Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8682961, 9 pages
https://doi.org/10.1155/2022/8682961

*Research Article*

# An Agile and Efficient Neural Network Based on Knowledge Distillation for Scene Text Detection

**Weiwei Lin**,[1,2] **Zeqing Zhang**,[3] **and Xingsi Xue**[4]

[1]*School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuqing 350300, China*
[2]*Engineering Research Center for ICH Digitalization and Multisource Information Fusion, Fujian Province University, Fuqing 350300, China*
[3]*Department of Earth Science and Engineering, West Yunnan University of Applied Sciences, Dali 671000, China*
[4]*Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian 350118, China*

Correspondence should be addressed to Weiwei Lin; linww_cn@hotmail.com

Received 31 May 2022; Accepted 11 August 2022; Published 26 August 2022

Academic Editor: Hao Lu



Text detection is increasingly in demand recently and poses significant challenges for the tradeoff among the detection accuracy, memory resources, and inference speed in the case of applying to the portable device such as mobile phones. Current methods mainly focus on the detection accuracy but neglect either the running speed or the memory consumption. To this end, an agile and efficient neural network for scene text detection that balances the detection performance, running speed, and the model size is hereby proposed. In order to reduce the network parameters and speed up, the neural network for text detection is firstly pruned; and then, the pruned neural network is trained with the structured knowledge distillation for improving the detection performance. The method is implemented on three benchmark text datasets, i.e., ICDAR2015, Total-Text, and MSRA-TD500. The experimental results demonstrate that the hereby proposed method achieves the best comprehensive performance with a faster running speed and much less memory resources while the text detection accuracy is comparable to that acquired using the excellent text detection methods.

## 1. Introduction

Recently, given the practical applications of scene text images such as machine reading, mobile translation, and license plate recognition, related studies have been a hot topic in the field of computer vision. However, the varied shapes and styles of the scene text make it still a challenging task.

As the primary premise of recognition and translation, scene text detection is aimed at locating the text and giving the text bounding box. In recent years, arbitrary-shaped text detection has attracted increasing attention. Researchers keep attempting to design a complex semantic segmentation network for handling scene texts with assorted shapes, which generally demands huge computation time and redundant model parameters. Recent research results [1, 2] show that lightweight network structure can settle scene text detection tasks, and indicate that the current network is subject to sparsity and redundancy in this task.

To solve this problem, a simple and compact text detector that can trade off between speed and performance is hereby proposed based on DBNet. Inspired by [3], a pruning scheme is also proposed, which helps obtain a compact backbone network without affecting the feature fusion network. Additionally, structured distillation is adopted to improve the detection performance of the compact network. The hereby proposed method is compared with the excellent methods on MSRA-TD500, as shown in Figure 1, which is proven to achieve the fastest speed and successfully strike a balance between speed and performance.

Contributions of the present study can be summarized into three aspects: (i) an efficient scene text detection method is proposed, which can strike a balance between

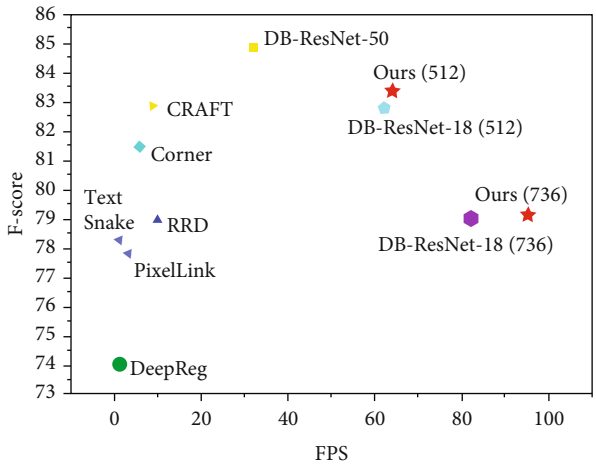


Figure 1: Comparison of different methods on MSRA-TD500. The hereby proposed methods strike a tradeoff between the speed and the $F$-score.

speed and accuracy; (ii) a pruning scheme is proposed for detection network, which helps obtain an agile and efficient detection network; and (iii) a distillation structure is designed for text detector, which transfers the dark knowledge of a complex text detector to the compact network and improves the detection performance of the compact network.

The rest of this paper is organized as follows: Section 2 briefly reviews related works on text detection; Section 3 details the proposed deep model; Section 4 gives experimental results and discusses the limitations; and finally, the conclusion is summarized in Section 5.

