TSFnet: a new two-stream fusion framework for scene text detection

Ziheng Zhou¹, Xuezhuan Zhao¹,³,a*, Lishen Pei²,³, Lao Li⁴, Jiahao Pan⁵

¹Zhengzhou University of Aeronautics, Zhengzhou, 450046, P. R. China
²Henan University of Economics and Law, Zhengzhou, 450046, P. R. China
³Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, 300300, P. R. China
⁴HeNan Radio & Television University, Zhengzhou 450046
⁵Wuhan University of Technology

azhaoxuezhuan@zua.edu.cn  *Corresponding author’s e-mail: xuezhuancv@126.com

Abstract. In order to solve the problem of algorithm robustness caused by scale change and imbalanced distribution of classes in the scene text detection task, we propose a new two-stream fusion framework TSFnet. It is constructed by the Detection Stream, the Judge Stream and the Merge output algorithm. In the Detection Stream, we propose a loss balance factor (LBF), which is used to optimize region proposal network. Then, the Regression-net and the Segmentation-net are used to predict text global segmentation map and its corresponding coordinates probability score. In Judge Stream, we use the feature pyramid network to extract the Judge map. In the process, the LBF is calculated to support the Detection Stream. Finally, we design a novel algorithm to fuse the outputs of the two-stream, and the precise position of the text is localized. Extensive experiments are conducted on the ICDAR 2015 and the ICDAR2017-MLT standard datasets. The results demonstrate that the framework performance is comparable with the state-of-the-art approaches.

1. Introduction

Scene text detection is an important research direction in the field of computer vision. The goal of this task is to locate the position of the text in the image captured in the natural environment. The task has a wide range of applications, such as geo-location, automatic driving, real-time translation and other fields.

In recent years, most scene text detection models are based on depth neural network. These models can be classified into two categories: regression-based model [1-5] and segmentation-based model [6-9]. Regression-based model obtains scene text position through coordinates of relevant regions. Its performance is very sensitive to the distribution of training data. Unbalanced distribution of training data categories leads to poor robustness of the algorithm. In the segmentation-based model, the text location task is regarded as the instance segmentation task, and convolution is used to obtain the segmentation map. The large change of the target scale in natural scenes is one of the key factors that affect the robustness of the algorithm. At present, in the process of algorithm training and testing, the above two problems need to be faced directly.
To solve above problems, we propose a new framework which includes two streams: Detection Stream and Judge Stream, as shown in Figure 1. In the Detection Stream, we propose to combine the regression-based model and segmentation-based model to make up for their shortcomings. And we propose a loss balance factor (LBF) to support the optimization of the detection stream. In the Judge Stream, we use the feature pyramid network to extract the Judge map. The LBF is calculated in this process. Finally, we design a novel algorithm to fuse the outputs of the two-stream. Experimental results show that the new framework effectively solves the problems.

The contributions of this paper are multifold. First, we propose a new framework, which can detect multi-directional, multi-shape and multi-language text. Second, we propose a new concept: loss balance factor, which is used to solve the imbalance problem caused by training samples. At last, we propose a Merge output algorithm to fuse the results of the two streams. As a result, the robustness of the algorithm is improved.

2. RELATED WORK

In recent years, a large number of high-level papers have been published. According to the relevance of this paper, it mainly introduces the related research work from two aspects: Regression-based method and Segmentation-based method.

**Regression-based methods.** Regression-based methods often based on general object detection frameworks. Zhou et al. [1] proposed score map which get through full convolutional network to predict text region. Liao et al. [3] present oriented convolution and oriented response pooling to get text region coordinates. In FOTs [22], traditional classification loss function and regression loss function have excellent performance in scene text location task. FSTN [5] introduced two-stage detection module and combine a text instance segmentation module to detect multi-oriented text.

Most of the regression-based methods design complex anchors and cumbersome pipelines, which make them very sensitive to the environment. Through experiments, it is found that the imbalanced distribution of training data categories will lead to the decrease of recall rate.

**Segmentation-based methods.** Segmentation-based methods regard the problem of text detection as the semantic segmentation of text regions. In Seglink [6], the coordinates of the entire text area are obtained through full convolution network (FCN) [7] and text link area. In PixelLink [8], the authors...
integrate multi-layer deep features to improve detection results. After the emergence of Mask RCNN, some two-stage methods combined with segmentation model. MaskSpotter [9] is a two-stage method which can detect arbitrary shape text and achieve text mask segmentation.

