Fast Arbitrary Shaped Scene Text Detection via Text Discriminator

Chengbin Zeng¹² and Chunli Song¹

¹Guizhou Institute of Technology,Guiyang,Guiyzhou,China
²Guizhou Carefreesky Technology Co., Ltd, Guiyang, Guiyizhou, China
Email: cbzeng@git.edu.cn; chunlisong@git.edu.cn

Abstract. Robust scene text detection is one of the difficult and significant challenges in the computer vision community. Most previous methods detect arbitrary-shaped text using complicated post-processing steps. In this paper, we propose a trainable fast arbitrary-shaped text detection network by using the text discriminator, sharing visual information among the two complementary tasks. Specifically, we extend PSENet [1] by adding a text discriminator to fuse multiple predictions for each text instance, rather than using complicated post-processing steps which are time consuming. The text discriminator shares visual information with text detection network, and thus can achieve much faster detection speed compared with PSENet, while maintaining a similar accuracy reported in PSENet. Furthermore, our text discriminator can reduce the false alarms effectively. Experiments on ICDAR 2017 MLT, ICDAR 2015, and ICDAR 2019 ART datasets demonstrate that the proposed approach can achieve nearly real-time detection speed while keeping state-of-the-art detection accuracy.

Keywords. Scene text detection; PSENet; Text discriminator; real-time detection speed.

1. Introduction

Scene text detection is one of the difficult and significant challenges in the computer vision community and has various application scenarios. Over the past few years, deep learning based scene text detection have achieved great success. Jaderberg et al. [2] firstly proposed a CNN based method for scene text detection and recognition. Liao et al. [3] modified the SSD [4] networks to detect various aspect ratios of scene text. Borisyuk et al. [5] adopted famous object detection network Faster-RCNN [6] for text detection, and a sequence-to-sequence CTC loss [7] for training text recognition.

In recent years, segmentation-based text detection, which were inspired by the fully convolutional networks (FCN) [8], have achieved great performances. EAST [9] adopted FCN to directly segment the text areas and predict their geometries. Textsnake [10] effectively detect the text instances in horizontal, oriented and curved forms by using FCN with a series of ordered disks. PSENet [1] adopted FCN to assign each text instance with multiple predicted segmentation areas, and then used a progressive scale expansion algorithm to obtain the final detection. However, many of these works used complicated post-processing steps and thus the inference is time consuming, which is hard to deploy in the real-world environment.

To address this problem, we propose a text discriminator based arbitrary-shaped text detection approach. As the illustration of figure 1, we first predict each text instance with multiple segmentation areas by using FCN. Then, a trainable text discriminator is used to fuse the multiple segmentations. Our text discriminator can share visual information with text detection network, and can be easily trained without using complicated post-processing. Thus, the inference speed our network is very fast.
Furthermore, our text discriminator can reduce the false alarms effectively (figure 1). Experiments on ICDAR 2017 MLT, ICDAR 2019 ART and ICDAR 2015 datasets demonstrate that our method can achieve nearly real-time detection speed while keeping state-of-the-art detection accuracy.

Figure 1. (a) The scene text image; (b) the multiple predicted segmentation results; (c) the result of feature fusion using the text discriminator; (d) the final detect result. It shows that our method can fuse the segmentation results and reduce false alarms effectively.

2. Proposed Method

2.1. Overall Architecture

An overview of our architecture is illustrated in figure 2. We follow a segmentation-based pipeline with a trainable text discriminator to detect arbitrary-shaped text instances. The backbone of our network is ResNet [11]. The low-level and high-level semantic features of ResNet are concatenated into a feature map. Then, we project the squeezed feature map into n segmentation results. Each segmentation is a text mask for one certain text instances at a certain scale. Finally, the text discriminator is used to fuse multiple predictions for each text instance.