## 2. Related Work

In recent years, scene text detection based on deep learning has achieved great results. Most of those methods can be roughly divided into two categories, i.e., regression-based methods and segmentation-based methods. Heavy network is generally adopted for high detection accuracy; agile structures are designed by a few other methods to maintain a balance between speed and accuracy.

Regression-based methods are usually inspired by generic object detectors such as Faster R-CNN [4] and SSD [5]. Based on SSD, TextBoxes [6] adjusts the anchor size to fit the long and narrow text shape; TextBoxes++ [7] and DMPNet [8] detect multidirectional texts by regression quadrilateral; EAST [9] reverts to the text rotation angle to detect multidirectional texts; SegLink [10] regresses text segments and their links to handle long texts; RRPN [11] proposes the rotation position suggestion based on Faster R-CNN; and SR_DeepText [12] constructs a scale-robust network for solving the problem of text scaling. Regression-based methods can achieve remarkable results in regular-shaped texts but have hard time handling text instances of arbitrary shapes.

Segmentation-based methods combine the segmentation results with the postprocessing algorithm for the text boundary box. PixelLink [13] predicts the category and relationship of pixels to get the text bounding box; Mask Text Spotter [14] segments text semantics based on regression to the location of text instances for processing arbitrarily shaped texts; TextSnake [15] treats the arbitrary shape text detection problem as a search for text centerlines and text regions; and PSENet [16] effectively solves the text proximity problem by fusing the postprocessing algorithm to predict the text center fields of different scales. Segment-based approaches often require more complex postprocessing algorithms but can effectively handle text instances of arbitrary shapes.

Fast text detection methods seek a balance between speed and performance. EAST [9] uses a lightweight backbone network and simple NMS for a faster speed; PANNet [2] designs a lightweight feature fusion module with high efficiency and powerful feature expression; and DBNet [1] proposes a differentiable binarization pixel prediction. Both PANNet and DBNet require a lightweight backbone network to perform quickly and efficiently, and illustrate the sparsity and redundancy in the current network. In this case, the hereby proposed method is aimed at solving the network redundancy problem and striking a balance between speed and performance.

## 3. Methods

The pipeline of this method is shown in Figure 2. First, a streamlined PrunedNet is obtained by pruning the baseline; next, the input image is fed into both PrunedNet and TeacherNet, with a series of feature maps and prediction probability maps obtained during the process; finally, dark knowledge in TeacherNet enhances the detection performance of PrunedNet through structured distillation in training.

*3.1. Baseline.* To better verify the efficiency and performance of this approach, DBNet [1], an efficient text detector with small parameters that is easy to implement, is chosen as the baseline, with its structure illustrated in Figure 3. First, four feature maps of different sizes are extracted through a backbone; second, these feature maps are spliced together after bottom-up fusion; third, the network predicts the probability map and the threshold map (the edge map of the predicted text) based on the fused features, and the target binary map is obtained through differentiable binarization; finally, the text boundary box is obtained using the postprocessing algorithm based on the binary map.

*3.1.1. Deformable Convolution.* Following [17], DBNet replaces the second $3 \times 3$ convolutional layers of the residual block with deforming convolution in conv3, conv4, and conv5.

*3.1.2. Deforming Convolution.* Deforming convolution can change the receptive field of convolution kernel and adapt it to changeable text instances [18]. Differentiable binarization is proposed to solve the problem that standard binarization cannot be optimized during training, which is