Compared with the regression-based method, the above methods utilized more semantic information. Most of the Segmentation-based methods are robust to the environment. However, the effect needs to be improved for the large scale changes samples.

Inspired by the above models, we propose a new two-stream fusion framework. The framework combines the advantages of the two algorithms and proposes a new concept to solve the problem of algorithm robustness caused by scale changes and imbalanced class distribution.

3. TSFnet

In this section, we describe the details of the TSFnet. As shown in Figure 1, the whole framework consists of three parts: Detection Stream, Judge Stream and Merge output algorithm. In Detection Stream, a backbone network and a region proposal network are introduced to get the feature maps and the text candidate regions. And then, a Regression-net is used to predict the text coordinates and probability score, and a Segmentation-net is used to predict text global segmentation map. In Judge Stream, a feature pyramid network is adopted to extract the Judge map. In this process, a loss balance factor is calculated to optimize the detection of the stream network. Finally, we propose a new merge output algorithm to fuse the two stream results to achieve precise positioning of the text.

The corresponding training strategy is designed for the framework. Judge Stream and Detection Stream are trained in parallel in the first stage. And then, in the second stage, the loss balance factor (LBF) is calculated in Judge Stream, the factor is used to help Detection stream training region proposal network (RPN). Finally, the Detection Stream continues training until accomplish whole training process.

3.1. Detection Stream

The Detection Stream realizes the integration of the traditional network and the optimization of the network. First, ResNet50 [10] is used as the backbone network to extract the different size feature maps. Then, RPN is introduced to obtain the text candidate regions based on the feature maps. At last, Faster R-CNN [11] is used as the Regression-net to predict the text coordinates and probability score. Mask R-CNN [12] is used as Segmentation-net to predict text global segmentation map.

Inspired by Faster R-CNN, RPN consist of a convolutional layer and two fully connection layers. The probability loss functions of our RPN are redesigned for the new framework. There are two different loss functions correspond to two training stages respectively.

In the first stage, the loss function is binary cross-entropy loss, which show as follows:

$$L = \frac{1}{N} \sum_{i=1}^{N} [y_i \times \ln p_i + (1 - y_i) \times \ln(1 - p_i)]$$

(1)

where is the output, is the label of . If it is a negative sample, the value is 0, otherwise it is 1. In the second stage, the loss function is as follows:

$$L_{rpn} = W_p \times L_p + W_N \times L_N$$

(2)

$$W_p = \frac{N_M}{N}$$

(3)

$$W_N = \frac{N_P}{N}$$

(4)

Where is the total loss function value, is the value of the loss function for positive samples, is the value of the loss function for negative samples. means false positive weight, means
false negative weight. \( N \) is the number of samples for back propagation. \( N_p \) is the number of positive samples, \( N_N \) is the number of negative samples. The \( L_p \) calculation is as follows:

\[
L_p = \lambda \times L_{NT} + (1 - \lambda) \times L_{NF}
\]  

(5)

Where \( L_{NT} \) is the loss value of the positive samples matching of text region that detect by Judge Stream, \( L_{NF} \) is the loss value of the positive samples matching of text region that Judge Stream miss to detect. Both are calculated according to formula (1). \( \lambda \) is loss balance factor, which is calculated from the Judge Stream.

![Figure 2. The Judge Map label and Judge Map.](image)

In the second stage of training, loss balance factor is introduced into the model. By adjusting the loss function through the balance factor, the model adapts to the change of the training data category distribution. After this process, the detection flow will pay more attention to the targets that are difficult to detect by the judgment flow, so that the detection flow and the judgment flow can achieve cooperation in the detection task.

3.2. Judge Stream

As shown in figure 1, inspired by ResNet50, the feature pyramid network is used to extract the different size feature maps. It can increase the receptive field of the convolution kernel without changing the scale of the convolution kernel. After the processing of this stream, two main outputs can be obtained: the Judge Map and the loss balance factor. The Judge map is used to combine the Detection map into a result map. The loss balance factor is used to optimize the detection stream process.

