coordinate offsets ($\Delta x'$, $\Delta y'$), the boundary point (x'_b, y'_b) can be obtained from

$$\begin{aligned} x'_b &= x'_d + w_0 \Delta x' \\ y'_b &= y'_d + h_0 \Delta y', \end{aligned} \tag{1}$$

where (x'_d, y'_d) represents default point. w_0 and h_0 are the width and height of the minimum rectangular box.

To be consistent with original features, we align the boundary points (x'_b, y'_b) in transformed horizontal feature maps to the original ones (x_b, y_b) using

$$\begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} = M^{-1} \begin{bmatrix} x'_b \\ y'_b \\ 1 \end{bmatrix}, \tag{2}$$

$$M = \begin{bmatrix} s_w cos\alpha & -s_h sin\alpha & -s_w x'_c cos\alpha + s_h y'_c sin\alpha \\ s_w sin\alpha & s_h cos\alpha & -s_w x'_c sin\alpha - s_h y'_c cos\alpha \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

(x'_c, y'_c) is the center point of the oriented rectangle. s_w and s_h equal to w_o/w' and h_o/h' respectively. And α is the angle from the positive direction of the x-axis to the direction parallel to the long side of the oriented rectangle.

	Type	Configurations [size,stride,padding]	Out Channels
Encoder	conv_bn_relu	[3,1,1]	256
	max-pool	[2,1,0]	256
	conv_bn_relu	[3,1,1]	256
	max-pool	[2,1,(0,1)]	256
	conv_bn_relu	[3,1,1]	256
	max-pool	[2,1,(0,1)]	256
Decoder	Att. GRU		256
	FC		$\|S\|$

Table 1: The architecture of the recognition branch, which consists of three stacked convolutional layers, "Att. GRU" which stands for attentional GRU decoder and a fully-connection layer. $|S|$ represents the number of decoded characters. We set the number of decoded characters in our experiments to 63, which corresponds to digits (0-9), English characters (a-z/A-Z), and an end-of-sequence symbol.

Following (Shi et al. 2019), ArbitraryRoIAlign is adopted to flatten features of text instances with arbitrary shapes. Specifically, given the boundary points of each text instance, Thin-Plate-Spline transformation (Bookstein 1989) is adopted to rectify the features to regular ones.

Recognition Network

CRNN (Shi, Bai, and Yao 2017) is the first method to treat text recognition as a sequence-to-sequence problem by combining CNN and RNN in an end-to-end network. In some latest works (Zhan and Lu 2019; Yang et al. 2019; Luo, Jin, and Sun 2019), recognition network is a common attentional sequence-to-sequence network. The recognizer predicts a character sequence from the rectified features. The architecture of the recognition branch is given in Tab. 1. Firstly, the rectified features are fed into encoder to extract higher-level feature sequence $F \in \Re^{n \times C}$. Then the attention-based decoder is adopted to translate F into a symbol sequence $y = (y_1, ..., y_T)$, where T is the length of the label sequence. At step t, the decoder predicts a character based on the encoder output F, the internal state s_{t-1} and the result y_{t-1} predicted in the last step. In the current step, the decoder starts by computing a vector of attention weights, α_t, through its attention mechanism. Then, the weighted feature g_t is calculated according to

$$\begin{aligned} g_t &= \sum_{i=1}^{n} \alpha_{t,i} F_i \\ \alpha_{t,i} &= exp(e_{t,i}) / \sum_{j=1}^{n} exp(e_{t,j}) \\ e_{t,j} &= w^T tanh(W s_{t-1} + V F_i + b), \end{aligned} \tag{4}$$

where w, W, V and b are trainable weights.

Taking s_{t-1}, g_t and y_{t-1} as inputs, the RNN calculates an output vector x_t and a new state vector s_t via

$$(x_t, s_t) = RNN(s_{t-1}, (g_t, onehot(y_{t-1}))), \tag{5}$$

where $(g_t, onehot(y_{t-1}))$ is the concatenation of g_t and the one-hot embedding of y_{t-1}. In our method, a GRU is used as RNN unit. Finally, a distribution of the current-step symbol is predicted through

$$p(y_t) = softmax(W_o x_t + b_o), \tag{6}$$

where W_o and b_o are learnable parameters.

