BorderNet: An Efficient Border-Attention Text Detector

Juntao Cheng[1]* (✉), Liangru Xie[1]*, and Cheng Du[1]

AI R&D Department, Kingsoft Office, China
{chengjuntao1,xieliangru,ducheng}@wps.cn

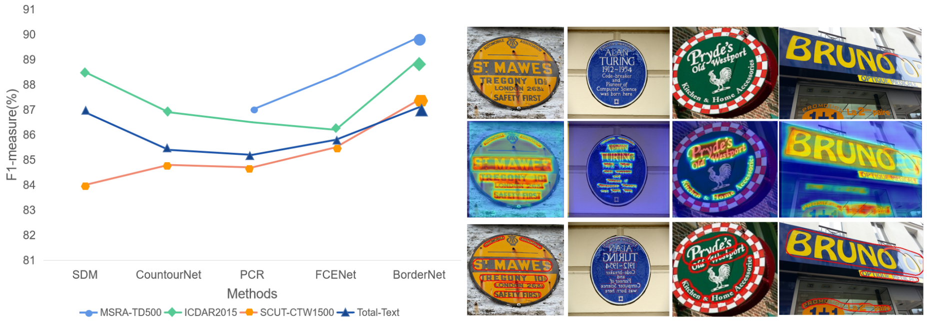
Abstract. Recently, segmentation-based text detection methods are quite popular in the scene text detection field, because of their superiority for text instances with arbitrary shapes and extreme aspect ratios. However, the vast majority of the existing segmentation-based methods are difficult to detect curved and dense text instances due to principle of these methods. In this paper, we propose a novel text detection method named BorderNet. The key idea of BorderNet is making full use of border-center information to detect the curve and dense text. Furthermore, a efficient Multi-Scale Feature Enhancement Module is proposed to improve the scale and shape robustness by enhancing features of different scales adaptively. Our method outperforms SOTA on multiple datasets, achieving 89% accuracy on ICDAR2015 and 87.1% accuracy on Total-Text. What's more, we can maintain 84.5% accuracy on DAST1500.

1 Introduction

Scene texts often appear in a variety of application , and provide a wealth of important information, such as intelligent office, visual search, scene understanding, automatic driving and other application directions. Therefore, reading scene text images is extremely important. Text detection that locates the text position is a very important part of reading text. Scene text detection faces greater challenges than general object detection owing to the extreme aspect ratio of the text, irregular shapes, different scales and other factors.

Owing to the development of object detection and segmentation based on deep learning in recent years, scene text detection has made great progress. Scene text detection can be roughly divided into three categories: Regression-based methods, Segmentation-based methods and Hybrid methods. Hybrid methods merge the advantages of segmentation and regression so that they complement each other. Regression-based methods and some hybrid methods can achieve excellent performance on benchmark testsets. However, they have a huge bottleneck which assume text instances have a linear shape. Hence, horizontal or multi-oriented quadrilaterals are used to represent text boxes. In addition, their performance in detecting text with irregular shapes such as curved and dense

* Authors contribute equally.



(a) Comparison with SOTA (b) Examples of BorderNet

Fig. 1: (a) The figure shows the comparison with recent SOTA methods such as SDM[28], CountourNet[26], PCR[3] and FCENet[33] on various benchmark datasets, in terms of accuracy (F-measure). Our proposed BorderNet can achieve higher accuracy. (b) The figure visualizes the entire inference process of BorderNet. The first row is the input image, the second row is the heatmap during network inference, and the third row is the final text detection result.

text drops significantly. In contrast, segmentation-based scene text detectors generally have an advantage in detecting text instances with irregular shapes and extreme aspect ratios due to their pixel-level representation and local prediction. Although segmentation-based methods can accurately predict text regions, it is difficult to separate close text instances. Many recent segmentation-based methods focus on how to separate the segmented text region into multiple text instances, such as SOTD[27], TextField[29], TextMountain[34]. Furthermore, most segmentation-based methods require complex post-processing to aggregate text regions, resulting in considerable time consumption. For example, PSENet[23] proposes a progressive scale expansion algorithm for post-processing, which integrates feature maps of multiple scales, resulting in a large time cost. In addition, although most segmentation-based methods using feature pyramids[10] or UNet[20] structure fuse multi-scale features to get higher accuracy, there is a semantic gap between different layers. Therefore, forcibly merging features of different scales will reduce the ability of multi-scale feature expression and cause feature redundancy.