In the framework, Judge Stream is needed to enhance the detection performance of small targets. Inspire by [16], the dice loss is used as loss function which is as follows:

\[
L_{\text{Judge map}} = 1 - \frac{2 \sum_{x,y}(J_{x,y} \times Y_{x,y})}{\sum_{x,y} J_{x,y} + \sum_{x,y} Y_{x,y}}
\]  

(6)

Where \( J \) is Judge Map, the output of Judge Stream. \( Y \) is Judge Map label which is a global text instance segmentation binary map. It’s generated by drawing all text polygons on a zero-initialized mask and filling the polygon region with the value 1. \( J_{x,y} \) and \( Y_{x,y} \) refer to the value of pixel \((x, y)\) in \( J \) and \( Y \), respectively.

**The loss balance factor** is calculated as follows:
\[
\lambda = \max \left( \frac{1}{n} \sum_{i=1}^{n} q_i, 1 - \frac{1}{n} \sum_{i=1}^{n} q_i \right)
\]  

(7)

\[
q_i = \begin{cases} 
0 & \text{if } \alpha_i < \text{Threshold} \\
1 & \text{if } \alpha_i \geq \text{Threshold} 
\end{cases} \quad (i = 1, 2, \ldots, n)
\]  

(8)

\[
\alpha_i = \frac{\sum_{p=0}^{H_i-1} \sum_{q=0}^{W_i-1} y_i(p, q) \times l_i(p, q)}{\sum_{p=0}^{H_i-1} \sum_{q=0}^{W_i-1} y_i(p, q)} \quad (i = 1, 2, \ldots, n)
\]  

(9)

Where \( \lambda \) is the loss balance factor. \( n \) is the number of all foreground region in Judge Map label. \( q_i \) is foreground region category label, if \( q_i \) corresponding to one text region in the label equal to 1, the text region is considered to be detected by Judge Stream otherwise it is not detected. \( \alpha_i \) is overlap rate. \( Y = \{y_1, y_2, \ldots, y_n\} \) is bounding rectangle region set which obtain by extracting the bounding rectangle regions of the foreground in Judge Map label, as show in figure 2. In figure 2, the yellow dotted frames is the border of the bounding rectangle regions. These can get by the coordinates of text regions in data labels. Extract the same size rectangle region from the same position in the Judge Map and get \( L = \{l_1, l_2, \ldots, l_n\} \), \( H_i \) is high and \( W_i \) is width of \( y_i \). \( y_i(p, q) \) and \( l_i(p, q) \) represent the value of pixel \((x, y)\) in \( y_i \) and \( l_i \), respectively.

3.3. Merge output algorithm

As show in figure 3, the merge output algorithm is proposed to fuse the results of Detection Stream and Judge Stream. This process is mainly based on global segmentation map, Judge Map, probability scores, and overlap rates. Detection Stream prone to false positives when detecting small targets, therefore, we use Judge Map to strengthen the small targets detection result output by Detection Stream.

![Figure 3. The whole process of merge output algorithm.](image)

The specific process is described as follows: First, we perform Low probability area suppression. In this stage, we delete small area prediction results that have a low overlap with the Judge map, and large area prediction results with probability score below the threshold. Second, we perform Overlap rectangle suppression. At this stage, some overlapped prediction results can be delete by the algorithm. Finally, we get the output result, Final Map.

The specific calculation process is as follows:

\[
S_{out} = \sum_{i=1}^{n} s_i
\]  

(10)
\[ s_i = \begin{cases} 
  c_i & \text{if } \delta_i \geq T_1 \text{, } f(c_i) \leq T_3 \\
  \text{zeros}(H,W) & \text{if } \delta_i < T_1 \text{, } f(c_i) \leq T_3 \\
  p_i \geq T_2 \text{, } f(c_i) > T_3 \\
  & \text{or } p_i < T_2 \text{, } f(c_i) > T_3 
\end{cases} \quad (i = 1,2, \ldots, n) \quad (11) \]

\[ \delta_i = \frac{\sum_{p=0}^{H-1} \sum_{q=0}^{W-1} c_i(p,q) \times j(p,q)}{\sum_{p=0}^{H-1} \sum_{q=0}^{W-1} c_i(p,q)} \quad (i = 1,2, \ldots, n) \quad (12) \]