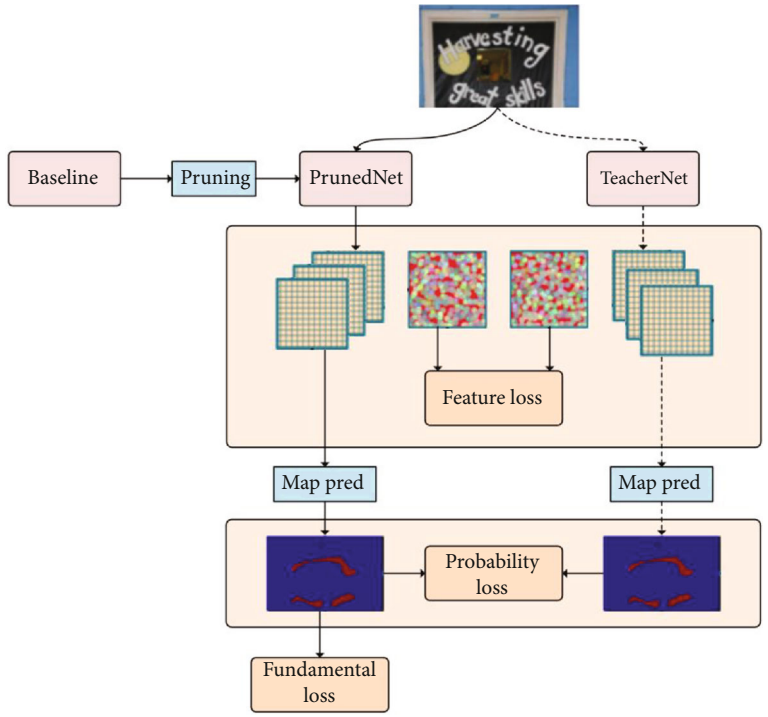


Figure 2: The structure of the hereby proposed method. Firstly, a PrinedNet network is obtained by pruning the baseline; secondly, the training data is introduced into two networks, which causes feature loss; finally, the prediction map generates Dice loss and BCE loss.

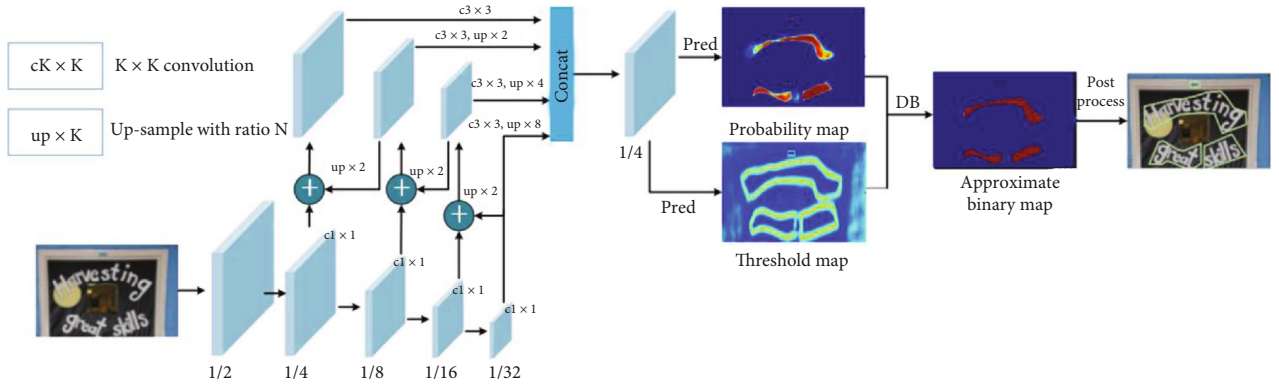


Figure 3: The pipeline of DBNet.

formulated as

$$B_{i,j} = \frac{1}{1 + e^{-k\left(P_{i,j} - T_{i,j}\right)}}, \tag{1}$$

where $B_{i,j}$ is the target binary map; $P_{i,j}$ and $T_{i,j}$, the probability map and the threshold map, respectively; and $k$, the amplifying factor, set to 50 during training. Differentiable binarization is similar to the sigmoid function, and makes the target binary map differentiable to better distinguish text boundaries.

*3.2. Prune.* The BN (Batch Normalization) layer, a data normalization operation in the network, is aimed at pulling the input data back to the standard normal distribution, so that the activated input values will fall in the region where the nonlinear function is sensitive to the input. Among them, learnable parameters $(\beta, \gamma)$ perform linear transformation on standard data. [3] points out that the channel feature will play a more limited role in subsequent operations such as convolution in the case of a small scaling factor $\gamma$. To this end, the convolution kernel of the network should be pruned according to the size of $\gamma$ for a compact network.

Inspired by [3], this paper proposes a pruning method for the detector. Simplifying the ResNet18 network structure using the pruning method can improve the detection speed while keeping the performance comparable to that of ResNet50. The algorithm flow is as follows:

(1) To select DBNet's backbone (ResNet18) and take the first BN layer of each residual block (yellow module in Figure 4)