The detection of curved and dense text quickly and accurately still faces severe challenges. The border of curved and dense text are easy to stick, which is difficult to distinguish by common methods. Therefore, this study proposes a text detector based on border learning (BorderNet) for curved and dense text. Our approach proposes to directly use image segmentation methods to learn the center and border regions of text, and use border regions to separate different text instances. In order to improve the efficiency and accuracy of the algorithm, based on DBNet[7] and DBNet++[9], we integrate Differentiable Binarization (DB) into boundary learning, which greatly reduces the number of calculations used for post-processing. To fuse multi-scale features more fully, we propose a multi-scale feature enhancement module (MSFEM), which enhances the features of each scale to make it more fully fused. MSFEM contains two feature fusions, one occurs when deep layers are fused with shallow layers gradually, and the other occurs when shallow layers are fused with deep layers gradually. In the fusion process, the multi-scale features with stronger semantics can be obtained by introducing the multi-head channel attention module (MHCA) which enhances the features of each layer. By directly fusing the enhanced multi-scale features, the center and border regions of the text can be better segmented. As shown in Fig 2 below, our method can detect curved and dense text better than other methods. Our method surpasses SOTA methods on various benchmark datasets such as ICDAR2015 and Total-Text. Our method reaches 89% on the ICDAR2015 and 87.1% on the Total-Text. It is evident from the experiments that by introducing border learning and fully fusing multi-scale features, the model has higher accuracy, stronger robustness, and better efficiency.

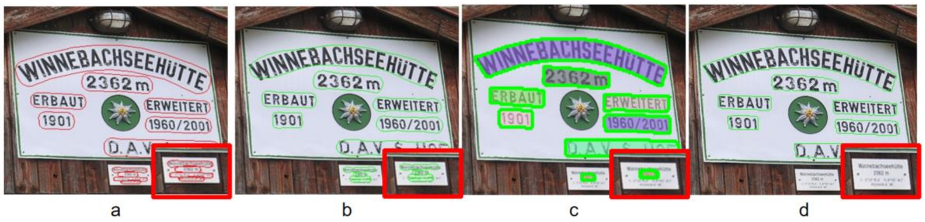


Fig. 2: Comparison with other methods. Fig a is our method, fig b is result from FCENet[33], fig c is result from FAST[2], fig d is result from PSENet[23].

The main contributions of this study are as follows: 1) We propose a text detector based on border learning, which directly uses the image segmentation method to learn the center and border regions of the text, and uses the border regions to separate different text instances which can solve the challenge of curved and dense text. 2) In order to detect scene text more accurately and faster, we proposed MSFEM, which can enhance multiple features of different scales to make feature fusion more fully and obtain higher semantic information.

2 Related Work

Scene text detection has received extensive attention recently, and many new methods have emerged. These methods can be roughly divided into three categories: regression-based methods, segmentation-based methods, and hybrid methods.

Regression-based methods Most of regression-based methods directly regress the text instance boxes. CTPN[21] divides the text instance into many small vertical text regions, regresses each text box directly, and then connects text boxes using a recurrent neural network. However, the regression anchor of CTPN[21] is vertical, and this method can only detect horizontal or slightly multi-oriented text. To solve this problem, RRPN[17] basing on the framework of Faster RCNN[19] integrates the rotated anchor to achieve the effect of detecting multi-oriented text boxes. However, this method uses a rectangular text box, which makes it difficult for text box to get close to the border of multi-oriented text. Then DMPNet[11] is proposed, which regresses the quadrilateral directly and the text box can be positioned close to the border of multi-oriented text. Although the development of regression-based methods has matured, and has good effects on horizontal and multi-oriented texts, there are still great challenges in the case of curved and dense text.

Segmentation-based method The segmentation-based method regards text detection as a text region segmentation problem, and usually requires pixel-level prediction and post-processing algorithms to obtain the text box. The border position of the text can be found, and distinguishing between text regions and non-text regions is very important in this type of method. SOTD[27] introduces the text border learning method, which uses the text border region as the third category for semantic segmentation. But there has a strong dependence on feature extraction ability of model. TextField[29] and TextMountain[34] achieve the purpose of detecting dense text by modeling the text center and text border. However, this requires complex modeling and post-processing, which is a time-consuming process. Although segmentation-based methods can detect horizontal, multi-oriented and curved texts well, dealing with dense texts quickly and efficiently still faces considerable challenges.

Hybrid methods It is also worth mentioning that some other methods utilize segmentation to classify text/non-text pixels and then localize the text via bounding box regression. For example, East[32] and Deep Regression[6] predict a rotated rectangle or quadrilateral for each pixel, but can not solve the detection problem on curved text. Mask TextSpotter[16] uses instance segmentation methods for detection, which can detect curved text. But it does not perform well on both curved and dense text, whihc have time consuming and low accuracy. Other hybrid methods[31] require complex and time-consuming post-processing[18]to remove duplicate prediction boxes.

In text detection, how to learn text borders quickly, efficiently and precisely is very important for detecting curved and dense text. Most of the above methods have relatively complex post-processing, which will bring a large time cost. Therefore, we propose a text detection method based on border learning and design a more concise framework. Meanwhile, we introduce a feature Fusion module, which can simplify post-processing and enable the network to learn where the text region is quickly and precisely.

