Real-Time Scene Text Detection with Differentiable Binarization

Article *in* Proceedings of the AAAI Conference on Artificial Intelligence · April 2020

DOI: 10.1609/aaai.v34i07.6812

Minghui Liao
Huawei Technologies
38 PUBLICATIONS **5,846** CITATIONS
SEE PROFILE

Yan Wan
Donghua University
40 PUBLICATIONS **1,447** CITATIONS
SEE PROFILE

Cong Yao
100 PUBLICATIONS **16,162** CITATIONS
SEE PROFILE

Xiang Bai
Huazhong University of Science and Technology
436 PUBLICATIONS **43,496** CITATIONS
SEE PROFILE

Real-time Scene Text Detection with Differentiable Binarization

Minghui Liao[1*], **Zhaoyi Wan**[2*], **Cong Yao**[2], **Kai Chen**[3,4], **Xiang Bai**[1†]
[1]Huazhong University of Science and Technology, [2]Megvii, [3]Shanghai Jiao Tong University, [4]Onlyou Tech.
{mhliao,xbai}@hust.edu.cn, i@wanzy.me, yaocong2010@gmail.com, kchen@sjtu.edu.cn

Abstract

Recently, segmentation-based methods are quite popular in scene text detection, as the segmentation results can more accurately describe scene text of various shapes such as curve text. However, the post-processing of binarization is essential for segmentation-based detection, which converts probability maps produced by a segmentation method into bounding boxes/regions of text. In this paper, we propose a module named Differentiable Binarization (DB), which can perform the binarization process in a segmentation network. Optimized along with a DB module, a segmentation network can adaptively set the thresholds for binarization, which not only simplifies the post-processing but also enhances the performance of text detection. Based on a simple segmentation network, we validate the performance improvements of DB on five benchmark datasets, which consistently achieves state-of-the-art results, in terms of both detection accuracy and speed. In particular, with a light-weight backbone, the performance improvements by DB are significant so that we can look for an ideal tradeoff between detection accuracy and efficiency. Specifically, with a backbone of ResNet-18, our detector achieves an F-measure of 82.8, running at 62 FPS, on the MSRA-TD500 dataset. Code is available at: https://github.com/MhLiao/DB.

Introduction

In recent years, reading text in scene images has become an active research area, due to its wide practical applications such as image/video understanding, visual search, automatic driving, and blind auxiliary.

As a key component of scene text reading, scene text detection that aims to localize the bounding box or region of each text instance is still a challenging task, since scene text is often with various scales and shapes, including horizontal, multi-oriented and curved text. Segmentation-based scene text detection has attracted a lot of attention recently, as it can describe the text of various shapes, benefiting from its prediction results at the pixel-level. However, most segmentation-based methods require complex post-processing for grouping the pixel-level prediction results into detected text instances, resulting in a considerable time cost in the inference procedure. Take two recent state-of-the-art methods for scene text detection as examples: PSENet (Wang et al. 2019a) proposed the post-processing of progressive scale expansion for improving the detection accuracies; Pixel embedding in (Tian et al. 2019) is used for clustering the pixels based on the segmentation results, which has to calculate the feature distances among pixels.

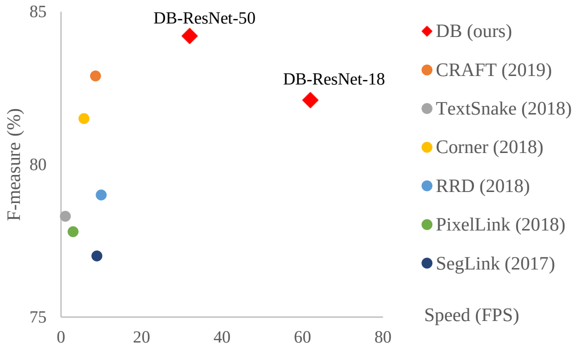

Figure 1: The comparisons of several recent scene text detection methods on the MSRA-TD500 dataset, in terms of both accuracy and speed. Our method achieves the ideal tradeoff between effectiveness and efficiency.

Most existing detection methods use the similar post-processing pipeline as shown in Fig. 2 (following the blue arrows): Firstly, they set a fixed threshold for converting the probability map produced by a segmentation network into a binary image; Then, some heuristic techniques like pixel clustering are used for grouping pixels into text instances. Alternatively, our pipeline (following the red arrows in Fig. 2) aims to insert the binarization operation into a segmentation network for joint optimization. In this manner, the threshold value at every place of an image can be adaptively predicted, which can fully distinguish the pixels from the foreground and background. However, the standard binarization function is not differentiable, we instead present an approximate function for binarization called Differentiable Binarization (DB), which is fully differentiable when train-

[*]Authors contribute equally.

[†]Corresponding author

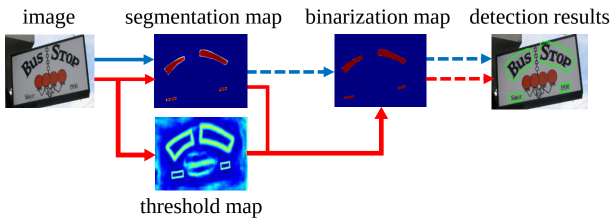

Figure 2: Traditional pipeline (blue flow) and our pipeline (red flow). Dashed arrows are the inference only operators; solid arrows indicate differentiable operators in both training and inference.

ing it along with a segmentation network.