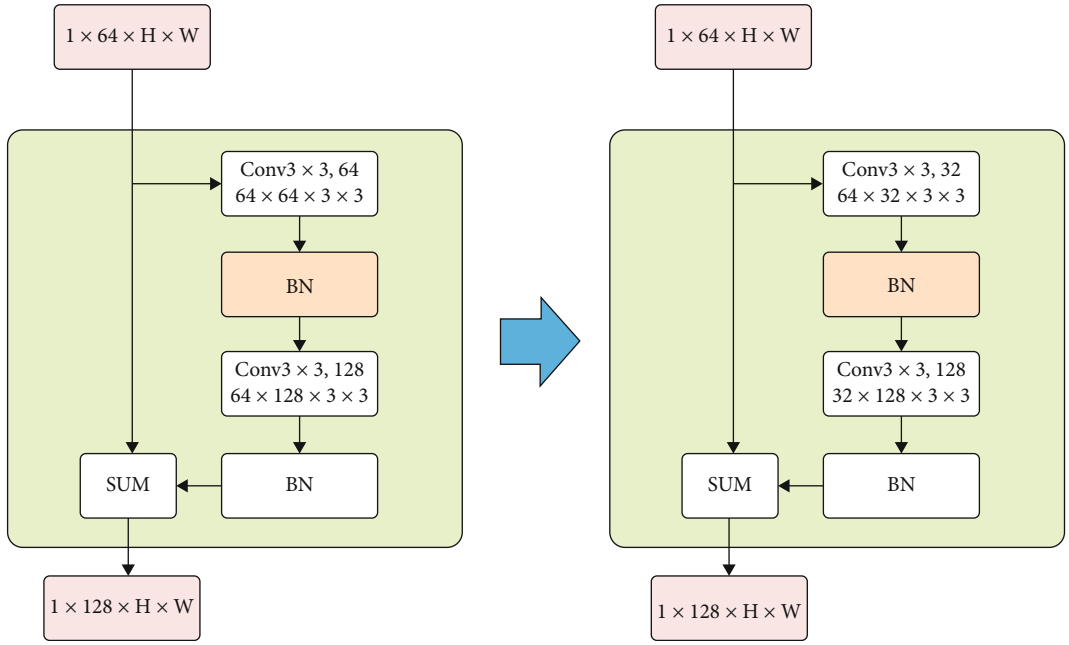


Figure 4: The pruning process of the hereby proposed method. The first convolution layer has fewer kernels, while the second has fewer convolution channels.

(2) To take the scaling factor of each such a BN layer to the set $A = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4 \cdots \gamma_m\}$

(3) To sort set $A$ for pruning the ratio $P_r$ and take the $K$ th largest scaling factor $\gamma_k$ according to $K = \lfloor m \times \Pr \rfloor$

(4) To traverse each BN layer and remove the convolution kernel corresponding to the upper and lower layers if $\gamma_i < \gamma_k$

The pruning process is shown in Figure 4. The parameter sizes of the upper and lower convolution kernel are reduced after calculating the yellow BN module. The number of the first convolution kernel is reduced from 64 to 32, which means that there are only 32 $\gamma$ greater than $\gamma_k$ in the BN layer. The second convolutional layer only changes the depth of the convolution kernel, because the number of channels in the previous feature map is 32 and the constant number of convolution kernels ensures the output of the residual block not to be affected.

Compared with [3], the pruning method in this paper retains the output channel of the convolution residual block, without affecting the subsequent feature fusion and knowledge distillation, which can also be extended to other areas, such as semantic segmentation and target tracking.

*3.3. Knowledge Distillation.* Knowledge distillation extracts the abstract knowledge learned from the complex network and improves the performance of the compact network. In order to achieve the same performance for two networks, some loss functions will be set to constrain the network during training for guaranteeing the consistency of their feature expressions. Inspired by [19, 20], the distillation framework is designed as shown in Figure 2, where the network of teachers in the distillation framework is DBNet with ResNet50 as the backbone network.

For the baseline, the four feature maps from the backbone network are unified in dimension through 1 × 1 convolution, respectively, and are provided with a one-to-one relationship in both the student network and the teacher network. In addition, there is a corresponding relationship between the fused feature maps, making it necessary to distil the knowledge from the teacher network into the student network using the $L_2$ loss constraint feature. The formula of feature map loss $L_f$ can be expressed as

$$L_f = \frac{1}{W \times H \times C} \sum_{i \in R} \sum_{j \in R} \sum_{k \in R} \left( y_{i,j,k} - x_{i,j,k} \right)^2, \tag{2}$$

where $W$, $H$, and $C$ represent the width, height, and channel number of the feature map, respectively; $y_{i,j,k}$, the eigenvalue of the teacher network; and $x_{i,j,k}$, the eigenvalue of the student network.