3 Methodology

3.1 Overall Architecture

Based on the fully convolutional network[14], we design the overall architecture in the form of an Encoder-Decoder. The specific structure includes a feature extraction module, MSFEM and Decoder module. In the feature extraction module, we use the ConvNeXt[12]as the backbone to learn the features of text images. Because the size distribution of text may vary greatly, it is difficult for single-layer features to adapt to texts of different scales. Therefore, MSFEM is introduced to fuse multi-scale features by means of feature pyramids. To improving the fusion quality of MSFEM, we design a module named MHCA. The features of different scales of stage 2, stage 3, stage 4 and stage 5 are fed into MSFEM to obtain multiple enhanced scale features through MHCA, and then concat them together and fully fuse them through a layer of convolution to obtain multi-scale fusion features with stronger semantics. Finally, the obtained multi-scale fusion features are fed into Decoder module. Decoder module first predicts the shrinkage probability map for the center region of the text and applies border learning to obtain the probability map for the text border region, and gets the threshold map corresponding to the predicted probability map. Afterwards,the DB module is introduced to optimize the learning process, and obtains the final result by fusing the probability map and the threshold map.Fig 3 shows the overall network of BorderNet.

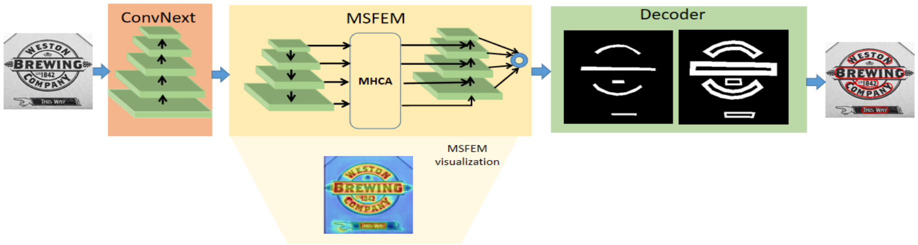


Fig. 3: BorderNet network structure.

3.2 MSFEM

In the process of extracting text image features, we found that features of different scales have different receptive fields and can pay attention to texts of different scales. Shallow features have smaller receptive fields and can perceive the details of small texts, but can not capture global information. Deep features have a large receptive field and can capture global information, but can not perceive detailed information. Therefore, for dense text detection, we need to capture the feature information of each scale to the greatest extent, and fully integrate the features of different scales. To achieve this purpose, we introduce MSFEM.

We first build MSFEM based on the feature pyramid. In this module, the feature pyramid gradually upsample deep features and fuse with shallow features. To fully fuse and amplify the feature information extracted from each scale to speed up the learning efficiency of the network, the fused features obtained from each scale are passed through MHCA to learn the attention weight, and multiplied by the original scale feature to obtain the filtered fusion feature. Therefore, to extract useful multi-scale feature information to a greater extent, the filtered fusion features from each scale are gradually down-sampled and fused with the deep features.The operation of fused refers to concat&conv. The particular network structure of MSFEM is shown in Fig 4.

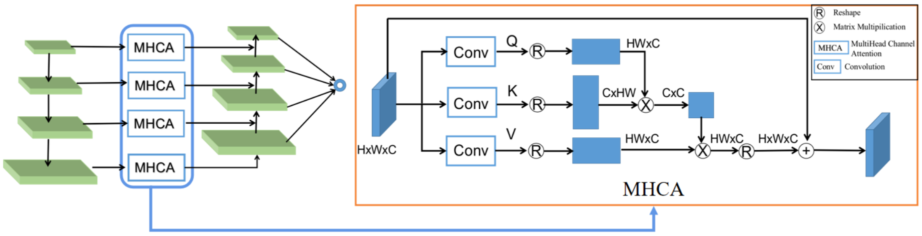


Fig. 4: MSFEM network structure.

By visualizing the feature map obtained by the last convolutional layer of MSFEM (as shown in Fig 4), we can see that after MSFEM, the network can focus better on the text region. The network can also learn well on curved and dense text.

MHCA MSFEM is designed to combine features of multiple scales, which inevitably leads to redundancy in features, which limits the network's ability to locate object rapidly and accurately. Introducing an attention mechanism can solve this problem. As some methods proved that transformer is a successful module based on attention mechanism, where multi-head self-attention (MHSA) can adaptively learn more types of features and filter key features in the spatial dimension. However, the computational time of MHSA is positively correlated with the required memory and input resolution, which is very resource-intensive. Inspired by Restormer[30], we propose the MHCA module, which is a multi-head channel attention mechanism that can expand the feature learning space, learn more feature types, and enhance the expressive ability of the model. Moreover, the MHCA module is an attention module with linear complexity. It builds the attention map not in the spatial dimension but in the feature channel dimension. Hence, MHCA is not affected by the input resolution, and the computational cost is greatly reduced.We use gradcam to visualize the feature map obtained by the last convolution layer of MSFEM, as shown in Fig 5.