Loss Functions

The objective function consists of four parts, which is defined as follows,

$$L = L_{rpn} + L_{or} + L_{bp} + L_{recog}, \tag{7}$$

where L_{rpn} is the loss of RPN, which is identical as in (Ren et al. 2015). L_{or} is the loss that regression from axis-aligned rectangular proposal to oriented rectangular box, which is similar as (Ma et al. 2018). Since the above losses are not our main contributions, we do not detail them here due to the page limit. L_{bp} is the loss of boundary point regression which is calculated as Smoothed-L1 loss. The loss function can be formulated as

$$\begin{aligned} L_{bp} = \frac{1}{2K} \sum_{i=1}^{2K} \big(& Smooth_{L1}(\hat{x}'_{b,i}, x'_{b,i}) + \\ & Smooth_{L1}(\hat{y}'_{b,i}, y'_{b,i}) \big), \end{aligned} \tag{8}$$

where $(x'_{b,i}, y'_{b,i})$ is the i-th predicted boundary point, whose associated target boundary point is $(\hat{x}'_{b,i}, \hat{y}'_{b,i})$.

In the recognition network, the recognition loss can be formulated as

$$L_{recog} = -\frac{1}{T} \sum_{t=1}^{T} \log p(y_t). \tag{9}$$

Experiments

To confirm the effectiveness of the proposed method on arbitrary-shaped text spotting, we conduct exhaustive experiments and compare with other state-of-the-art methods on four popular benchmarks which consist of a horizontal text set ICDAR2013 (Karatzas et al. 2013), two oriented text sets ICDAR2015 (Karatzas et al. 2015) and COCO-Text (Veit et al. 2016), a curved text set TotalText (Ch'ng and Liu 2019). The details about these datasets are as follows.

Datasets

SynthText (Gupta, Vedaldi, and Zisserman 2016) has about 800,000 images, which are generated via synthesizing engine. Text instances within the images are multi-oriented, whose annotations consist of word-level and character-level oriented bounding boxes, as well as text sequences.

TotalText contains horizontal, multi-oriented, and curved text in images. The dataset contains 1,255 training images and 300 test images. All images are annotated with polygons and transcriptions in word-level.

ICDAR2015 focuses on multi-oriented scene text detection and recognition in natural images. There are 1,000 training images and 500 test images. Word-level quadrangles and transcriptions of each image are given.

ICDAR2013 is a dataset which focuses on the horizontal scene text detection and recognition in natural images. The

Algorithm 1 Generate Target Boundary Points

Require:
Points on each long side: $P = \{p_0, p_1, ..., p_{N-1}\}$.
The excepted number of sampled points: K.
Ensure:
The generated target boundary points:
$Q = \{q_0, q_1, ..., q_{K-1}\}$.

$distance[0]$ =0; $len[0] = 0$;
for i=0; i < N-1; i++ **do**
  // calculate distance between two adjacent points
  $distance[i+1]$ = distance(p_i, p_{i+1});
end for
for j=1; j<N; j++ **do**
  $len[j] = len[j-1] + distance[j]$;
end for
$average\_distance = \frac{1}{K-1} * \sum distance[i]$;
for i=0; i < K; i++ **do**
  $cur\_pos = average\_distance * i$;
  for j=0; j < N-1; j++ **do**
    if $len[j] \leqslant cur\_pos < len[j+1]$ **then**
      $q_i = \frac{(p_{j+1}-p_j)*(cur\_pos-len[j])}{len[j+1]-len[j]} + p_j$;
    end if
  end for
end for

dataset consists of 229 images in the training set and 233 images in the test set. Similar to SynthText, the bounding box and the transcription for each word-level and character-level text instance are also provided.

COCO-Text contains 63,686 images. Though it is evaluated with axis-aligned bounding boxes, the text instances in the images are distributed in various orientations. Owing to no lexicon provided in the evaluation, the text must be recognized without any prior knowledge.

Implementation Details

Different from previous text spotting methods (Liu et al. 2018; Li, Wang, and Shen 2017) which use alternating training strategy, our boundary point detector and text recognizer could be trained in an end-to-end manner. The whole training process contains two steps: first, we pretrain the network on SynthText, then real data is adopted to finetune the model.

During pretraining, the mini-batch is set to 16, and the longer sides of input images are resized to 800 while keeping the aspect ratio. The maximum number of proposals in each image on the recognition branch is set to 16. In the finetuning stage, for the data augmentation, we randomly crop a patch whose edges range from 210 to 1100 while keeping all the text instance not cropped and resize the patch to (640, 640). Finally, the resized patch is randomly rotated 90° with a probability of 0.2. We collect the training images from ICDAR2013, ICDAR2015, and TotalText to finetune the model with the mini-batch set to 16. We optimize our model using SGD with a weight decay of 0.0001 and momentum of 0.9. We train our model for 270k iterations for pretraining, with an initial learning rate of 0.01, and decayed to a tenth at the 100k and the 200k iteration. In the finetuning stage, the initial learning rate is set to 0.001 and then is decreased to 0.0001 and 0.00001 at the 80k and 120k iteration. The finetuning process is terminated at the 140k iteration. We implement our method in Pytorch and conduct all experiments on a regular workstation with Nvidia Titan Xp GPUs. The model is trained in parallel and evaluated on a single GPU.