For the prediction map, KL Loss (Kullback-Leibler Divergence Loss) is adopted in most methods to make the probability distribution of the class prediction of the student network approximate to that of the teacher network, while for text detection, the classification of pixels is binary (text and background), with only a floating point number from 0 to 1 required to represent the probability. In this paper, Dice loss is used for constraining the predicted probability map and the binary map. The formula for the probability loss $L_p$ is as follows:

$$L_p = 1 - \frac{2|X \cap Y|}{|X| + |Y|}, \tag{3}$$

Table 1: The results of different pruning ratios. "$P$", "$R$", and "$F$" represent "Precision," "Recall," and "$F$-score," respectively.

| Method | Pruning ratio | $P$ (%) | $R$ (%) | $F$ (%) | FPS |
|---|---|---|---|---|---|
| DBNet | — | 88.3 | 82.5 | 82.8 | 50 |
| The proposed | 0.1 | 86.5 | 80.4 | 83.3 | 52 |
| The proposed | 0.2 | 86.6 | 80 | 83.2 | 54 |
| The proposed | 0.5 | 89.9 | 79.1 | 82.8 | 57 |

where $X$ denotes the prediction map of the student network and $Y$ represents the prediction map of the teacher network. During training, the feature loss can enhance the feature expression ability of the network, while the probability loss enables the network to flexibly predict the pixel. Combined with pruning, the simplified network still achieves excellent performance after removing redundant parameters.

*3.4. Loss Functions.* In this article, the proposed method inherits three losses from the baseline, i.e., probability map loss, binary map loss, and threshold map loss, the corresponding loss functions of which are defined as $L_s$, $L_b$, and $L_t$, where $L_s$ and $L_b$ adopt binary cross-entropy (BCE) loss. The formula can be expressed as

$$L_s = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log (1 - x_i), \tag{4}$$

where $x_{\mathrm{i}}$ is the predicted value and $y_{\mathrm{i}}$ represents the true value. $L_{\mathrm{t}}$ adopts $L_1$ loss, and the formula is as follows:

$$L_t = \sum_{i \in R_d} |y_i - x_i|. \tag{5}$$

Combined with the supervisory loss of knowledge distillation, the total experimental loss can be expressed as

$$L = L_s + \alpha \times L_b + \beta \times L_t + \sum_{i=1}^{5} L_{f_i} + L_{p_p} + L_{p_b}. \tag{6}$$

where $\alpha$ and $\beta$ are set as 1 and 10, respectively, in the experiment; $L_{f_i}$ represents the feature loss of the ith feature map; $L_{p_p}$, the probability loss of the probability map; and $L_{p_b}$, the probability loss of the binary map.

## 4. Experiment

In order to evaluate the performance of this method, the proposed method is implemented on three public benchmark datasets, i.e., Total-Text [21], ICDAR2015 [22], and MSRA-TD500 [23], and is compared with the excellent text detection methods. Then, ablation study is conducted to investigate the effects of pruning ratio and knowledge distillation.

*4.1. Datasets.* Total-Text is a word-level-based English curve text dataset, known as the first relatively large scene text dataset with three different text directions, i.e., horizontal, multidirectional, and curvilinear, and contains 1,255 training images and 300 test images.

ICDAR2015 is a multidirectional text dataset from Challenge 4 of ICDAR2015 Robust Reading Competition and contains 1500 training images and 500 test images collected from natural scenes. All text instances are labeled as multidirectional quadrilaterals.

MSRA-TD500 is a multilingual dataset containing both Chinese and English texts, involving 300 training images and 200 test images. The text instance is labeled as a rotating rectangle. In the experiment, 400 images from HUST-TR400 [24] are added to expand the training sample.

*4.2. Experimental Details.* For all models, the two models (ResNet18 and ResNet50) of DBNet are first trained according to [1]; then, the model of ResNet18 is pruned and fine-tuned; finally, the distillation training is carried out with the reduced model. During the training, the batch size is set to 16 when the model is trained alone and 8 for the distillation training due to memory limitations. In all the trainings, the number of epochs is 1,200; the initial learning rate is 0.007; and the learning rate is decreased exponentially, with a decrease rate of 0.9. The input size is $640 \times 640$, obtained by randomly flipping, rotating, and clipping.

During the whole testing, the same test image size is kept as the baseline. Although the GPU shows a little difference, the speed of the implementation is tested without any acceleration techniques using a single 2080 ti GPU in a single thread.

*4.3. Ablation Study.* Ablation study is conducted on Total-Text to investigate the effectiveness of pruning and knowledge distillation.

Table 1 shows the experimental results of the proposed method under different pruning ratios compared with the baseline. The accuracy begins to decrease with the increase of the pruning ratio, but the speed increases. When the pruning ratio reaches 0.1 or 0.2, this method can simultaneously improve both the performance and the speed compared to the baseline, and when the pruning ratio reaches 0.5, this method performs as well as the benchmark method in terms of $F$-score and achieves 7 gains in terms of FPS.