The major contribution in this paper is the proposed DB module that is differentiable, which makes the process of binarization end-to-end trainable in a CNN. By combining a simple network for semantic segmentation and the proposed DB module, we proposed a robust and fast scene text detector. Observed from the performance evaluation of using the DB module, we discover that our detector has several prominent advantages over the previous state-of-the-art segmentation-based approaches.

1. Our method achieves consistently better performances on five benchmark datasets of scene text, including horizontal, multi-oriented and curved text.
2. Our method performs much faster than the previous leading methods, as DB can provide a highly robust binarization map, significantly simplifying the post-processing.
3. DB works quite well when using a light-weight backbone, which significantly enhances the detection performance with the backbone of ResNet-18.
4. As DB can be removed in the inference stage without sacrificing the performance, there is no extra memory/time cost for testing.

Related Work

Recent scene text detection methods can be roughly classified into two categories: Regression-based methods and segmentation-based methods.

Regression-based methods are a series of models which directly regress the bounding boxes of the text instances. TextBoxes (Liao et al. 2017) modified the anchors and the scale of the convolutional kernels based on SSD (Liu et al. 2016) for text detection. TextBoxes++ (Liao, Shi, and Bai 2018) and DMPNet (Liu and Jin 2017) applied quadrilaterals regression to detect multi-oriented text. SSTD (He et al. 2017a) proposed an attention mechanism to roughly identifies text regions. RRD (Liao et al. 2018) decoupled the classification and regression by using rotation-invariant features for classification and rotation-sensitive features for regression, for better effect on multi-oriented and long text instances. EAST (Zhou et al. 2017) and DeepReg (He et al. 2017b) are anchor-free methods, which applied pixel-level regression for multi-oriented text instances. SegLink (Shi, Bai, and Belongie 2017) regressed the segment bounding boxes and predicted their links, to deal with long text instances. DeRPN (Xie et al. 2019b) proposed a dimension-decomposition region proposal network to handle the scale problem in scene text detection. Regression-based methods usually enjoy simple post-processing algorithms (e.g. non-maximum suppression). However, most of them are limited to represent accurate bounding boxes for irregular shapes, such as curved shapes.

Segmentation-based methods usually combine pixel-level prediction and post-processing algorithms to get the bounding boxes. (Zhang et al. 2016) detected multi-oriented text by semantic segmentation and MSER-based algorithms. Text border is used in (Xue, Lu, and Zhan 2018) to split the text instances, Mask TextSpotter (Lyu et al. 2018a; Liao et al. 2019) detected arbitrary-shape text instances in an instance segmentation manner based on Mask R-CNN. PSENet (Wang et al. 2019a) proposed progressive scale expansion by segmenting the text instances with different scale kernel. Pixel embedding is proposed in (Tian et al. 2019) to cluster the pixels from the segmentation results. PSENet (Wang et al. 2019a) and SAE (Tian et al. 2019) proposed new post-processing algorithms for the segmentation results, resulting in lower inference speed. Instead, our method focus on improving the segmentation results by including the binarization process into the training period, without the loss of the inference speed.

Fast scene text detection methods focus on both the accuracy and the inference speed. TextBoxes (Liao et al. 2017), TextBoxes++ (Liao, Shi, and Bai 2018), SegLink (Shi, Bai, and Belongie 2017), and RRD (Liao et al. 2018) achieved fast text detection by following the detection architecture of SSD (Liu et al. 2016). EAST (Zhou et al. 2017) proposed to apply PVANet (Kim et al. 2016) to improve its speed. Most of them can not deal with text instances of irregular shapes, such as curved shape. Compared to the previous fast scene text detectors, our method not only runs faster but also can detect text instances of arbitrary shapes.

Methodology

The architecture of our proposed method is shown in Fig. 3. Firstly, the input image is fed into a feature-pyramid backbone. Secondly, the pyramid features are up-sampled to the same scale and cascaded to produce feature F. Then, feature F is used to predict both the probability map (P) and the threshold map (T). After that, the approximate binary map ($\hat{B}$) is calculated by P and F. In the training period, the supervision is applied on the probability map, the threshold map, and the approximate binary map, where the probability map and the approximate binary map share the same supervision. In the inference period, the bounding boxes can be obtained easily from the approximate binary map or the probability map by a box formulation module.

Binarization

Standard binarization Given a probability map $P \in R^{H\times W}$ produced by a segmentation network, where H and W indicate the height and width of the map, it is essential to convert it to a binary map $P \in R^{H\times W}$, where pixels with