Label Generation During training, we need equidistantly spaced boundary points to train BPDN. However, only corner points are given in the groundtruth. So we need to sample points on the longer sides of text boundary using Algorithm 1. In our experiments, K is set to 7.

Method	Detection			E2E	
	P	R	F	None	Full
TotalText (Ch'ng and Liu 2019)	40.0	33.0	36.0	-	-
TextBoxes (Liao et al. 2017)	62.1	45.5	52.5	36.3	48.9
MaskTextSpotter (Lyu et al. 2018a)	87.0	80.2	83.4	52.9	71.8
Boundary (det only)	85.2	83.5	84.3	-	-
Boundary (end-to-end)	**88.9**	**85.0**	**87.0**	**65.0**	**76.1**

Table 2: Results on TotalText. "P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "E2E" means end-to-end, "None" means recognition without any lexicon, "Full" lexicon contains all words in test set. Following tables follow the same usage.

Curved Text

The proposed method focuses on arbitrary-shaped text spotting. To verify its effectiveness, we first conduct experiments on TotalText. During testing, the longer sides of images are resized to 1,100. For a fair comparison, we follow the evaluation protocols in the latest method (Lyu et al. 2018a).

The performance on TotalText is given in Tab. 2. Our method achieves state-of-the-art performance on both detection and end-to-end recognition. Specifically, the proposed method outperforms MaskTextSpotter with improvements of 3.6% and 12.1% respectively on detection and end-to-end recognition without lexicon. The improvement over other methods gives credit to the following four points: 1) Compared with MaskTextSpotter, the attention-based decoder in our recognition module can capture the relationship between characters of a word, which is helpful for the recognition task. However, MaskTextSpotter predicts the characters separately, ignoring the context within them. 2) Compared with other methods, before being fed into the text recognizer, text with arbitrary shapes is rectified to a regular one, which attenuates text irregularities and therefore decreases the recognition difficulty. 3) Due to the better recognition results, the detection results could be implicitly improved through the shared backbone features. 4) Using boundary points to describe the shapes of text instances is more flexible and efficient to locate the text instances.

Method	ICDAR2015						ICDAR2013					
	Detection			E2E			Detection			E2E		
	P	R	F	S	W	G	P	R	F	S	W	G
DeepTextSpotter (Busta et al. 2017)	-	-	-	54.0	51.0	47.0	-	-	-	89.0	86.0	77.0
TextBoxes++ (Liao, Shi, and Bai 2018)	87.2	76.7	81.7	73.3	65.9	51.9	88.0	74.0	81.0	93.0	92.0	85.0
He* *et al*. (He et al. 2018)	87.0	86.0	87.0	**82.0**	**77.0**	63.0	91.0	**89.0**	90.0	91.0	89.0	86.0
FOTS* (Liu et al. 2018)	91.0	85.2	88.0	81.1	75.9	60.8	-	-	88.3	88.8	87.1	80.8
MaskTextSpotter (Lyu et al. 2018a)	**91.6**	81.0	86.0	79.3	73.0	62.4	**95.0**	88.6	**91.7**	**92.2**	**91.1**	**86.5**
Boundary (det only)	88.1	82.2	85.0	-	-	-	89.3	85.2	87.2	-	-	-
Boundary (end-to-end)	89.8	**87.5**	**88.6**	79.7	75.2	**64.1**	93.1	87.3	90.1	88.2	87.7	84.1

Table 3: Results on ICDAR2015 and ICDAR2013 (DetEval). "S", "W" and "G" mean recognition with strong, weak and generic lexicon respectively. "*" denotes that training dataset of MLT2017 is used for training. Following tables follow the same usage.

Method	Detection			E2E		
	P	R	F	P	R	F
Baseline A** (Veit et al. 2016)	**83.8**	23.3	36.5	**68.4**	28.3	40.0
Baseline B** (Veit et al. 2016)	**59.7**	10.7	19.1	9.97	**54.5**	16.9
Baseline C** (Veit et al. 2016)	18.6	4.7	7.5	1.7	4.2	2.4
DeepTextSpotter (Busta et al. 2017)	-	-	-	31.4	16.8	21.9
Boundary (end-to-end)	59.0	**67.7**	**63.0**	55.7	32.8	**41.3**

Table 4: Results on COCO-Text. Lexicon is not used for end-to-end testing. "MS" means testing with multiple scales. Methods with "**" are evaluated using V1.1 annotations.