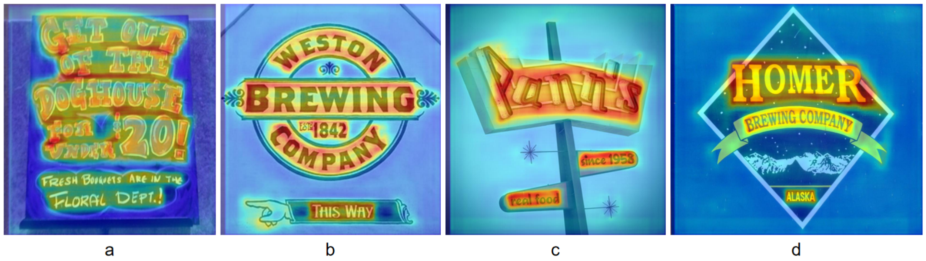


Fig. 5: Visualize the feature map obtained by the last convolutional layer of MSFEM.Fig a, b, c and d are samples.

The MHCA module first uses a 1x1 convolution on the input to communicate the information between each channel, and then a 3x3 depthwise separable convolution to obtain spatial local context information on each channel. Then, query (Q), key (K) and value (V) are generated from the obtained features. Multi-head of MHCA is achieved by dividing it into multiple heads based on the number of channels. MHCA changes the dimensions of Q and K, and then dot-multiply each other to form a channel-based attention map. After that, the attention map performs dot multiplication with V to determine the attention feature for the current head. Finally, attention feature maps of multiple heads are fused to obtain the module output. The calculation process of attention can be expressed as:

$$Attention(Q, K, V) = V \cdot Softmax(\frac{K \cdot Q}{\alpha}) \tag{1}$$

where α is a learnable scale parameter to control the weights of the dot product of K and Q.

3.3 Decoder

In this section, we design two key modules, including Convolution Map(CM) module and Fusion module. The main function of the CM module is to map the feature map obtained in the previous part to a single-channel grayscale image with a value range of [0,1]. We designed the CM module consist of a convolution and two deconvolutions.The convolution part is responsible for feature mapping, the deconvolution part is responsible for restoring the image size without losing effective information, and finally obtaining the corresponding grayscale image.

The Fusion module is used to fuse the multiple grayscale images through the DB module to get the final result. The Fusion module is used for training to improve the accuracy of the network and can be unused during inference phase to reduce inference time and memory consumption.

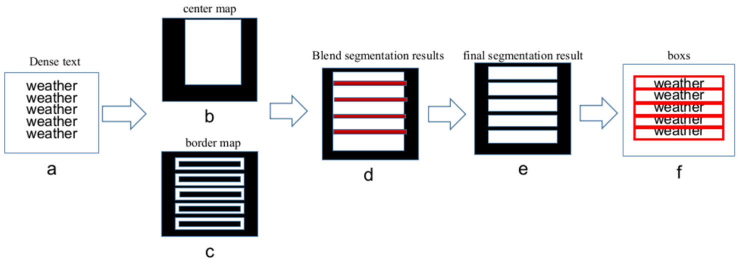


Fig. 6: The fusion process of border map and center map. Fig a is the dense text to be detected, fig b and fig c are the center map corresponding to the text center region and the border map corresponding to the text edge region. Then, the two maps are mixed to obtain fig e as shown in fig d, and finally the boxes of dense text in fig f are obtained based on fig e.

The decoder uses the feature map from the MSFEM to predict the approximate position of the text. The feature maps are respectively created by two CM modules to create center map and border map, which are probability maps for predicting the text center and border regions, and then the probability maps are fed into the Fusion module. In the Fusion module, we first map the feature to the threshold map through the CM module. The threshold map obtained at this time corresponds to the probability map obtained earlier. The center map corresponds to the threshold map of the text border region. The border map corresponds to the threshold map of the text center region. The corresponding probability map and threshold map are learned by DB optimization. Finally, the results of the optimized learning are integrated to get the text detection results.

In this section, we not only retain the commonly used center map for the center region of the text, but also design border map for learning the text border to enhance the accuracy of text border learning. Because center map is fused with border map, the detection results of dense text can be well separated, improving the accuracy.The whole process is shown in Fig 6.

In addition to learning the center and border regions of the text image, the model also learns the center point of the text as well as the region of the text border, to estimate the size and location of the text more accurately. The design is shown in Fig 7.

