2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Metaverse)

Arbitrary-Shaped Text Detection With Watershed Segmentation Network

Zhikui Chen[1], Yipeng Lv[1], Suhua Zhang[1], Hao Ren[1], Feng Wang[2], and Fangming Zhong[1,*]
[1] School of Software Technology, Dalian University of Technology, Dalian, China
[2] Dalian Bingshan Guardian Automation Co., Ltd, Dalian, China
fmzhong@dlut.edu.cn

***Abstract*—Arbitrary-shaped text detection is an important task in computer vision which has achieved great success in text reading systems. However, there still exist several challenges to be solved. For example, the regression-based text detection methods always fail to fit highly-curved text contours accurately. While the semantic segmentation-based methods could fit any geometries accurately, most of them employ single-size convolution kernels which ignore the diversity of receptive fields. To tackle the above issues, we propose a Watershed Segmentation Network (WSS-Net) which consists of two stages. Firstly, the pixel-level predictions of masks including text regions mask and text kernel regions mask are generated by the semantic segmentation network, in which two strategies i.e., deepening feature fusion and expanding receptive field are designed. Thus, the generated region masks can fit the contours of various-scale text instances better. Secondly, a rebuilding module based on the watershed algorithm is proposed to rebuild text instances according to the original image and two region masks, which is a widely used post-processing technique. Extensive experiments are carried out on Total-Text, CTW1500, and MSRA-TD500 to evaluate the effectiveness of the proposed WSS-Net. The results show that our WSS-Net achieves the highest F-measure score against several state-of-the-arts on three datasets.**

***Index Terms*—Arbitrary-shaped text detection, Watershed segmentation, Semantic segmentation**

I. Introduction

Scene text detection is a challenging task in computer vision, which has been widely used in many applications such as visual place recognition [1], robot navigation [2], and image understanding [3]. The text detecting models aim at locating the text regions accurately in the natural image, so as to improve the recognition accuracy of the subsequent task. During text detecting, a candidate set is usually proposed firstly for the text regions, and then following a post-processing to rebuild the text instances. Although grate success have been achieved in recent years, scene text detection is still challenging due to the diverse scales, arbitrary shapes, especially fitting highly-curved text contours.

Recently, many scene text detectors have been proposed, which can be roughly classified into two categories, i.e., segmentation-based text detectors and regression-based text detectors. The segmentation-based text detectors can be seen as bottom-up perspective methods. These methods classify each image pixel, thus they can detect arbitrary-shaped texts tightly. In addition, a post-processing is usually employed to rebuild text instances which can avoid detecting multiple adhesive text instances as one [4–7]. Regression-based text detectors can be seen as top-down perspective methods. These methods treat text instance in image as a whole object, and directly estimate the bounding boxes as the detection results [8–11]. Although huge progress has been achieved in scene text detection, there are still several issues to be solved. For example, the regression-based methods [8, 11] detect arbitrary shape text instances by text boxes. Since the boxes output from the models are quadrilateral, they cannot fit contours of irregular-shaped text tightly, resulting in poor detection performance. While the segmentation-based methods can fit any geometries accurately, most of them employ single-size convolution kernels which ignores the diversity of receptive fields [12], leading to weak detecting performance for highly-curved text instances.

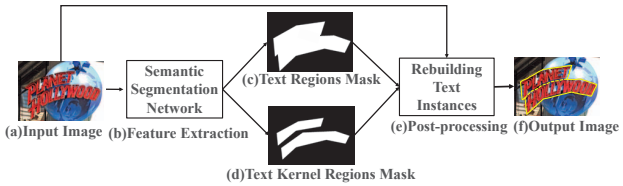

Fig. 1: The overall pipeline of WSS-Net.

In this study we propose a novel Watershed Segmentation Network (WSS-Net) as shown in Fig. 1. There are two stages in WSS-Net, i.e., predicting text regions mask and text kernel regions mask by semantic segmentation network (Fig. 1 (b)), and rebuilding text instances in image from the predicted regions masks (Fig. 1 (e)). Text regions mask (Fig. 1 (c)) indicates text instances regions, and text kernel regions mask (Fig. 1 (d)) indicates central regions of text instances. In the first stage we predict these two masks based on semantic segmentation for accurately fitting the irregular contours of text instances in image. Firstly, the semantic segmentation network adopts Resnet-18 as the backbone network. Due to the poor feature extraction ability of backbone network,

* Corresponding author. This work is supported by the National Natural Science Foundation of China (62076047,62006035), Dalian Science and Technology Key Plan (2021JB12GX019) and Innovation Foundation (2021JJ12SN44), and the Fundamental Research Funds for the Central Universities (DUT22RC(3)011).