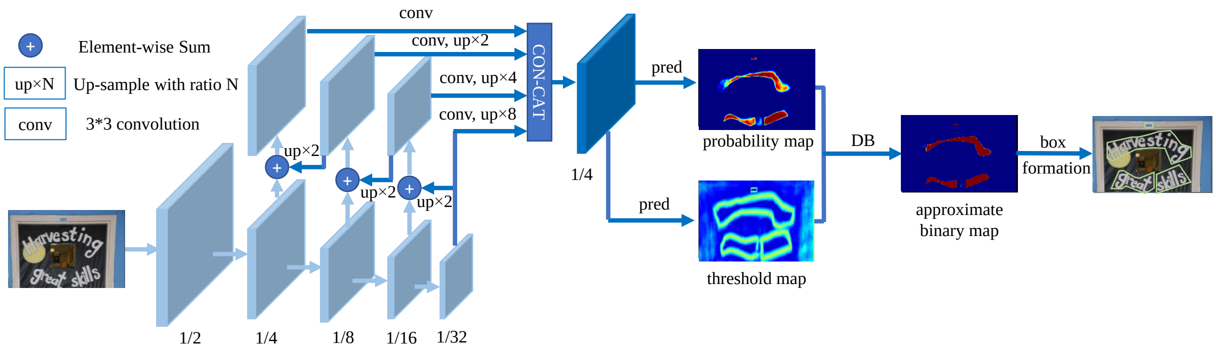

Figure 3: Architecture of our proposed method, where "pred" consists of a 3×3 convolutional operator and two de-convolutional operators with stride 2. The "1/2", "1/4", ... and "1/32" indicate the scale ratio compared to the input image.

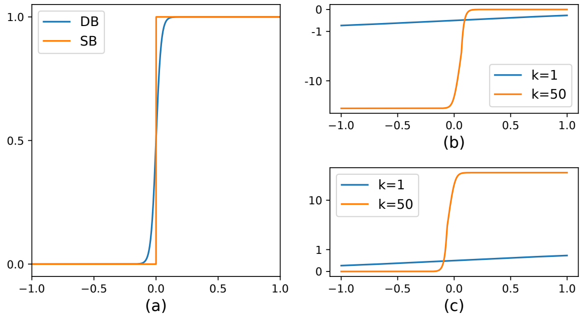

Figure 4: Illustration of differentiable binarization and its derivative. (a) Numerical comparison of standard binarization (SB) and differentiable binarization (DB). (b) Derivative of l_+. (c) Derivative of l_-.

value 1 is considered as valid text areas. Usually, this binarization process can be described as follows:

$$B_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} >= t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where t is the predefined threshold and (i, j) indicates the coordinate point in the map.

Differentiable binarization The standard binarization described in Eq. 1 is not differentiable. Thus, it can not be optimized along with the segmentation network in the training period. To solve this problem, we propose to perform binarization with an approximate step function:

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}} \tag{2}$$

where $\hat{B}$ is the approximate binary map; T is the adaptive threshold map learned from the network; k indicates the amplifying factor. k is set to 50 empirically. This approximate binarization function behaves similar to the standard binarization function (see Fig 4) but is differentiable thus can be optimized along with the segmentation network in the training period. The differentiable binarization with adaptive thresholds can not only help differentiate text regions from the background, but also separate text instances which are closely jointed. Some examples are illustrated in Fig.7.

The reasons that DB improves the performance can be explained by the backpropagation of the gradients. Lets take the binary cross-entropy loss as an example. Define $f(x) = \frac{1}{1+e^{-kx}}$ as our DB function, where $x = P_{i,j} - T_{i,j}$. Then the losses l_+ for positive labels and l_- for negative labels are:

$$\begin{aligned} l_+ &= -\log \frac{1}{1+e^{-kx}} \\ l_- &= -\log(1 - \frac{1}{1+e^{-kx}}) \end{aligned} \tag{3}$$

We can easily compute the differential of the losses with the chain rule:

$$\begin{aligned} \frac{\partial l_+}{\partial x} &= -kf(x)e^{-kx} \\ \frac{\partial l_-}{\partial x} &= kf(x) \end{aligned} \tag{4}$$

The derivatives of l_+ and l_- are also shown in Fig. 4. We can perceive from the differential that (1) The gradient is augmented by the amplifying factor k; (2) The amplification of gradient is significant for most of the wrongly predicted region ($x < 0$ for L_+; $x > 0$ for L_-), thus facilitating the optimization and helping to produce more distinctive predictions. Moreover, as $x = P_{i,j} - T_{i,j}$, the gradient of P is effected and rescaled between the foreground and the background by T.

Adaptive threshold

The threshold map in Fig. 1 is similar to the text border map in (Xue, Lu, and Zhan 2018) from appearance. However, the motivation and usage of the threshold map are different from the text border map. The threshold map with/without supervision is visualized in Fig. 6. The threshold map would highlight the text border region even without supervision for the threshold map. This indicates that the border-like threshold map is beneficial to the final results. Thus, we apply border-like supervision on the threshold map for better guidance. An ablation study about the supervision is discussed in the Experiments section. For the usage, the text border map in (Xue, Lu, and Zhan 2018) is used to split the text instances while our threshold map is served as thresholds for the binarization.

Deformable convolution

Deformable convolution (Dai et al. 2017; Zhu et al. 2019) can provide a flexible receptive field for the model, which is especially beneficial to the text instances of extreme aspect ratios. Following (Zhu et al. 2019), modulated deformable convolutions are applied in all the 3×3 convolutional layers in stages conv3, conv4, and conv5 in the ResNet-18 or ResNet-50 backbone (He et al. 2016a).