Label Generation A total of 4 kinds of labels are required in our network design, namely the label corresponding to the border map, the label corresponding

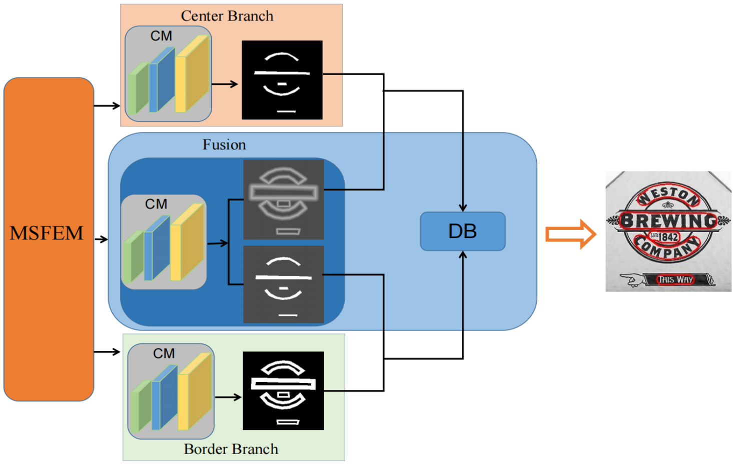


Fig. 7: Decoder network structure.

to the center map, the threshold map label corresponding to the border map, and the threshold map label corresponding to the center map.

The label of the border map mainly describes the border of the text. Given an image of text, each polygonal region of its text is described by a set of line segments, as shown in the following formula:

$$G = \{S_k\}_{k=1}^{n} \tag{2}$$

where n is the number of vertices. Next, the polygon G is reduced to G1 by using the Vatti clipping algorithm [22]. The positive sample region obtained by subtracting G1 from G is the border region of the text. The shrinking offset D is usually calculated from the perimeter and region of the original polygon, which can be expressed as:

$$D = \frac{A(1-r^2)}{L} \tag{3}$$

where r is the shrinkage ratio, which is empirically set to 0.4.

The label of the center map is used to describe the center region of the text in the image. The positive sample region generated by reducing the above-mentioned original polygon G to polygon G1 is the center region of the text. This process draws on the generation method of the probability map of PSENet[23] and DBNet[7].

The threshold map label corresponding to the border map is mainly aimed at the center region of the text. Hence, the generation process is similar to the label of the center map. After subtracting the reduced polygon G1 from the original polygon G to obtain the positive sample region, the region is dilated to obtain the edge of center region in text. Based on the edge, it assigns values to pixels from far to near, and the farther away from the edge, the smaller the value of the pixel.

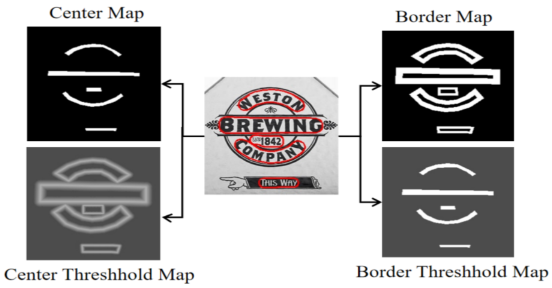


Fig. 8: Label generation.

The threshold map label corresponding to the center map is mainly aimed at the text border region, so the generation process is similar to the label of the border map. We set the pixel value in the positive sample region that is obtained after reducing the polygon to 1, and other regions are set to 0.3. The schematic diagram of the various labels generated is shown in Fig 8.

4 Experiments

4.1 Datasets

In our experiments, we first pre-train on a large multi-language dataset and then finetune on a relatively small dataset. We chose the MLT-2017 as the pre-training dataset, which is a large real-world dataset with 9 languages. In the MLT-2017, there are a total of 7,200 training images, 1,800 validation images, and 9,000 testing images. We used a total of 9,000 images in the training and validation datasets for pre-training.

In downstream tasks for finetuning, we used public datasets such as ICDAR2015, MSRA-TD500, SCUT-CTW1500, Total-Text, and DAST1500 datasets to test the effectiveness of our ideas.

The characteristics of these types of datasets are different. We test on text datasets with different characteristics to comprehensively evaluate the effectiveness and verify the superiority of our method. Furthermore, it shows that our method has significant advantages over other popular methods in detecting curved and dense text.

4.2 Implementation Details

Training Our training strategy is to first train 100 epochs on the MLT-2017 dataset, use F1 as the evaluation criterion, and select the optimal epoch training result as the pre-trained model. Then, based on the pre-trained model obtained above, fine-tuning is performed on different real datasets for 1,200 epochs. Among them, we follow the poly learning rate decay strategy, and the initial learning rate is set to 0.0002. In addition, we set the batch-size at training time to 16 and use the Adam optimizer with parameter decay of 0.05 to speed up the training network convergence. In terms of training data augmentation, we follow common augmentation methods: (1) random rotation with an angle variation range of (-10, 10); (2) random cropping of images; (3) random flipping. To improve the efficiency of training, all images are scaled to 640x640 for training.