979-8-3503-4655-8/22/$31.00 ©2022 IEEE

DOI 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00060

enhanced feature pyramid network (FPN) is used to fuse the output feature maps from backbone network. Inspired by Feature Pyramid Enhancement Module (FPEM) [12], we propose a Deepening Feature Fusion (DFF) module which can fuse the output feature maps from the second to the fifth convolution stages of backbone network. Different from FPN, DFF module performs up-sampling fusion and down-sampling fusion which can enhance the fusion of spatial information and semantic information. The feature maps generated from the backbone and DFF module are processed only by 3x3 convolution kernels which causes monotonous receptive field of model. In order to enrich receptive field of model, we propose a expanding receptive field (ERF) module which includes three different rates of dilation convolution which are multi-size convolution kernels. In addition, ERF module expands receptive field by processing four levels feature maps from DFF module. Finally, the semantic segmentation network generates prediction results of text regions mask and text kernel regions mask.

In the second stage, we propose a Rebuilding Text Instances module (RTI) as post-processing to rebuild contours of text instances in image with the help of original image and two masks. Firstly, original image and two masks are preprocessed with smoothing or noise reduction. Then, by taking the text regions mask as the water storage area and the text kernel regions as the starting water injection area, the watershed algorithm [13] continuously inject water to find the contours of different water areas and rebuild text instances. To evaluate the performance of our model on arbitrary-shaped text detection, extensive experiments are carried out on Total-Text, CTW1500, and MSRA-TD500 datasets. Our method achieves a F-measure score of 84.56%, 81.68%, and 83.51% on four datasets, respectively.

The main contributions of this work are as follows:

- An Expanding Receptive Field (ERF) module is proposed to enrich receptive field. ERF module with multi-size convolution kernels can increase the model robustness for detecting various scales text instances.

- A Rebuilding Text Instances (RTI) module is proposed to rebuild text instances in post-processing. The module treats regions masks as priori knowledge and uses noise reduction, smoothing, and watershed algorithm to rebuild contours of text instances.

II. Related Works

In recent years, a number of detectors have been proposed for scene text detection task, which can be roughly divided into regression-based text detection and segmentation-based text detection.

A. Regression-based Text Detection

In regression-based text detection methods, they treat each text instance as a whole and mark them with a text box. For example, Xie et al. [9] mainly inspired by Mask R-CNN, proposed a supervised pyramid context network (SPCNET) to precisely locate text regions while suppressing false positives. Huang et al. [14] proposed the pyramid attention network as a new backbone network of Mask R-CNN which can enhance the feature representation ability of Mask R-CNN. Jiang et al. [10] proposed an intersection-over-union overlap loss and a CMax-OMin strategy to select tighter positive samples for training. Besides, they trained a bounding box regressor as post-processing to further improve model performance. In addition, several works also propose free anchor modules to detect text. Zhou et al. [11] utilized VGG-16 and feature pyramid networks to extract features from image. Then the redundant proposal text boxes are removed by NMS operation. Liu et al. [15] employed parameterized bezier-curve to predict the key point of oriented or curved scene text and bezier align layer module to regular text boxes.In [8], the Fourier Contour Embedding (FCE) is used to fit closed shape of text instances and inverse fourier transformation to reconstruct text contours in the image spatial.

However, the regression-based text detection methods cannot well fit the arbitrary-shaped text instances because of the uncertain contour of text instances.

B. Segmentation-based Text Detection

In these methods, text instance is treated as a cluster of pixels. These models not only predict whether each pixel belongs to text instances, but also need to predict which text instance that the pixel belongs to. Therefore, in addition to generating the text regions mask, these models also need auxiliary information to rebuild text instance by post-processing. For example, Wang et al. [12] predicted text regions mask and text kernel regions mask at first. To rebuild text contours, they use similar vectors to expand text kernel regions to text regions. Wang et al. [4] generated different scale kernel regions masks for each text instance. Then the model gradually expands different scale kernels to the text instances with the complete shape. Hu et al. [5] predicted score maps of text contour (TC), text center intensity (TCI), and text kernel (TK) as intermediate results. The TC can introduce text contour information, the TCI can help to learn the accurate text segmentation, and the TK can generate the complete shape of text instances. In [6], the authors utilized U-net to generate heatmaps of candidate text regions. Then they use the text fill algorithm to generate text contours. Liao et al. [7] predicted text regions map and contour threshold map and rebuild text instances by differentiable binarization.