Label generation

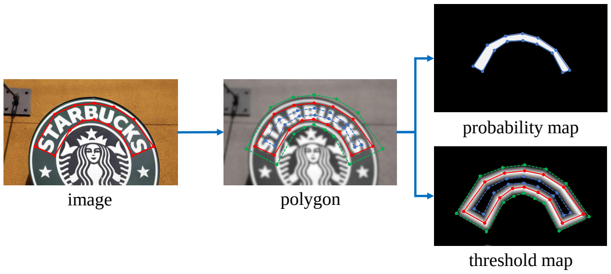

Figure 5: Label generation. The annotation of text polygon is visualized in red lines. The shrunk and dilated polygon are displayed in blue and green lines, respectively.

The label generation for the probability map is inspired by PSENet (Wang et al. 2019a). Given a text image, each polygon of its text regions is described by a set of segments:

$$G = \{S_k\}_{k=1}^{n} \tag{5}$$

n is the number of vertexes, which may be different in different datasets, e.g, 4 for the ICDAR 2015 dataset (Karatzas et al. 2015) and 16 for the CTW1500 dataset (Liu et al. 2019a). Then the positive area is generated by shrinking the polygon G to G_s using the Vatti clipping algorithm (Vati 1992). The offset D of shrinking is computed from the perimeter L and area A of the original polygon:

$$D = \frac{A(1 - r^2)}{L} \tag{6}$$

where r is the shrink ratio, set to 0.4 empirically.

With a similar procedure, we can generate labels for the threshold map. Firstly the text polygon G is dilated with the same offset D to G_d. We consider the gap between G_s and G_d as the border of the text regions, where the label of the threshold map can be generated by computing the distance to the closest segment in G.

Optimization

The loss function L can be expressed as a weighted sum of the loss for the probability map L_s, the loss for the binary map L_b, and the loss for the threshold map L_t:

$$L = L_s + \alpha \times L_b + \beta \times L_t \tag{7}$$

where L_s is the loss for the probability map and L_b is the loss for the binary map. According to the numeric values of the losses, α and β are set to 1.0 and 10 respectively.

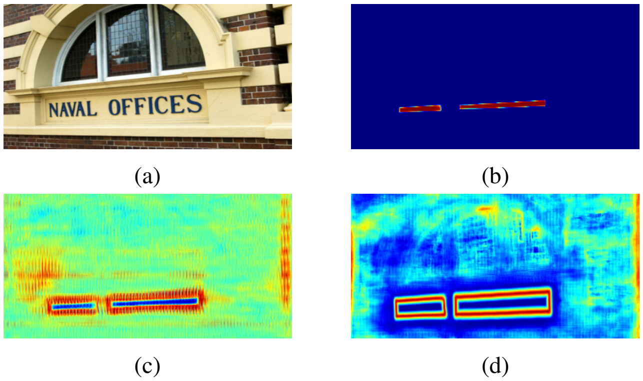

Figure 6: The threshold map with/without supervision. (a) Input image. (b) Probability map. (c) Threshold map without supervision. (d) Threshold map with supervision.

We apply a binary cross-entropy (BCE) loss for both L_s and L_b. To overcome the unbalance of the number of positives and negatives, hard negative mining is used in the BCE loss by sampling the hard negatives.

$$L_s = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log (1 - x_i) \tag{8}$$

S_l is the sampled set where the ratio of positives and negatives is $1 : 3$.

L_t is computed as the sum of $L1$ distances between the prediction and label inside the dilated text polygon G_d:

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*| \tag{9}$$

where R_d is a set of indexes of the pixels inside the dilated polygon G_d; y^* is the label for the threshold map.

In the inference period, we can either use the probability map or the approximate binary map to generate text bounding boxes, which produces almost the same results. For better efficiency, we use the probability map so that the threshold branch can be removed. The box formation process consists of three steps: (1) the probability map/the approximate binary map is firstly binarized with a constant threshold (0.2), to get the binary map; (2)the connected regions (shrunk text regions) are obtained from the binary map; (3) the shrunk regions are dilated with an offset D' the Vatti clipping algorithm(Vati 1992). D' is calculated as

$$D' = \frac{A' \times r'}{L'} \tag{10}$$

where A' is the area of the shrunk polygon; L' is the perimeter of the shrunk polygon; r' is set to 1.5 empirically.

Experiments

Datasets

SynthText (Gupta, Vedaldi, and Zisserman 2016) is a synthetic dataset which consists of $800k$ images. These images are synthesized from 8k background images. This dataset is only used to pre-train our model.