Table 1: Comparison table of ablation experiment results.

Backbone	Border Branch	MSFEM	ICDAR2015				DAST1500			
			P	R	F	FPS	P	R	F	FPS
ConvNeXt-Tiny	✗	✗	89.7	84.6	87.1	10	86.4	79.1	82.6	39
ConvNeXt-Tiny	✓	✗	89.5	85.9	87.6	8	87.8	79.1	83.2	38
ConvNeXt-Tiny	✗	✓	90.4	87.1	88.7	6	83.5	83.3	83.4	25
ConvNeXt-Tiny	✓	✓	92.4	86.0	89.0	5	86.9	82.1	84.5	23

Inference During the inference phase, we maintain the aspect ratio of the test image and scale the image by the appropriate shortest side. The center map and border map obtained from the two CM modules and the corresponding probability map results obtained in the Fusion module are used for inference. Originally, the DB module will use Equation 4 to generate center binary image and border binary image to get the final result. To speed up the inference speed, the Fusion module can be discarded and only the results of the center map and border map are combined for inference. The specific fusion method is as follows:

$$output = centermap \cdot (1 - bordermap > t) \tag{4}$$

where t is the pixel classification threshold of the border map.

4.3 Ablation Study

To verify the effect of border learning and the multi-scale feature augmentation module on curved and dense text, we construct an ablation experiment on the ICDAR2015 with horizontal and multi-orientation text and DAST1500 with curved and dense text. The experimental results are shown in Table 1.

Border branch It can be seen from Table 1, border branch can strengthen border learning and improve the overall text detection effect. Border branch improves the accuracy of the network on ICDAR2015 and DAST1500 by 0.5% and 0.6%, respectively. Moreover, border branch does not bring about a large time consumption, and the speed in the inference phase does not change much.The comparison result with and without border branch is shown in Fig 9.

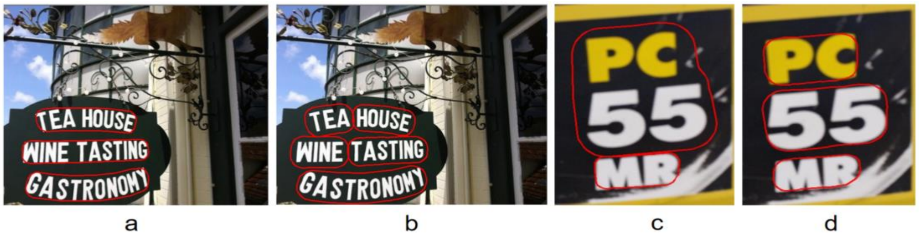


Fig. 9: Comparison results with and without border branch. Fig a and fig c are the results without border branch, and fig b and fig d are the corresponding results with border branch.

MSFEM MSFEM can fully fuse and amplify the feature information extracted at each scale, and strengthen feature semantic. As shown in Table 1, when joining MSFEM to the network improve ICDAR2015 and DAST1500 by 1.6% and 0.8%, respectively. When joining MSFEM and border branch to the network can improve both ICDAR2015 and DAST1500 by 1.9%.

Backbone In this study, we use ConvNeXt[12] as backbone, which is an enhanced version of ResNet[5], and its effect of use in various visual fields has been affirmed in recent years. To be fair with other methods, we use the tiny version of ConvNeXt[12] as backbone. The computation of the tiny version is similar to ResNet-50[5] network, and the basic feature extraction ability is stronger.

4.4 Comparisons with Previous Methods

Comparisons with previous methods is conducted on four datasets (including curved text and multi-oriented text).

Curved Text Detection We demonstrate the robustness of our method on two datasets with curved text (Total-Text and SCUT-CTW1500). As shown in Table 2, our method achieves accuracy that exceeds state-of-the-art methods.

Total-Text The dataset is word-level annotated. In the inference phase, the test image is scaled according to the shortest side of 800. As can be seen from a and b in the Fig 10, our method can accurately detect irregular text at the word-level. Furthermore, our method achieves F-measure accuracy of 87.1% shown in Table 2.

SCUT-CTW1500 The dataset is sentence-level annotated. In the inference phase, the test image is scaled according to the shortest side of 800. The visualization results are shown in c and d of Fig 10, our method can detect the border of text instances more precisely. As can be seen from Table 2, compared with other SOTA methods, the accuracy of our method exceeds by 2.1%.