Oriented Text

We also conduct experiments on ICDAR2015 to confirm the superiority of the proposed method on the oriented scene text. Images are resized to 1080×1920 before being fed into the framework. As shown in Tab. 3, our method slightly outperforms previous methods in detection and end-to-end recognition with a general lexicon by 0.6% and 1.1% respectively. However, besides SynthText, extra 9,000 images in MLT2017 are used for training by (Liu et al. 2018; He et al. 2018). For a fair comparison, we follow the same settings with MaskTextSpotter, in which the images of MLT2017 are not used. We can observe that our method gets respectively 2.6% and 1.7% improvements in detection and end-to-end tasks with general lexicon.

Horizontal Text

Besides promising performances have been achieved on curved and oriented benchmarks, we also evaluate the proposed method on horizontal scene text. The longer sides of input images are resized to 1280 while keeping the aspect ratio of the images. As shown in Tab. 3, our method gets comparable performance on both tasks. However, character-level annotations are required for MaskTextSpotter. Additionly, extra 9,000 images from MLT2017 are added to improve the performance by FOTS and (He et al. 2018).

Datasets	Proposal	Detection			E2E
		P	R	F	None
ICDAR2015	axis-aligned	88.2	**87.5**	87.8	63.1
	oriented	**89.8**	**87.5**	**88.6**	**64.1**
TotalText	axis-aligned	87.3	84.3	85.8	63.8
	oriented	**88.9**	**85.0**	**87.0**	**65.0**

Table 5: Comparison of axis-aligned proposal and oriented rectangular proposal. "axis-aligned" means that axis-aligned proposal is used to predict boundary points, "oriented" means that oriented rectangular proposal is used.

Generalization Evaluation

We evaluate the generalization of our method on COCO-Text. Following (Lyu et al. 2018b), our model is not trained with the training set of COCO-Text. The detection task in Tab. 4 is evaluated with the annotations V1.4 for a fair comparison with previous methods, while the end-to-end task is evaluated with the newest annotations V2.0. The longer sides of input images are resized to 1280 while keeping the aspect ratio of images. As shown in Tab. 4, our method achieves state-of-the-art performance on both tasks, which confirms that our method has stronger generalization ability.

Visualization

The results of several images containing text instances with arbitrary shapes are illustrated Fig. 4. With the novel representation of text regions, the proposed system can read texts of arbitrary shape. Even for some vertical text instances (pictures at bottom left and bottom right of Fig. 4), our method can successfully localize and recognize them. Those challenging samples confirm the superiority and robustness of the proposed boundary points.

Although our method can achieve promising performances, there are still some failure cases, as shown in Fig. 5. We observe that it is difficult to correctly spot texts composed of rare art fonts, since there are few samples in the training set. Besides, our method also struggles to detect or recognize extremely tiny or long texts and blurred texts.

Ablation Study

Oriented Rectangular Box Detector As we mentioned before, the oriented rectangular box detector plays an important role in our whole pipeline. It provides better features

Figure 4: Examples of text spotting results of our method on Total-Text, ICDAR2015, and ICDAR2013.

Figure 5: Some failure cases produced by our method.

through RotatedRoIAlign for boundary point regression. If we do not predict the oriented rectangular box and use the axis-aligned proposal from RPN, the boundary of text instance in the proposal changes dramatically, and it's hard to predict the precise location by regression.

As shown in Tab. 5, using oriented rectangular proposal obtained from oriented rectangular box detector improves the performance of the detection and end-to-end recognition stably compared to the axis-aligned proposal. On ICDAR2015, the performance respectively improves by 0.8% and 1.0% in the detection and end-to-end tasks. On Total-Text, the oriented rectangular proposal provides significant improvements of 1.2% and 1.2%. These results show that the oriented rectangular proposal can reduce the difficulty of boundary point prediction and make it more precise.

Conclusion

In this paper, we present an end-to-end trainable network that defines text of arbitrary shape as a set of boundary points. Our method has achieved the state-of-the-art in both tasks of scene text detection and end-to-end text recognition on the standard benchmarks including oriented text and curved text, which confirms its robustness and effectiveness in reading scene text. The flexible and accurate representation of boundary points is potential to become the mainstream description for scene text spotting. Besides, both text detection and recognition tasks benefit from boundary point representation. In the future, we would like to improve the efficiency of detecting boundary points.

Acknowledgements

This work was supported by National Natural Science Foundation of China (61733007, 61572207) and the Alibaba Innovative Research Program.