As shown in Table 2, the $F$-score is only 81.1% after pruning at the baseline, which presents a decrease from the baseline. However, after fine-tuning the model, the $F$-score reaches 82.7%, close to the baseline, but the $F$-score of the model reaches 83.0%, exceeding the baseline. Finally, under the supervision of probability loss, the performance of the model improves to 83.3%, with a 0.5% of gain in terms of $F$-score.

Table 2: The experiment results with different losses. Pr: the model is obtained by pruning (the pruning ratio is 0.1); Ft: the model is fine-tuned; Fl: feature loss is adopted; Pl: probability loss is adopted. "*P*," "*R*," and "*F*," represent "Precision," "Recall," and "*F*-score," respectively.

| Method | Pruning ratio | *P* (%) | *R* (%) | *F* (%) |
|---|---|---|---|---|
| DBNet | — | 88.3 | 82.5 | 82.8 |
| Pr | 0.1 | 83.4 | 80 | 81.1 |
| Pr+Ft | 0.1 | 85.5 | 80.1 | 82.7 |
| Pr+Ft+Fl | 0.1 | 86.4 | 79.8 | 83 |
| Pr+Ft+Fl+Pl | 0.1 | 86.5 | 80.4 | 83.3 |


Figure 5: Detection results on the three datasets.

Table 3: Comparison results on Total-Text. "*P*," "*R*," and "*F*" represent "Precision," "Recall," and "*F*-score", respectively. The best and second-best terms are highlighted in italic and bold, respectively.

| Method | *P* (%) | *R* (%) | *F* (%) | FPS |
|---|---|---|---|---|
| TextSnake [15] | 82.7 | 74.5 | 78.4 | — |
| ATRR [25] | 80.9 | 76.2 | 78.5 | — |
| MTS [14] | 82.5 | 75.6 | 78.6 | — |
| TextField [26] | 81.2 | 79.9 | 80.6 | — |
| LOMO [27] | 87.6 | 79.3 | 83.3 | — |
| CRAFT [28] | 87.6 | 79.9 | **83.6** | — |
| CSE [29] | 81.4 | 79.1 | 80.2 | — |
| PSE-1s [16] | 84 | 78 | 80.9 | 3.9 |
| DB-ResNet-18 [1] | 88.3 | 77.9 | 82.8 | 50 |
| DB-ResNet-50 [1] | 87.1 | 82.5 | *84.7* | 32 |
| The proposed (Pr0.1) | 86.5 | 80.4 | 83.3 | 52 |
| The proposed (Pr0.2) | 86.6 | 80 | 83.2 | **54** |
| The proposed (Pr0.5) | 86.9 | 79.1 | 82.8 | *57* |

*4.4. Comparisons with the Excellent Methods.* The hereby proposed method is compared with previous methods on three standard benchmarks, including a benchmark for curved text and two benchmarks for multioriented text. Some qualitative results are visualized in Figure 5.

The proposed method is compared with 10 text detection methods on the Total-Text, and the results are shown in Table 3. This method reaches 83.3% for the *F*-score, ranking third. In terms of speed, compared with the baseline (DB-ResNet-18), this method improves the speed without

Table 4: Comparison results on MSRA-TD500. "*P*," "*R*," and "*F*" represent "Precision," "Recall," and "*F*-score," respectively. The best and second-best terms are highlighted in italic and bold, respectively.

| Method | *P*(%) | *R*(%) | *F*(%) | FPS |
|---|---|---|---|---|
| DeepReg [30] | 77 | 70 | 74 | 1.1 |
| RRD [31] | 87 | 73 | 79 | 10 |
| MCN [32] | 88 | 79 | 83 | — |
| PixelLink [13] | 83 | 73.2 | 77.8 | 3 |
| Corner [33] | 87.6 | 76.2 | 81.5 | 5.7 |
| TextSnake [15] | 83.2 | 73.9 | 78.3 | 1.1 |
| CRAFT [28] | 88.2 | 78.2 | 82.9 | 8.6 |
| DB-ResNet-18 (512) [1] | 85.7 | 73.2 | 79 | 82 |
| DB-ResNet-18 (736) [1] | 90.4 | 76.3 | 82.8 | 62 |
| DB-ResNet-50 (736) [1] | 91.5 | 79.2 | *84.9* | 32 |
| The proposed (Pr0.5) (512) | 85.8 | 73.5 | 79.2 | *95* |
| The proposed (Pr0.5) (736) | 87.2 | 79.9 | **83.4** | **64** |