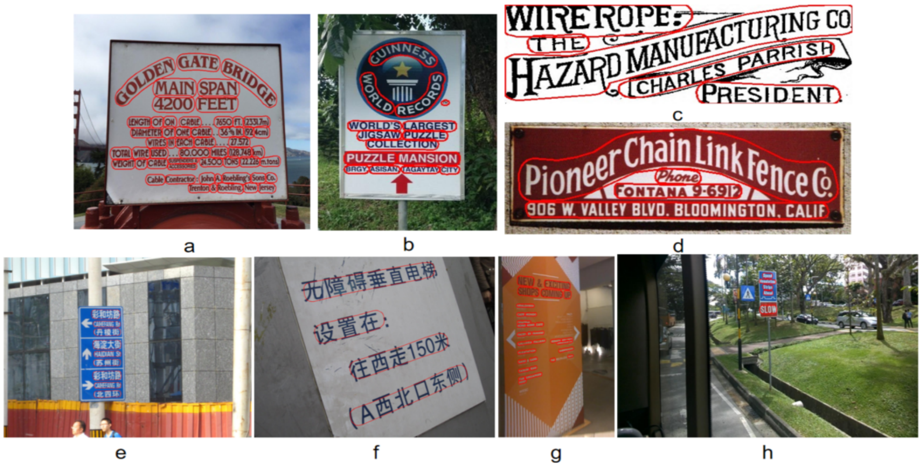


Fig. 10: Visualization of Text results.Fig a and fig b are the results of Total-Text, fig c and fig d are the results of SCUT-CTW1500. Fig e and fig f are the results of ICDAR2015, fig g and fig h are the results of MSRA-TD500.

Multi-Orientation Text Detection As is evident from the above, BorderNet can significantly outperform other methods in curved text detection. To further verify the ability of BorderNet to detect text of arbitrary shapes, we perform validation on the ICDAR2015 and MSRA-TD500 datasets, proving that BorderNet can still achieve competitive results on multi-oriented text detection tasks.

ICDAR2015 The dataset is word-level annotated. In the inference phase, the test image is scaled according to the shortest side of 1152. The visualization results are shown in a and b of Fig 10, some difficult samples of different scales can still be detected accurately. As can be seen from Table 2, BorderNet achieves 89.0% accuracy, outperforming other SOTA methods by 1.7%.

MSRA-TD500 The dataset is sentence-level annotated. In the inference phase, to improve the inference speed, the test image is scaled according to the shortest side of 736. The visualization results are shown in c and d of Fig 10. BorderNet can accurately detect long text lines with multi-Oriented characteristic. As is evident from the Table 2, BorderNet achieves accuracy of 89.9% in F-measure, surpassing other SOTA methods by 2.7%.

From the above, it can be seen that our method can accurately detect both line-annotated and word-annotated multi-oriented text. This further proves the stability and versatility of BorderNet.

Table 2: Comparison table of experiments results with SOTA.

Methods	Ext	Total-Text				SCUT-CTW1500				ICDAR2015				MSRA-TD500			
		P	R	F	FPS	P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
ATRR[25]	Syn	80.9	76.2	78.5	-	80.1	80.2	80.1	-	89.2	86.0	87.6	-	85.2	82.1	83.6	10.0
CTPN[21]	-	-	-	-	-	60.4	53.8	56.9	7.14	74.2	51.6	60.9	7.1	-	-	-	-
EAST[32]	-	50.0	36.2	42.0	-	78.7	49.1	60.4	21.2	83.6	73.5	78.2	13.2	87.3	67.4	76.1	-
RRD[8]	-	-	-	-	-	-	-	-	-	88.0	80.0	83.8	6.5	87.0	73.0	79.0	10.0
MCN[13]	-	-	-	-	-	-	-	-	-	72.0	80.0	76.0	-	88.0	79.0	83.0	-
PixelLink[4]	-	-	-	-	-	-	-	-	-	85.5	82.0	83.7	-	83.0	73.2	77.8	3.0
TextSnake[15]	Syn	82.7	74.5	78.4	-	67.9	85.3	75.6	1.1	84.9	80.4	82.6	1.1	83.2	73.9	78.3	1.1
TextField[29]	Syn	81.2	79.9	80.6	-	83.0	79.8	81.4	-	84.3	83.9	84.1	1.8	87.4	75.9	81.3	-
PSENet[23]	MLT	84.0	78.0	80.0	3.9	84.8	79.7	82.2	3.9	86.9	84.5	85.7	1.6	-	-	-	-
CRAFT[1]	Syn	87.6	79.9	83.6	-	86.0	81.1	83.5	-	89.8	84.3	86.9	-	88.2	78.2	82.9	8.6
PAN[24]	Syn	89.3	81.0	85.0	39.6	86.4	81.2	83.7	39.8	84.0	81.9	82.9	26.1	84.4	83.8	84.1	30.2
FAST[2]	MLT	90.5	82.5	86.3	46.0	87.2	80.4	83.7	66.5	89.9	84.4	87.0	15.7	90.9	83.0	86.7	56.8
DBNet[7]	Syn	87.1	82.5	84.7	32.0	86.9	80.2	83.4	22.0	91.8	83.2	87.3	12.0	91.5	79.2	84.9	32.0
DBNet++[9]	Syn	88.9	83.2	86.0	28.0	87.9	82.8	85.3	26.0	90.9	83.9	87.3	10.0	91.5	83.3	87.2	29.0
BorderNet(Ours)	MLT	89.4	84.9	**87.1**	19.0	87.8	87.0	**87.4**	19.0	92.4	86.0	**89.0**	5.0	90.9	88.8	**89.9**	21.0