MLT-2017 dataset [1] is a multi-language dataset. It includes

[1] https://rrc.cvc.uab.es/?ch=8

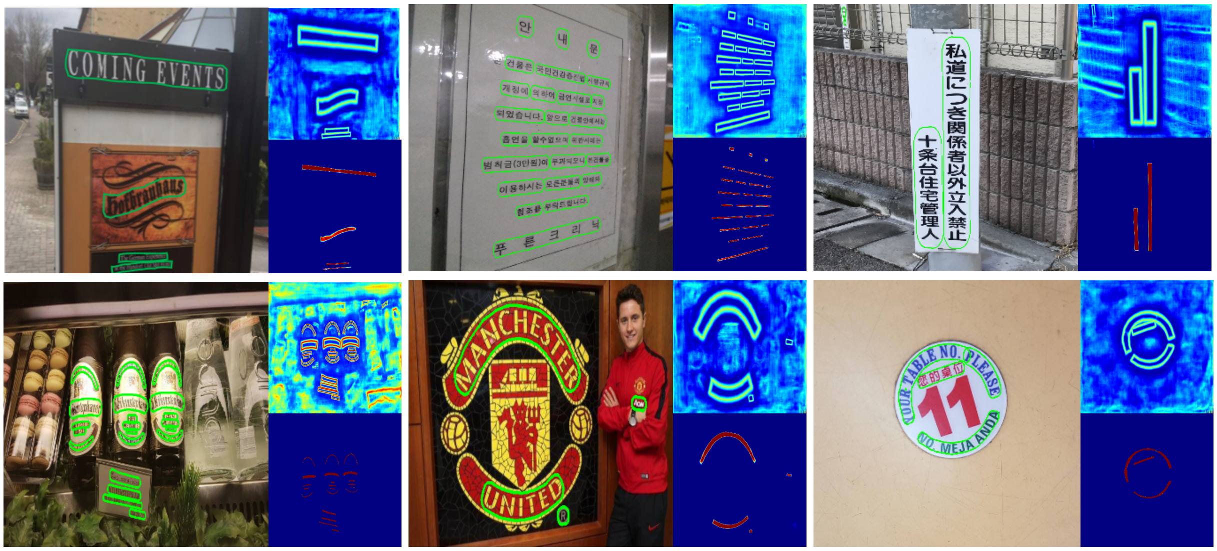

Figure 7: Some visualization results on text instances of various shapes, including curved text, multi-oriented text, vertical text, and long text lines. For each unit, the top right is the threshold map; the bottom right is the probability map.

Table 1: Detection results with different settings. "DConv" indiates deformable convolution. "P", "R", and "F" indicate precision, recall, and f-measure respectively.

Backbone	DConv	DB	MSRA-TD500				CTW1500			
			P	R	F	FPS	P	R	F	FPS
ResNet-18	×	×	85.5	70.8	77.4	**66**	76.3	72.8	74.5	**59**
ResNet-18	✓	×	86.8	72.3	78.9	62	80.9	75.4	78.1	55
ResNet-18	×	✓	87.3	75.8	81.1	**66**	82.4	76.6	79.4	**59**
ResNet-18	✓	✓	**90.4**	**76.3**	**82.8**	62	**84.8**	**77.5**	**81.0**	55
ResNet-50	×	×	84.6	73.5	78.7	**40**	81.6	72.9	77.0	**27**
ResNet-50	✓	×	90.5	77.9	83.7	32	86.2	78.0	81.9	22
ResNet-50	×	✓	86.6	77.7	81.9	**40**	84.3	79.1	81.6	**27**
ResNet-50	✓	✓	**91.5**	**79.2**	**84.9**	32	**86.9**	**80.2**	**83.4**	22

9 languages representing 6 different scripts. There are 7,200 training images, 1,800 validation images and 9,000 testing images in this dataset. We use both the training set and the validation set in the finetune period.

ICDAR 2015 dataset (Karatzas et al. 2015) consists of 1000 training images and 500 testing images, which are captured by Google glasses with a resolution of 720 × 1280. The text instances are labeled at the word level.

MSRA-TD500 dataset (Yao et al. 2012) is a multi-language dataset that includes English and Chinese. There are 300 training images and 200 testing images. The text instances are labeled in the text-line level. Following the previous methods (Zhou et al. 2017; Lyu et al. 2018b; Long et al. 2018), we include extra 400 training images from HUST-TR400 (Yao, Bai, and Liu 2014).

CTW1500 dataset CTW1500 (Liu et al. 2019a) is a dataset which focuses on the curved text. It consists of 1000 training images and 500 testing images. The text instances are annotated in the text-line level.

Total-Text dataset Total-Text (Chng and Chan 2017) is a dataset that includes the text of various shapes, including horizontal, multi-oriented, and curved. They are 1255 training images and 300 testing images. The text instances are labeled at the word level.

Implementation details

For all the models, we first pre-train them with the SynthText dataset for $100k$ iterations. Then, we finetune the models on the corresponding real-world datasets for 1200 epochs. The training batch size is set to 16. We follow a poly learning rate policy where the learning rate at current iteration equals the initial learning rate multiplying $(1 - \frac{iter}{max\_iter})^{power}$, where the initial learning rate is set to 0.007 and $power$ is 0.9. We use a weight decay of 0.0001 and a momentum of 0.9. The max_iter means the maximum iterations, which depends on the maximum epochs.

The data augmentation for the training data includes: (1) Random rotation with an angle range of $(-10°, 10°)$; (2) Random cropping; (3) Random Flipping. All the processed images are re-sized to 640×640 for better training efficiency.

In the inference period, we keep the aspect ratio of the test images and re-size the input images by setting a suitable height for each dataset. The inference speed is tested with a batch size of 1, with a single 1080ti GPU in a single thread. The inference time cost consists of the model forward time cost and the post-processing time cost. The post-processing time cost is about 30% of the inference time.

Ablation study

We conduct an ablation study on the MSRA-TD500 dataset and the CTW1500 dataset to show the effectiveness of our proposed differentiable binarization, the deformable convolution, and different backbones. The detailed experimental results are shown in Tab. 1.