Table 5: Comparison results on ICDAR2015. "*P*," "*R*," and "*F*" represent "Precision," "Recall," and "*F*-score," respectively. The best and second-best terms are highlighted in italic and bold, respectively.

| Method | *P* (%) | *R* (%) | *F* (%) | FPS |
|---|---|---|---|---|
| EAST [9] | 83.6 | 73.5 | 78.2 | 13.2 |
| Corner [33] | 94.1 | 70.7 | 80.7 | 3.6 |
| TB [31] | 87.2 | 76.7 | 81.7 | 11.6 |
| RRD [31] | 85.6 | 79 | 82.2 | 6.5 |
| TextSnake [15] | 84.9 | 80.4 | 82.6 | 1.1 |
| PSE-1 s [16] | 86.9 | 84.5 | 85.7 | 1.6 |
| LOMO [27] | 91.3 | 83.5 | **87.2** | — |
| CDAFT [28] | 89.8 | 84.3 | 86.9 | — |
| SAE (720) [34] | 85.1 | 84.5 | 84.8 | 3 |
| SAE (990) [34] | 88.3 | 85 | 86.6 | — |
| DB-ResNet-18 (736) [1] | 86.8 | 78.4 | 82.3 | **48** |
| DB-ResNet-50 [1] | 91.8 | 82.3 | *87.3* | 12 |
| The proposed (Pr0.5) (736) | 86.4 | 79.5 | 82.8 | *51* |

decreasing the *F*-score and achieves 57 FPS, achieving the fastest speed compared to all other methods.

Table 4 shows the experimental results of the proposed method and the previous method on MSRA-TD500. DB-ResNet-18 (512) represents a short edge of the input image of 512, while DB-ResNet-18 (736) represents a short edge of the input image of 736. The proposed method comes in second with an *F*-score of 83.4%, just 1.5% behind the teacher network. In addition, the speed of the proposed method can reach 95 FPS, the fastest among all the methods and 13 FPS higher than the baseline.

Figure 1 depicts the visualization result of the combined *F*-score and speed FPS on MSRA-TD500. The proposed methods are located on the outermost side of the figure and away from their baseline, which means that these methods are provided with a strong competitive advantage over others in both speed and performance.

The proposed method is compared with 12 methods on the ICDAR2015 dataset. As shown in Table 5, the *F*-score of this method is 82.8, ranking the 8th, but is still competitive. This method improves the *F*-score of the benchmark by 0.5% and FPS by 3 and remains the fastest detector compared to other methods.

Table 6: Comparison results of the parameters size. The smallest size is highlighted in italic.

| Method | Backbone | Size |
|---|---|---|
| PSENet [16] | ResNet50 | 27.3 M |
| TextSnake [15] | VGG16 | 18.2 M |
| CRAFT [28] | VGG16 | 19.8 M |
| DBNet [1] | ResNet18 | 13.2 M |
| The proposed (Pr0.5) | ResNet18 | *6.8* M |

*4.5. Parameter Size.* The proposed method is compared with segmentation-based methods [35] including DBNet [1], PSENet [16], TextSnake [15], and CRAFT [28] in terms of the parameter size [36]. They are all source codes from the Internet and some are reimplement codes. The comparison

results are shown in Table 6, and the size of the proposed method is only half as large as that of DBNet, i.e., 6.8 M, the smallest among the five methods.

## 5. Conclusions

An efficient scene text detection method that balances speed and performance is hereby proposed. A pruning algorithm for text detection network is first introduced, which can simplify network parameters and speed up the computation. Furthermore, a structured knowledge distillation method is proposed to improve the detection performance of compact networks. These two features make the proposed method an agile and efficient arbitrary shape text detector. Besides, this method also presents competitive performance and fastest detection speed in three datasets.

## Data Availability

The English curve text dataset used to support the findings of this manuscript is a public benchmark dataset named Total-Text. Copies of these data can be obtained free of charge from https://github.com/cs-chan/Total-Text-Dataset. The multidirectional text dataset used to support the findings of this manuscript is a public benchmark dataset named ICDAR2015. Copies of these data can be obtained free of charge from https://pan.baidu.com/s/1pN-St9-aTpmxcHsgFXl8jA?pwd=i11a. The multilingual dataset containing both Chinese and English used to support the findings of this manuscript is a public benchmark dataset named MSRA-TD500. Copies of these data can be obtained free of charge from http://www.iapr-tc11.org/dataset/MSRA-TD500/MSRA-TD500.zip.

## Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

## Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province, China (Nos. 2019J01889 and 2020J018751); the "Tiancheng Huizhi" Innovation and Education Promotion Fund, China (No. 2018A02005); and the National Natural Science Foundation of China (No. 62172095).

## References

[1] W. H. Tok and S. Bressan, "DBNet: a service-oriented database architecture," in *17th International Workshop on Database and Expert Systems Applications (DEXA'06)*, pp. 727–731, Krakow, Poland, September 2006.

[2] W. Wang, E. Xie, X. Song et al., "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019, pp. 8440–8449, 2019.

[3] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017, pp. 2736–2744, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[5] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *European Conference on Computer Vision*, pp. 21–37, Springer, Cham, 2016.

[6] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: a fast text detector with a single deep neural network," *Thirty-First AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[7] M. Liao, B. Shi, and X. Bai, "TextBoxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 8, pp. 3676–3690, 2018.

[8] Y. Liu and L. Jin, "Deep matching prior network: toward tighter multi-oriented text detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017, pp. 1962–1969, 2017.

[9] X. Zhou, C. Yao, H. Wen et al., "EAST: an efficient and accurate scene text detector," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2017, pp. 5551–5560, 2017.

[10] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2017, pp. 2550–2558, 2017.

[11] J. Ma, W. Shao, H. Ye et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 11, no. 2018, pp. 3111–3122, 2018.

[12] Y. Zheng, Y. Xie, Y. Qu, X. Yang, C. Li, and Y. Zhang, "Scale robust deep oriented-text detection network," *Pattern Recognition*, vol. 102, article 107180, 2020.

[13] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: detecting scene text via instance segmentation," *Thirty-second AAAI Conference on Artificial Intelligence*, vol. 32, pp. 6773–6780, 2018.

[14] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes," *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 2018, pp. 67–83, 2018.

[15] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: a flexible representation for detecting text of arbitrary shapes," *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 2018, pp. 20–36, 2018.

[16] W. Wang, E. Xie, X. Li et al., "Shape robust text detection with progressive scale expansion network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 9336–9345, 2019.

[17] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: more deformable, better results," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 9308–9316, 2019.

[18] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, article 107343, 2021.

[19] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 106, pp. 2604–2613, 2019.

[20] X. Xue, P. Tsai, and Y. Zhuang, "Matching biomedical ontologies through adaptive multi-modal multi-objective evolutionary algorithm," *Biology*, vol. 10, no. 12, article 1287, 2021.

[21] C. K. Ch'ng and C. S. Chan, "Total-text: a comprehensive dataset for scene text detection and recognition," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 935–942, 2017.

[22] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou et al., "ICDAR 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1156–1160, Tunis, Tunisia, August 2015.

[23] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, Providence, RI, USA, June 2012.

[24] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *Image Processing IEEE Transactions*, vol. 11, pp. 4737–4749, 2014.

[25] X. Wang, Y. Jiang, Z. Luo, C. L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 6449–6458, 2019.

[26] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 11, no. 2019, pp. 5566–5579, 2019.

[27] C. Zhang, B. Liang, Z. Huang et al., "Look more than once: an accurate detector for text of arbitrary shapes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 10552–10561, 2019.

[28] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 9365–9374, 2019.

[29] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh, "Towards robust curve text detection with conditional spatial expansion," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 7269–7278, 2019.

[30] W. He, X. Y. Zhang, F. Yin, and C. L. Liu, "Deep direct regression for multi-oriented scene text detection," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017, pp. 745–753, 2017.

[31] M. Liao, Z. Zhu, B. Shi, G. S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, vol. 2018, pp. 5909–5918, 2018.

[32] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh, "Learning markov clustering networks for scene text detection," 2018, http://arxiv.org/abs/1805.08365.

[33] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2018, pp. 7553–7563, 2018.

[34] Z. Tian, M. Shu, P. Lyu et al., "Learning shape-aware embedding for scene text detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 4234–4243, 2019.

[35] Y. Yan, J. Ren, H. Zhao et al., "Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos," *Cognitive Computation*, vol. 10, no. 1, pp. 94–104, 2018.

[36] Y. Yan, J. Ren, G. Sun et al., "Unsupervised image saliency detection with gestalt-laws guided optimization and visual attention based refinement," *Pattern Recognition*, vol. 79, pp. 65–78, 2018.