Although segmentation-based text detection methods can fit arbitrary-shaped text instances effectively, several models such as [5] needs more complex feature extraction structures and post-processing to get accurate text detection results.

III. Methods

In this section, we first introduce the method of segmentation-based scene text detector in this study, which can predict binarization text regions mask and text kernel

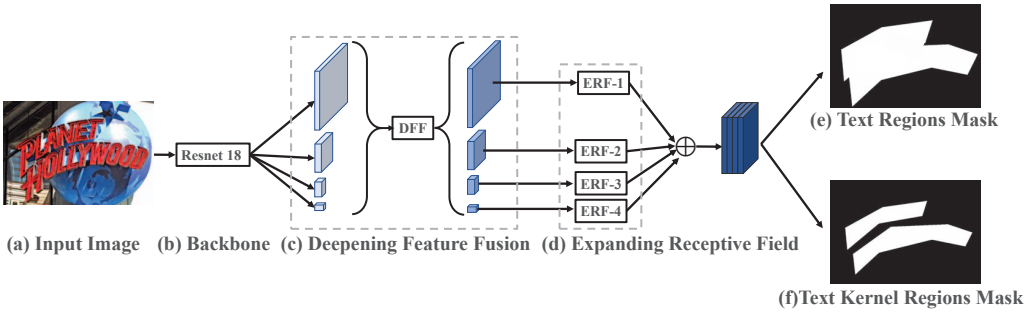

Fig. 2: The structure of semantic segmentation network. The feature maps from backbone network are enhanced by deepening feature fusion module and expanding receptive field module. $\oplus$ means resizing and concatenating feature maps. The network predicts text regions mask and text kernels regions mask to describe the text instances.

regions mask. Then we propose Rebuilding Text Instances module, which can rebuild text instances by masks generated before.

A. Semantic Segmentation Stage

The proposed model follows a segmentation-based pipeline to detect arbitrary-shaped text instances as shown in Fig. 2. Here, Resnet-18 is used as backbone network of WSS-Net. There are 4 different levels output feature maps (see Fig. 2) generated by the 2nd to 5th convolution stages of backbone network.

The main difference of four backbone output feature maps lies in that high-level convolution feature maps have more semantic information but lack spatial information due to their small size. In contrast, low-level convolution feature maps are larger in size and rich in spatial information but less in semantic information. Therefore, it is necessary to combine spatial information and semantic information by fusing low-level and high-level convolution feature maps. Following FPEM [12], we also propose deepening feature fusion (DFF) module (See Fig. 3) which is an improved pyramid feature network. DFF module consists of two phases, namely, up-scale fusion and down-scale fusion. In order to match the input size of DFF module, four output feature maps from backbone network are changed the channel number of each feature map to 128 by 1×1 convolution. Then, the four feature maps are fed into the up-scale fusion phase which upsamples high-level feature maps step by step and fuses with low-level feature maps in turn. Next, the combined feature maps are fed into down-scale fusion phase which down samples low-level feature maps step by step and fuses with high-level feature maps in turn.

In previous, the feature maps generated by DFF are only processed by 3x3 convolution kernels which lacks the diversity of detection receptive fields. In order to expand receptive field and enhance the detection ability of detecting various scales of text instances, we propose a expanding receptive field (ERF) module (see Fig. 2 (d)). The ERF has four independent parallel branches which have the same structure. One branch of ERP module is shown in Fig. 4

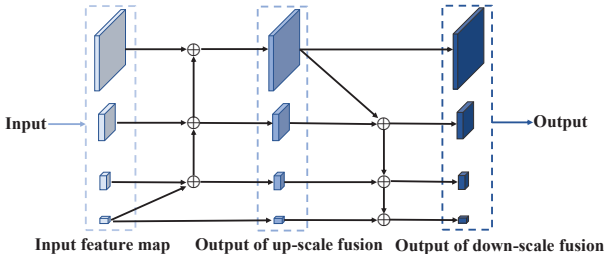

Fig. 3: The details of DFF. $\oplus$ represents mixing feature maps by resize, element-wise addition and convolution.