Differentiable binarization In Tab. 1, we can see that our proposed DB improves the performance significantly for both ResNet-18 and ResNet-50 on the two datasets. For the ResNet-18 backbone, DB achieves 3.7% and 4.9% performance gain in terms of F-measure on the MSRA-TD500 dataset and the CTW1500 dataset. For the ResNet-50 backbone, DB brings 3.2% (on the MSRA-TD500 dataset) and 4.6% (on the CTW1500 dataset) improvements. Moreover, since DB can be removed in the inference period, the speed is the same as the one without DB.

Deformable convolution As shown in Tab. 1, the deformable convolution can also brings $1.5 - 5.0$ performance gain since it provides a flexible receptive field for the backbone, with small extra time costs. For the MSRA-TD500 dataset, the deformable convolution increase the F-measure by 1.5% (with ResNet-18) and 5.0% (with ResNet-50). For the CTW1500 dataset, 3.6% (with ResNet-18) and 4.9% (with ResNet-50) improvements are achieved by the deformable convolution.

Table 2: Effect of supervising the threshold map on the MLT-2017 dataset. “Thr-Sup” denotes applying supervision on the threshold map.

Backbone	Thr-Sup	P	R	F	FPS
ResNet-18	×	81.3	63.1	71.0	41
ResNet-18	✓	**81.9**	**63.8**	**71.7**	41
ResNet-50	×	81.5	64.6	72.1	19
ResNet-50	✓	**83.1**	**67.9**	**74.7**	19

Supervision of threshold map Although the threshold maps with/without supervision are similar in appearance, the supervision can bring performance gain. As shown in Tab. 2, the supervision improves 0.7% (ResNet-18) and 2.6% (ResNet-50) on the MLT-2017 dataset.

Backbone The proposed detector with ResNet-50 backbone achieves better performance than the ResNet-18 but runs slower. Specifically, The best ResNet-50 model outperforms the best ResNet-18 model by 2.1% (on the MSRA-TD500 dataset) and 2.4% (on the CTW1500 dataset), with approximate double time cost.

Comparisons with previous methods

We compare our proposed method with previous methods on five standard benchmarks, including two benchmarks for curved text, one benchmark for multi-oriented text, and two multi-language benchmarks for long text lines. Some qualitative results are visualized in Fig. 7.

Table 3: Detection results on the Total-Text dataset. The values in the bracket mean the height of the input images. “*” indicates testing with multiple scales. “MTS” and “PSE” are short for Mask TextSpotter and PSENet.

Method	P	R	F	FPS
TextSnake (Long et al. 2018)	82.7	74.5	78.4	-
ATRR (Wang et al. 2019b)	80.9	76.2	78.5	-
MTS (Lyu et al. 2018a)	82.5	75.6	78.6	-
TextField (Xu et al. 2019)	81.2	79.9	80.6	-
LOMO (Zhang et al. 2019)*	87.6	79.3	83.3	-
CRAFT (Baek et al. 2019)	87.6	79.9	83.6	-
CSE (Liu et al. 2019b)	81.4	79.1	80.2	-
PSE-1s (Wang et al. 2019a)	84.0	78.0	80.9	3.9
DB-ResNet-18 (800)	**88.3**	77.9	82.8	**50**
DB-ResNet-50 (800)	87.1	**82.5**	**84.7**	32

Table 4: Detection results on CTW1500. The methods with “*” are collected from (Liu et al. 2019a). The values in the bracket mean the height of the input images.

Method	P	R	F	FPS
CTPN*	60.4	53.8	56.9	7.14
EAST*	78.7	49.1	60.4	21.2
SegLink*	42.3	40.0	40.8	10.7
TextSnake (Long et al. 2018)	67.9	**85.3**	75.6	1.1
TLOC (Liu et al. 2019a)	77.4	69.8	73.4	13.3
PSE-1s (Wang et al. 2019a)	84.8	79.7	82.2	3.9
SAE (Tian et al. 2019)	82.7	77.8	80.1	3
Ours-ResNet18 (1024)	84.8	77.5	81.0	**55**
Ours-ResNet50 (1024)	**86.9**	80.2	**83.4**	22

Curved text detection We prove the shape robustness of our method on two curved text benchmarks (Total-Text and CTW1500). As shown in Tab. 3 and Tab. 4, our method achieves state-of-the-art performance both on accuracy and speed. Specifically, “DB-ResNet-50” outperforms the previous state-of-the-art method by 1.1% and 1.2% on the Total-Text and the CTW1500 dataset. “DB-ResNet-50” runs faster than all previous method and the speed can be further improved by using a ResNet-18 backbone, with a small performance drop. Compared to the recent segmentation-based detector (Wang et al. 2019a), which runs 3.9 FPS on Total-Text, “DB-ResNet-50 (800)” is 8.2 times faster and “DB-ResNet-18 (800)” is 12.8 times faster.

Multi-oriented text detection The ICDAR 2015 dataset is a multi-oriented text dataset that contains lots of small and low-resolution text instances. In Tab. 5, we can see that

"DB-ResNet-50 (1152)" achieves the state-of-the-art performance on accuracy. Compared to the previous fastest method (Zhou et al. 2017), "DB-ResNet-50 (736)" outperforms it by 7.2% on accuracy and runs twice faster. For "DB-ResNet-18 (736)", the speed can be 48 fps when ResNet-18 is applied to the backbone, with an f-measure of 82.3.

Table 5: Detection results on the ICDAR 2015 dataset. The values in the bracket mean the height of the input images. "TB" and "PSE" are short for TextBoxes++ and PSENet.