2.2. Network Design

We adopt the 50-layers ResNets [11] as the feature extractor stem of our network. The ResNets extracts four levels of feature maps (denoted as F1, F2, F3, F4, see figure 2b) from the input image, whose size are $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$ of the input image. Then we modify the strategy described in EAST [9] to gradually merge these four maps:

Figure 2. Illustration of our overall architecture: (a) is the scene text image; (b) is the FCN; (c) refers to the multiple predicted segmentation results; (d) is the feature fusion by using a trainable text discriminator; (e) is the final detect result.
\[ G_i = \begin{cases} \text{upsample}(H_i) & \text{if } i < 3 \\ \text{smooth}(H_i) & \text{if } i = 4 \end{cases} \]  

\[ H_i = \begin{cases} F_i & \text{if } i = 1 \\ \text{smooth}(F_i) & \text{otherwise} \end{cases} \]

where \( G_i \) is the merge base, and \( H_i \) refers to concatenated feature map, and the smooth operator refer to the \([\text{Conv}(3,3), \text{Batch Normalization}[12], \text{rectified linear units}]\) operation. Then, we follow the steps of PSENet [1], transform the feature map with a \( \text{Conv}(1,1) \)-Sigmoid operation and produces \( n \) segmentation maps \( S_1, S_2, \ldots, S_n \) (figure 2c).

### 2.3. Multiple Segmentations Fusion

The procedure of our text discriminator algorithm is explained in figure 3, whose central idea is to score for each of the segmentation area. We choose the segmentation with the highest score from the multiple segmentation results for the same text instance, and abandon the one whose score is lower than certain threshold (e.g., 0.5).

![Figure 3. Multiple segmentations fusion.](image)

At first, we obtain RoI using the bounding box operation for each of these segmentation results. Then, we project the RoI to the feature maps produced by FCN, and perform the RoIAlign [13] operation for extracting a small feature map (11x11 is used in this paper) from the RoI. Subsequently, the small feature map is fed into the smooth layer. Finally, the score of RoI is computed by the full connection & softmax layer.

By using the RoIAlign operation, our text discriminator can share visual information with text detection network, and thus can achieve much fast detection speed compared with PSENet. Furthermore, if the score of RoI is lower than certain threshold, our text discriminator treats it background, and thus can reduce the false alarms effectively.

### 2.4. Label Generation

As illustrated in figures 1 and 2, our networks need to produce multiple segmentation results with different scales. Therefore, multiple scales of corresponding ground truths are required during training. We follow the steps of PSENet [1] to shrink the original curved polygon of text area. Then, each shrunk polygon is converted into a binary mask as the ground truth of multiple scales. We also obtain the bounding box of each text instance for computing the loss function.

### 2.5. Loss Function

The training process of our network is divided into two stages. The first stage is to train an FCN that can predict each text instance with multiple segmentation results, and the second stage is to train the
A text discriminator network that can fuse the multiple segmentation results. The model trained in the first step is used as an initialization of the text discriminator network.

For the first step of training, we adopt the loss function described in \cite{pse2019} which can be written as follows:

$$
\mathcal{L} = \lambda \mathcal{L}_c + (1 - \lambda) \mathcal{L}_s
$$

(3)

where $\mathcal{L}_c$ is the loss of the original text instances and $\mathcal{L}_s$ is the loss of the shrunk text, and $\lambda$ is the balance factor. $\mathcal{L}_c$ and $\mathcal{L}_s$ are mainly described in dice loss \cite{dice} and OHEM (Online Hard Example Mining) \cite{ohem}. See \cite{pse2019} for detailed description.

For the second step of training, we adopt the multi-task loss defined in \cite{mask_rcnn2017} which can be written as follows:

$$
\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask}
$$

(4)

where $\mathcal{L}_{cls}$ is the text discrimination loss, $\mathcal{L}_{box}$ is the bounding-box loss, and $\mathcal{L}_{mask}$ is the segmentation loss. See \cite{mask_rcnn2017} for detailed description.

3. Experiment

3.1. Benchmark Datasets

**ICDAR 2015** \cite{icdar2015} includes 1500 pictures, 1000 of which are used for training and the rest of which are for evaluation. The text regions are annotated by 4 vertices of the quadrangle. This dataset is commonly used for oriented text detection.