Define global segmentation maps set \( C = \{c_1, c_2, \ldots, c_n\} \) and probability score set \( P = \{p_1, p_2, \ldots, p_n\} \), these are the outputs of Detection Stream, \( c_i \) is a binary map. \( f \) is a function for calculating the foreground area of input. \( j \) is Judge Map which is the outputs of Judge Stream. \( c_i(p,q) \) and \( j(p,q) \) represent the value of pixel \((p, q)\) in \( c_i \) and \( j \), respectively. \( H \) and \( W \) represent the width and height of the map, respectively. \( T_1, T_2 \) and \( T_3 \) represent the corresponding thresholds, respectively. Then, \( S_{\text{out}} \) is obtained, we normalize all pixel values greater than 1 in \( S_{\text{out}} \) to 1 and get Final Map. Final Map is a binary map, if one pixel is predict to belong to text region, it’s value equal to 1, otherwise it is equal to 0.

Through the fusion process, the false-positive results of detection stream output are suppressed, and the problem of region overlap is solved. Finally, the algorithm can be applied to scenarios with large-scale changes and imbalanced distribution of categories.

4. Experiment

In this section, we compare our method with several state-of-the-art methods. In order to prove the effectiveness of our framework, we carried out experiments on two common datasets, ICDAR 2015 and ICDAR 2017 MLT, and analysed the experimental results.

In the training stages, we optimize our model using SGD [19]. The batch sizes of the RPN and Regression-net are set to 256 and 512, respectively. The batch size of the Segmentation-net is 16. In the first stage, Detection Stream train on ICDAR2017 [14] and ICDAR2015 [13] for 300K iterations, the initial learning rate is set to 0.005 and momentum is set to 0.9, the weight decay is 0.0001. Learning rate decreased to 0.0001 at the 160k iteration. Judge Stream train on ICDAR2017 or ICDAR2015 for 180K iterations and initial learning rate is set to 0.001. We use a weight decay of 0.0005 and set momentum of 0.99 without dampening. In the second stages, the Detection stream trains 300K iterations on the same dataset as Judge Stream. The training strategy the same as first stage. In the test stage, batch-size is set to 1. The number of proposal regions is set to 1000.

In our experiment, threshold set to 0.8. In the test stage, equal to 0.8, equal to 0.5, equal to 2000. In table 1 and 2, P is precision, R is recall, F is F-score, ICDAR 2015

ICDAR 2015 (IC15) [13] contains 1500 pictures, 1000 of which are for training and the rest for testing. The text regions are annotated by 4 vertices of the quadrangle. These pictures were taken indoors, on the street, and in different environments. Therefore, this dataset contains a large number of scenes, and some samples contain noise caused by motion blur and lighting.

We compare the performance of our work with 9 state-of-the-art methods on the IC15 dataset. To evaluate the performance of the self-adaptive loss balance factor, we test the effect with and without the self-adaptive loss balance factor separately.

The statistical result shows that the model has advantages over other algorithms, especially the values of F-measure and recall. In addition, the proposed adaptive balance factor has great practical value. As show in figure 4, our model improves the F-score and robust to multi-scale rectangular text in complex scenes. Its significance lies in improving the robustness of the algorithm.
Figure 4. The partial detection results on ICDAR2015

Table 1. The result on ICDAR2015

| Method            | P   | R   | F   |
|-------------------|-----|-----|-----|
| Seglink [6]       | 73.1% | 76.8% | 75.0% |
| EAST [1]          | 83.5% | 73.4% | 78.2% |
| Wordsup [4]       | 79.3% | 77.0% | 78.2% |
| SSTD [2]          | 80.0% | 73.0% | 77.0% |
| FSTN [5]          | 88.6% | 80.0% | 84.1% |
| Boundary[21]      | 85.2% | 83.5% | 84.3% |
| PixelLink[8]      | 82.9% | 81.7% | 82.3% |
| MaskSpotter[9]    | 85.8% | 81.2% | 83.4% |
| CRAFT[20]         | 89.8% | 84.3% | 86.9% |
| TSFnet            | **90.0%** | 82.2% | 85.9% |
| TSFnet + λ        | 88.2% | **87.4%** | **87.8%** |

4. ICDAR 2017 MLT

ICDAR 2017 MLT (IC17-MLT) [14] is a lamprey scale multi-lingual text dataset, which includes 7200 training images, 1800 validation images and 9000 testing images. It contains a variety of pictures of different sizes, and includes more natural scenes, such as roads, forests, etc., and the proportion of text areas in the pictures is also different.