References

Bai, X.; Yang, M.; Lyu, P.; Xu, Y.; and Luo, J. 2018. Integrating scene text and visual appearance for fine-grained image classification. *IEEE Access* 6:66322–66335.

Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI* 11(6):567–585.

Busta, M.; Neumann, L.; ; and Matas, J. 2017. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *ICCV 2017*, 2223–2231.

Ch'ng, C.-K., and Liu, C.-L. 2019. Total-text: toward orientation robustness in scene text detection. *IJDAR* 1–22.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *CVPR*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, W.; Zhang, X.-Y.; Yin, F.; and Liu, C.-L. 2017. Deep direct regression for multi-oriented scene text detection. In *ICCV*, 745–753.

He, T.; Tian, Z.; Huang, W.; Shen, C.; Qiao, Y.; and Sun, C. 2018. An end-to-end textspotter with explicit alignment and attention. In *CVPR*, 5020–5029.

Huang, J.; Sivakumar, V.; Mnatsakanyan, M.; and Pang, G. 2018. Improving rotated text detection with rotation region proposal networks. *CoRR* abs/1811.07031.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *IJCV* 116(1):1–20.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazán, J.; and de las Heras, L. 2013. ICDAR 2013 robust reading competition. In *ICDAR*, 1484–1493.

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.; and Valveny, E. 2015. ICDAR 2015 competition on robust reading. In *ICDAR*, 1156–1160.

Li, H.; Wang, P.; and Shen, C. 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. In *ICCV*, 5238–5246.

Liao, M.; Shi, B.; Bai, X.; Wang, X.; and Liu, W. 2017. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 4161–4167.

Liao, M.; Shi, B.; and Bai, X. 2018. Textboxes++: A single-shot oriented scene text detector. *TIP* 27(8):3676–3690.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Liu, Y., and Jin, L. 2017. Deep matching prior network: Toward tighter multi-oriented text detection. In *CVPR*, 1962–1969.

Liu, X.; Liang, D.; Yan, S.; Chen, D.; Qiao, Y.; and Yan, J. 2018. Fots: Fast oriented text spotting with a unified network. In *CVPR*, 5676–5685.

Long, S.; He, X.; and Yao, C. 2018. Scene text detection and recognition: The deep learning era. *CoRR* abs/1811.04256.

Luo, C.; Jin, L.; and Sun, Z. 2019. Moran: A multi-object rectified attention network for scene text recognition. *PR* 90:109–118.

Lyu, P.; Liao, M.; Yao, C.; Wu, W.; and Bai, X. 2018a. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, 67–83.

Lyu, P.; Yao, C.; Wu, W.; Yan, S.; and Bai, X. 2018b. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, 7553–7563.

Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; and Xue, X. 2018. Arbitrary-oriented scene text detection via rotation proposals. *IEEE TMM* 20(11):3111–3122.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Rong, X.; Yi, C.; and Tian, Y. 2016. Recognizing text-based traffic guide panels with cascaded localization network. In *ECCV*, 109–121.

Shi, B.; Bai, X.; and Yao, C. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI* 39(11):2298–2304.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2019. Aster: An attentional scene text recognizer with flexible rectification. *TPAMI* 41(9):2035–2048.

Sun, Y.; Zhang, C.; Huang, Z.; Liu, J.; Han, J.; and Ding, E. 2018. Textnet: Irregular text reading from images with an end-to-end trainable network. In *ACCV*, 83–99.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. J. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *CoRR* abs/1601.07140.

Wang, X.; Jiang, Y.; Luo, Z.; Liu, C.; Choi, H.; and Kim, S. 2019. Arbitrary shape scene text detection with adaptive text region representation. In *CVPR*, 6449–6458.

Yang, M.; Guan, Y.; Liao, M.; He, X.; Bian, K.; Bai, S.; Yao, C.; and Bai, X. 2019. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, 9147–9156.

Zhan, F., and Lu, S. 2019. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, 2059–2068.

Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding, E.; and Ding, X. 2019. Look more than once: An accurate detector for text of arbitrary shapes. In *CVPR*, 10552–10561.

Zhu, Y., and Du, J. 2018. Sliding line point regression for shape robust scene text detection. In *ICPR*, 3735–3740.

Zhu, Y.; Liao, M.; Yang, M.; and Liu, W. 2018. Cascaded segmentation-detection networks for text-based traffic sign detection. *IEEE TITS* 19:209–219.

Zhu, Y.; Yao, C.; and Bai, X. 2016. Scene text detection and recognition: Recent advances and future trends. *FCS* 10(1):19–36.