**ICDAR 2017 MLT** \cite{icdar2017} includes 7200 training images, 1800 validation images and 9000 testing images. This dataset is a large scale multi-lingual text detection dataset, and contains a lot of scene text with arbitrary orientations.

**ICDAR 2019 ART** \cite{icdar2019} contains a training set with 5603 images, and a testing set of 4563 images, which is aimed at introducing the arbitrary-shaped (i.e., horizontal, multi-oriented, and curved) text problem to the scene text community. Text instances in this dataset were annotated with quadrilateral bounding boxes, 8, 10 and 12 vertexes polygon bounding box.

3.2. Implementation Details

For the first stage of training, we use the PSENet \cite{pse2019} pre-trained on ICDAR 2017 MLT as our backbone, and optimize the network by using stochastic gradient descent (SGD). The number of segmentations is 7, and the balance factor $\lambda$ is set to 0.7. The batch size is 16 on 4 GTX 1080Ti GPUs. The number of iterations is 20K, and the initial learning rate is $1 \times 10^{-4}$, decays to one-tenth every 5k iterations. Subsequently, the model is used as an initialization of the second stage of training, which uses the ADAM optimizer, and set the initial learning rate to $1 \times 10^{-3}$, divide the learning rate by 10 at 30K and 60K. The second stage of network training is stopped until the performance cannot be improved.

3.3. Quantitative Results

We evaluate our algorithm based on the standard recall, precision, and F-score metrics. Tables 1 and 2 show that the detection accuracy of our proposed approach is comparable to state-of-the-art methods, while achieving nearly real-time detection speed.

In ICDAR 2015 Challenge 4, the 1000 of pictures are used for training, and the rest of 500 pictures are for evaluation. The proposed method achieves a F-score of 85.59% when using the original scale images, which is 7.39% higher than EAST \cite{east} and 2.99% higher than TextSnake \cite{textsnake} and 0.1% lower than PSENet \cite{pse2019} on the F-measure. Notably, the detection speed of our approach is 11.5fps, which is much faster than TextSnake and PSENet (table 1).

We then test our method in ICDAR 2017 MLT to evaluate the robustness of our method to multiple languages. Table 2 shows that our proposed method can achieve a F-score of 71.89%, which is 5%
higher than Lyu et al.’s method [19] and 0.24% lower than PSENet [1] on the F-measure. This proves that our method can accurately locate the scene text in complex natural scenarios.

Table 1. Comparisons with state-of-the-art methods on ICDAR 2015 Challenge 4.

| Method      | ICDAR 2015 |            |            |            |
|-------------|------------|------------|------------|------------|
|             | Recall     | Precision  | F-score    | FPS        |
| EAST [9]    | 73.47      | 83.57      | 78.2       | 13.2       |
| TextSnake [10] | 80.4      | 84.9       | 82.6       | 1.1        |
| PSENET [1]  | 83.77      | 86.1       | 84.92      | 3.5        |
| Our approach| 83.6       | 85.8       | 84.68      | 11.5       |

Table 2. Comparisons with state-of-the-art methods on ICDAR 2017 MLT.

| Method         | ICDAR 2017 MLT |            |            |            |
|----------------|----------------|------------|------------|------------|
|                | Recall         | Precision  | F-score    | FPS        |
| Lyu et al. [19]| 55.6           | 83.8       | 66.8       | 6          |
| PSENET [1]     | 69.18          | 75.35      | 72.13      | 3.8        |
| Our approach   | 68.91          | 75.15      | 71.89      | 15.2       |

Finally, we evaluate the performance of our approach in different stages. As shown in table 3, the “1s” means the size of output feature map is same as that of the input image, and the “4s” means the size of output feature map is 1/4 of the input image. On ICDAR 2019 ART, if the long edge of input image is 1280 pixels, our approach-1s obtains the best F-score, and the performance of our approach-4s is slightly decreased, but the speed is faster. When the long edge of input image is scaled to 960 pixels, our approach-4s can achieve real-time detection speed (18fps) with the F-score is slightly decreased to 79.3. These results prove that our method can accurately locate the arbitrary-shaped text with real-time speed. Some detection results are shown in figure 4.