We compare the performance of our work with 6 state-of-the-art methods on the IC17-MLT dataset. Similar to the IC15, The effects with and without the self-adaptive loss balance factor are shown in Table.

Figure 5. The partial detection results on ICDAR2017MLT

Table 2. The result on ICDAR2017MLT

| Method                      | P     | R     | F     |
|-----------------------------|-------|-------|-------|
| TextSnake [17]              | 28.99% | 16.85% | 21.31% |
| Linkage-ER-flow [15]        | 44.48% | 25.59% | 32.49% |
| YY AI OCR Group [15]        | 52.60% | 64.77% | 44.28% |
| TH-DL [15]                  | 67.75% | 34.78% | 45.97% |
As shown in Table 2, our algorithm gets acceptable results. Although precision is reduced a little, the F-score results have improved significantly. As shown in figure 5, compared with other methods, our model is more robust in complex scenarios. In this experiment, the balance factor has a more significant effect on the optimization of the results. In figure 5, we present partial detection results of our algorithm. All of the source images are selected from the IC17-MLT datasets.

5. Conclusion
We propose TSFnet for scene text detection. It is used to solve the problem of algorithm robustness caused by scale change and imbalanced distribution of classes. Experimental results show that the two-stream model can handle text detection tasks in complex scenes. The performance of our algorithm is comparable with the state-of-the-art works. Meanwhile, there are several respects of this work need to be improved. For example, the execution efficiency of the algorithm needs to be improved. As a novel framework, this work is the beginning. We will continue to improve this work.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China (61806073, U1904119, 31700858) and the Research Programs of Henan Science and Technology Department (192102210097, 192102210126, 192102210269) and the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (NO. CAAC-ITRB-201607) and the Key scientific research projects of colleges and universities in Henan Province (18A520050).

References
[1] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: an efficient and accurate scene text detector. In: Proc. CVPR. pp. 2642–2651 (2017)
[2] He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X.: Single shot text detector with regional attention. In: Proc. ICCV. pp. 3066–3074 (2017)
[3] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In CVPR (2018)
[4] Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., Ding, E.: Wordsup: Exploiting word annotations for character based text detection. In: Proc. ICCV. pp. 4950–4959 (2017)
[5] Y. Dai, H. Zheng, Y. Gao, C. Kai, Fused text segmentation networks for multi-oriented scene text detection, in: International Conference on Pattern Recognition, pp. 3604–3609 (2018)
[6] B. Shi, B. Xiang, S. Belongie, Detecting oriented text in natural images by linking segments, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3482–3490.
[7] Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(4), 640–651 (2017)
[8] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI (2018)
[9] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, & Xiang Bai. Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(99) (2019)
[10] Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, & Alemi, Alex. Inception-v4, inception-resnet and the impact of residual connections on learning. In CVPR (2016)
[11] Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39(6), 1137–1149 (2017)
[12] He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: Proc. ICCV. pp. 2980–2988 (2017)
[13] He, W., Zhang, X., Yin, F., Liu, C.: Deep direct regression for multi-oriented scene text detection. In: Proc. ICCV. pp. 745–753 (2017)
[13] Karatzas D, Gomez-Bigorda L, Nicolaou A, et al. ICDAR 2015 competition on Robust Reading[C]// 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE Computer Society, (2015)
[14] Pratikakis I, Zagoris K, Barlas G, et al. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017) [C]// 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE Computer Society, (2017)
[15] ICDAR2017 Competition on Multi-Lingual Scene Text Detection and Script Identification. http://rrc.cvc.uab.es/?ch=8&com=introduction, (2017)
[16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In IC3DV, 2016.
[17] Long S, Ruan J, Zhang W, et al. TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes[J]. 2018.
[18] Liu Y, Jin L. Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection[J]. 2017.
[19] Léon Bottou. Stochastic Gradient Descent Tricks[M]// Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, 2012.
[20] Baek Y, Lee B, Han D, et al. Character Region Awareness for Text Detection[J]. 2019.
[21] Wang H, Lu P, Zhang H, et al. All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting[C]. national conference on artificial intelligence, 2020.
[22] Liu X, Liang D, Yan S, et al. FOTS: Fast Oriented Text Spotting with a Unified Network. In CVPR (2018)