5 Limitation

Because BorderNet is a segmentation-based method, it can not handle cases where a text instance is centered inside another text instance. Although our method improves the accuracy of the border region and helps to distinguish text instances that are close together, it can not handle the case when a text instance locate in the center of another text, which is also a common problem with segmentation-based methods. In the future, we will explore the idea of instance segmentation to solve this problem by fusing the advantages of detection and segmentation.

6 Conclusion

In this study, we propose an novel and efficient framework for detecting scene text of curved and dense. The framework can detect scene text by learning text border regions. For detecting scene text more accurately and faster, we propose MSFEM, which makes the feature fusion more reasonable and improves the detection efficiency. This method has surpassed the accuracy of other methods on four benchmark datasets of text detection with different text instances, particularly for distinguishing text instances with close distances and arbitrary shapes. In the future, we will simplify our network while ensuring accuracy, making the network more lightweight. Moreover, we will focus on distinguishing dense or even sticky text instances to improve the detection accuracy on this type of text.

References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9365–9374 (2019)
2. Chen, Z., Wang, W., Xie, E., Yang, Z., Lu, T., Luo, P.: Fast: Searching for a faster arbitrarily-shaped text detector with minimalist kernel representation. arXiv preprint arXiv:2111.02394 (2021)
3. Dai, P., Zhang, S., Zhang, H., Cao, X.: Progressive contour regression for arbitrary-shape scene text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7393–7402 (2021)
4. Deng, D., Liu, H., Li, X., Cai, D.: Pixellink: Detecting scene text via instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
6. He, W., Zhang, X.Y., Yin, F., Liu, C.L.: Deep direct regression for multi-oriented scene text detection. In: Proceedings of the IEEE international conference on computer vision. pp. 745–753 (2017)
7. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11474–11481 (2020)
8. Liao, M., Zhu, Z., Shi, B., Xia, G.s., Bai, X.: Rotation-sensitive regression for oriented scene text detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5909–5918 (2018)
9. Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X.: Real-time scene text detection with differentiable binarization and adaptive scale fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
10. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
11. Liu, Y., Jin, L.: Deep matching prior network: Toward tighter multi-oriented text detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1962–1969 (2017)
12. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022)
13. Liu, Z., Lin, G., Yang, S., Feng, J., Lin, W., Goh, W.L.: Learning markov clustering networks for scene text detection. arXiv preprint arXiv:1805.08365 (2018)
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
15. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: Textsnake: A flexible representation for detecting text of arbitrary shapes. In: Proceedings of the European conference on computer vision (ECCV). pp. 20–36 (2018)
16. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 67–83 (2018)

17. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia **20**(11), 3111–3122 (2018)
18. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 3, pp. 850–855. IEEE (2006)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
21. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: European conference on computer vision. pp. 56–72. Springer (2016)
22. Vatti, B.R.: A generic solution to polygon clipping. Communications of the ACM **35**(7), 56–63 (1992)
23. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9336–9345 (2019)
24. Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., Shen, C.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8440–8449 (2019)
25. Wang, X., Jiang, Y., Luo, Z., Liu, C.L., Choi, H., Kim, S.: Arbitrary shape scene text detection with adaptive text region representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6449–6458 (2019)
26. Wang, Y., Xie, H., Zha, Z.J., Xing, M., Fu, Z., Zhang, Y.: Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11753–11762 (2020)
27. Wu, Y., Natarajan, P.: Self-organized text detection with minimal post-processing via border learning. In: proceedings of the IEEE international conference on computer vision. pp. 5000–5009 (2017)
28. Xiao, S., Peng, L., Yan, R., An, K., Yao, G., Min, J.: Sequential deformation for accurate scene text detection. In: European Conference on Computer Vision. pp. 108–124. Springer (2020)
29. Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., Bai, X.: Textfield: Learning a deep direction field for irregular scene text detection. IEEE Transactions on Image Processing **28**(11), 5566–5579 (2019)
30. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5728–5739 (2022)
31. Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., Ding, X.: Look more than once: An accurate detector for text of arbitrary shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10552–10561 (2019)

32. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 5551–5560 (2017)
33. Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W.: Fourier contour embedding for arbitrary-shaped text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3123–3131 (2021)
34. Zhu, Y., Du, J.: Textmountain: Accurate scene text detection via instance segmentation. Pattern Recognition **110**, 107336 (2021)