Table 3. Time consumption on ICDAR 2019 ART.

| Method         | ICDAR 2019 ART |            |            |
|----------------|----------------|------------|------------|
|                | Resolution     | F-score    | FPS        |
| Our approach-1s| 1280           | 82.5       | 11.5       |
| Our approach-4s| 1280           | 80.2       | 13         |
| Our approach-4s| 960            | 79.3       | 18         |

Figure 4. Scene text detection results on three benchmark datasets.
4. Conclusion and Future Work

We propose an arbitrary-shaped text detection approach based on the text discriminator. The text discriminator shares visual information with text detection network, and thus can achieve much faster detection speed compared with state-of-the-art methods, while maintaining a similar accuracy. In the future, we can extend our text discriminator to recognize the scene text, i.e., simultaneous detection and recognition of scene text with a unified network.

Acknowledgements

This work was partly supported by the following projects: the National Natural Science Foundation of China (Grant No 61966006), the Science Research Foundation for High-level Talents of Guizhou Institute of Technology (Grant No. XJGC20150108), the Science and Technology Foundation of Guizhou Province (Grant No [2020]1Y281) and the Innovation Team of Guizhou Provincial Education Department (Grant No [2014]34).

References

[1] Wang W H, Xie E, Li X, et al. 2019 Shape robust text detection with progressive scale expansion network *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[2] Jaderberg M, Simonyan K, Vedaldi A and Zisserman A 2016 Reading text in the wild with convolutional neural networks *International Journal of Computer Vision (IJCV)* 116 (1) 1-20.

[3] Liao M, Shi B and Bai X 2017 Textboxes: A fast text detector with a single deep neural network *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

[4] Liu W, Anguelov D, Erhan D, et al. 2016 SSD: Single shot multibox detector *European Conference on Computer Vision (ECCV)*.

[5] Borisyuk F, Gordo A and Sivakumar V 2018 Rosetta: Large scale system for text detection and recognition in images *24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[6] Ren S, He K M, Girshick R and Sun J 2015 Faster R-CNN: Towards real-time object detection with region proposal networks *Advances in Neural Information Processing Systems (NeurIPS)*.

[7] Graves A, Fernandez S, Gomez F and Schmidhuber J 2006 Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks *23rd International Conference on Machine learning (ICML)*.

[8] Long J, Shelhamer E, et al. 2015 Fully convolutional networks for semantic segmentation *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[9] Zhou X, Yao C and Wen H 2017 East: An efficient and accurate scene text detector. in *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[10] Ruan J, Zhang W, He X and Long S 2018 Textsnake: A flexible representation for detecting text of arbitrary shapes *Proceedings of European Conference on Computer Vision (ECCV)* (IEEE).

[11] He K M, Zhang X Y, Ren S Q and Sun J 2016 Deep residual learning for image recognition *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[12] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *arXiv preprint arXiv:1502.03167*.

[13] He K M, Gkioxari G, Dollr P and Girshick R 2017 Mask R-CNN *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[14] Milletari F, Navab N and Ahmadi S A 2016 U-net: Fully convolutional neural networks for volumetric medical image segmentation *Proceedings of International Conference on 3D Vision* (IEEE).
[15] Shrivastava A, Zupta A and Girshick R 2016 Training region-based object detectors with online hard example mining *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).

[16] Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A and Shafait F 2015 ICDAR 2015 competition on robust reading *International Conference on Document Analysis and Recognition (ICDAR)* (IEEE).

[17] ICDAR2017 competition on multi-lingual scene text detection http://rrc.cvc.uab.es/?ch=8&com=introduction.

[18] ICDAR2019 robust reading challenge on arbitrary-shaped text http://rrc.cvc.uab.es/?ch=14.

[19] Lyu P, Yao C, Wu W, Yan S and Bai X 2018 Multioriented scene text detection via corner localization and region segmentation *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (IEEE).