Method	P	R	F	FPS
CTPN (Tian et al. 2016)	74.2	51.6	60.9	7.1
EAST (Zhou et al. 2017)	83.6	73.5	78.2	13.2
SSTD (He et al. 2017a)	80.2	73.9	76.9	7.7
WordSup (Hu et al. 2017)	79.3	77	78.2	-
Corner (Lyu et al. 2018b)	**94.1**	70.7	80.7	3.6
TB (Liao, Shi, and Bai 2018)	87.2	76.7	81.7	11.6
RRD (Liao et al. 2018)	85.6	79	82.2	6.5
MCN (Liu et al. 2018)	72	80	76	-
TextSnake (Long et al. 2018)	84.9	80.4	82.6	1.1
PSE-1s (Wang et al. 2019a)	86.9	84.5	85.7	1.6
SPCNet (Xie et al. 2019a)	88.7	**85.8**	87.2	-
LOMO (Zhang et al. 2019)	91.3	83.5	87.2	-
CRAFT (Baek et al. 2019)	89.8	84.3	86.9	-
SAE(720) (Tian et al. 2019)	85.1	84.5	84.8	3
SAE(990) (Tian et al. 2019)	88.3	85.0	86.6	-
DB-ResNet-18 (736)	86.8	78.4	82.3	**48**
DB-ResNet-50 (736)	88.2	82.7	85.4	26
DB-ResNet-50 (1152)	91.8	83.2	**87.3**	12

Multi-language text detection Our method is robust on multi-language text detection. As shown in Tab. 6 and Tab. 7, "DB-ResNet-50" is superior to previous methods on accuracy and speed. For the accuracy, "DB-ResNet-50" surpasses the previous state-of-the-art method by 1.9% and 3.8% on the MSRA-TD500 and MLT-2017 dataset respectively. For the speed, "DB-ResNet-50" is 3.2 times faster than the previous fastest method (Liao et al. 2018) on the MSRA-TD500 dataset. With a light-weight backbone, "DB-ResNet-18 (736)" achieves a comparative accuracy compared to the previous state-of-the-art method (Liu et al. 2018) (82.8 vs 83.0) and runs at 62 FPS, which is 6.2 times faster than the previous fastest method (Liao et al. 2018), on the MSRA-TD500. The speed can be further accelerated to 82 FPS ("ResNet-18 (512)") by decreasing the input size.

Limitation

One limitation of our method is that it can not deal with cases "text inside text", which means that a text instance is inside another text instance. Although the shrunk text regions are helpful to the cases that the text instance is not in the center region of another text instance, it fails when the text instance exactly locates in the center region of another text instance. This is a common limitation for segmentation-based scene text detectors.

Table 6: Detection results on the MSRA-TD500 dataset. The values in the bracket mean the height of the input images.

Method	P	R	F	FPS
(He et al. 2016b)	71	61	69	-
DeepReg (He et al. 2017b)	77	70	74	1.1
RRPN (Ma et al. 2018)	82	68	74	-
RRD (Liao et al. 2018)	87	73	79	10
MCN (Liu et al. 2018)	88	79	83	-
PixelLink (Deng et al. 2018)	83	73.2	77.8	3
Corner (Lyu et al. 2018b)	87.6	76.2	81.5	5.7
TextSnake (Long et al. 2018)	83.2	73.9	78.3	1.1
(Xue, Lu, and Zhan 2018)	83.0	77.4	80.1	-
(Xue, Lu, and Zhang 2019)	87.4	76.7	81.7	-
CRAFT (Baek et al. 2019)	88.2	78.2	82.9	8.6
SAE (Tian et al. 2019)	84.2	81.7	82.9	-
DB-ResNet-18 (512)	85.7	73.2	79.0	**82**
DB-ResNet-18 (736)	90.4	76.3	82.8	62
DB-ResNet-50 (736)	**91.5**	**79.2**	**84.9**	32

Table 7: Detection results on the MLT-2017 dataset. Methods with "*" are collected from (Lyu et al. 2018b). The images in the MLT-2017 dataset are re-sized to 768×1024 in our method. "PSE" is short for PSENet.

Method	P	R	F	FPS
SARI_FDU_RRPN_V1*	71.2	55.5	62.4	-
Sensetime_OCR*	56.9	**69.4**	62.6	-
SCUT_DLVlab1*	80.3	54.5	65.0	-
e2e_ctc01_multi_scale*	79.8	61.2	69.3	-
Corner (Lyu et al. 2018b)	83.8	55.6	66.8	-
PSE (Wang et al. 2019a)	73.8	68.2	70.9	-
DB-ResNet-18	81.9	63.8	71.7	**41**
DB-ResNet-50	**83.1**	67.9	**74.7**	19

Conclusion

In this paper, we have presented a novel framework for detecting arbitrary-shape scene text, which includes the proposed differentiable binarization process (DB) in a segmentation network. The experiments have verified that our method (ResNet-50 backbone) consistently outperforms the state-the-the-art methods on five standard scene text benchmarks, in terms of speed and accuracy. In particular, even with a lightweight backbone (ResNet-18), our method can achieve competitive performance on all the testing datasets with real-time inference speed. In the future, we are interested in extending our method for end-to-end text spotting.

Acknowledgments

This work was supported by National Key R&D Program of China (No. 2018YFB1004600), to Dr. Xiang Bai by the National Program for Support of Top-notch Young Professionals and the Program for HUST Academic Frontier Youth Team 2017QYTD08.

References

Baek, Y.; Lee, B.; Han, D.; Yun, S.; and Lee, H. 2019. Character region awareness for text detection. In *Proc. CVPR*, 9365–9374.

Chng, C. K., and Chan, C. S. 2017. Total-text: A comprehensive dataset for scene text detection and recognition. In *Proc. ICDAR*, 935–942.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proc. ICCV*, 764–773.

Deng, D.; Liu, H.; Li, X.; and Cai, D. 2018. Pixellink: Detecting scene text via instance segmentation. In *Proc. AAAI*.

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

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

He, T.; Huang, W.; Qiao, Y.; and Yao, J. 2016b. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. Image Processing* 25(6):2529–2541.

He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; and Li, X. 2017a. Single shot text detector with regional attention. In *Proc. ICCV*, 3047–3055.

He, W.; Zhang, X.; Yin, F.; and Liu, C. 2017b. Deep direct regression for multi-oriented scene text detection. In *Proc. ICCV*.

Hu, H.; Zhang, C.; Luo, Y.; Wang, Y.; Han, J.; and Ding, E. 2017. Wordsup: Exploiting word annotations for character based text detection. In *Proc. ICCV*, 4940–4949.

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 *Proc. ICDAR*.

Kim, K.; Cheon, Y.; Hong, S.; Roh, B.; and Park, M. 2016. PVANET: deep but lightweight neural networks for real-time object detection. *CoRR* abs/1608.08021.

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

Liao, M.; Zhu, Z.; Shi, B.; Xia, G.; and Bai, X. 2018. Rotation-sensitive regression for oriented scene text detection. In *Proc. CVPR*, 5909–5918.

Liao, M.; Lyu, P.; He, M.; Yao, C.; Wu, W.; and Bai, X. 2019. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*

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

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

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; and Reed, S. E. 2016. SSD: single shot multibox detector. In *Proc. ECCV*.

Liu, Z.; Lin, G.; Yang, S.; Feng, J.; Lin, W.; and Goh, W. L. 2018. Learning markov clustering networks for scene text detection. In *Proc. CVPR*, 6936–6944.

Liu, Y.; Jin, L.; Zhang, S.; Luo, C.; and Zhang, S. 2019a. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition* 90:337–345.

Liu, Z.; Lin, G.; Yang, S.; Liu, F.; Lin, W.; and Goh, W. L. 2019b. Towards robust curve text detection with conditional spatial expansion. In *Proc. CVPR*, 7269–7278.

Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; and Yao, C. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proc. ECCV*, 20–36.

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 *Proc. 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 *Proc. 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 Trans. on Multimedia* 20(11):3111–3122.

Shi, B.; Bai, X.; and Belongie, S. J. 2017. Detecting oriented text in natural images by linking segments. In *Proc. CVPR*.

Tian, Z.; Huang, W.; He, T.; He, P.; and Qiao, Y. 2016. Detecting text in natural image with connectionist text proposal network. In *Proc. ECCV*.

Tian, Z.; Shu, M.; Lyu, P.; Li, R.; Zhou, C.; Shen, X.; and Jia, J. 2019. Learning shape-aware embedding for scene text detection. In *Proc. CVPR*, 4234–4243.

Vati, B. R. 1992. A generic solution to polygon clipping. *Communications of the ACM* 35(7):56–64.

Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; and Shao, S. 2019a. Shape robust text detection with progressive scale expansion network. In *Proc. CVPR*, 9336–9345.

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

Xie, E.; Zang, Y.; Shao, S.; Yu, G.; Yao, C.; and Li, G. 2019a. Scene text detection with supervised pyramid context network. In *Proc. AAAI*, volume 33, 9038–9045.

Xie, L.; Liu, Y.; Jin, L.; and Xie, Z. 2019b. Derpn: Taking a further step toward more general object detection. In *Proc. AAAI*, volume 33, 9046–9053.

Xu, Y.; Wang, Y.; Zhou, W.; Wang, Y.; Yang, Z.; and Bai, X. 2019. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Trans. Image Processing* 28(11):5566–5579.

Xue, C.; Lu, S.; and Zhan, F. 2018. Accurate scene text detection through border semantics awareness and bootstrapping. In *Proc. ECCV*, 355–372.

Xue, C.; Lu, S.; and Zhang, W. 2019. MSR: multi-scale shape regression for scene text detection. In *Pro. IJCAI*, 989–995.

Yao, C.; Bai, X.; and Liu, W. 2014. A unified framework for multioriented text detection and recognition. *IEEE Trans. Image Processing* 23(11):4737–4749.

Yao, C.; Bai, X.; Liu, W.; Ma, Y.; and Tu, Z. 2012. Detecting texts of arbitrary orientations in natural images. In *Proc. CVPR*.

Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; and Bai, X. 2016. Multi-oriented text detection with fully convolutional networks. In *Proc. CVPR*.

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 *Proc. CVPR*.

Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; and Liang, J. 2017. EAST: an efficient and accurate scene text detector. In *Proc. CVPR*.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *Proc. CVPR*, 9308–9316.