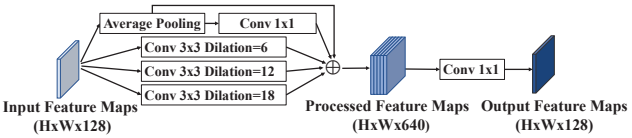

Fig. 4: One branch of ERP's structure details. $\oplus$ means resizing and concatenating feature maps.

in detail. The four branches have different input and output image sizes (H and W in Fig. 4) to process four different level of feature maps from DFF, respectively. Inspired by ASPP [16], each branch performs three parallel dilation convolution with different dilation rates of 6, 12, 18 on input feature maps. At the same time, the input feature map is convolved with 1x1 kernel after average pooling. At this point, the branch produces five output feature maps in total: three feature maps after dilation convolution, one feature map after average pooling and one feature map after 1x1 convolution. We keep the output feature maps consistent with input shape through resize and 1x1 convolution.

Then, through a two-layers full convolution network and upsampling to generate two binarization masks as the output of semantic segmentation stage. One is the text regions mask (see Fig. 2 (e)), where the value of the pixel within the text regions is 1 and the value of the pixel outside the text regions is 0. The other is text kernel regions mask (see Fig. 2 (f)). The value of the pixel within the text kernel regions is 1,

and the value of the pixel outside the text kernel regions is 0. It is worth noting that the objective of the text regions mask is to find the pixels where all the text in the image is located. Differently, the objective of the text kernel regions mask is finding the pixels of all text instances' center region in the image.

B. Rebuilding Text Instances

In this stage, Rebuilding Text Instances (RTI) module is proposed to complete the detection text instances in image (see Fig. 5) by using the original image (see Fig. 5 (a)) and two masks (see Fig. 5 (b) (c)) generated by semantic segmentation network. In RTI module, three input images must be preprocessed firstly. The original image is processed by median filter to reduce the noise. The two masks are smoothed by morphological operation to avoid RTI from making a wrong segmentation in the rough region contour. To avoid pixels in non-text regions affecting subsequent processing, text regions mask is used to combine with denoised original image by using hadamard products. The processed original image is shown in Fig. 5 (d).

Then RTI generats the mark image (see Fig. 5 (e)) in which different colors represent different label numbers. The mark image provides prior knowledge for the subsequent watershed algorithm to avoid over-segmentation results. Pixels in mark image are divided into three classes, i.e., uncertain regions of text regions, non-text regions, and text kernel regions. Uncertain regions are generated by text regions mask minus text kernel regions mask. Non-text regions include the pixels in the text regions mask with pixels' values of 0. The text kernel regions are the regions whose pixels' value is 1 in the text kernel regions mask. In mark image, uncertain segmented regions are marked with 0, the non-text regions are marked with 1, and connected components of text kernel regions are marked with different numbers starting from 2.

Finally, processed original image and mark image are input into the watershed segmentation algorithm. The traditional watershed algorithm based on gradient image treats the region with high value as mountain peak and the region with low value as valley. Each valley is treated as a water injection location. With more water pouring into the image, the water level will continue to rise, filling the valley and potentially flooding the peak. Dams are built to prevent the confluence of water from different water injection locations and these dams are the contours formed after image segmentation. In order to obtain the edge information of the image, the gradient image in Eq. (1) is usually taken as the input image,

$$\begin{aligned} G(x,y) &= grad[f(x,y)] = \sqrt{g_x^2 + g_y^2} \\ g_x &= f(x,y) - f(x-1,y) \\ g_y &= f(x,y) - f(x,y-1) \end{aligned} \tag{1}$$

where $f(x,y)$ represents the original image and $grad[\cdot]$ is the gradient operation. Since the noise or tiny brightness changes on image can easily leads to over-segmentation, it is necessary to use priori knowledge to limit the segmentation regions. Therefore, RTI module uses the watershed algorithm based on mark image (see Fig. 5(e)) generated by two masks (see Fig. 5(b)(c)). The result is shown in Fig. 5(f). The dotted line in Fig. 5(f) generated by RTI is regarded as the contour between two close text instances.

C. Loss Function

The objective of WSS-Net is given by Eq. (2):

$$\mathcal{L} = \mathcal{L}_{text} + \lambda \mathcal{L}_{kernel} \tag{2}$$

where $\mathcal{L}_{text}$ and $\mathcal{L}_{kernel}$ are the loss for the text region detection branch and text kernel region detection branch, respectively. λ is a parameter to balance $\mathcal{L}_{text}$ and $\mathcal{L}_{kernel}$. Since both branches are considered equally important, λ is set to 1 in all experiments.

we follow [4] and adopt dice loss which performs well in tasks with extreme imbalance number of positive and negative sample pixels to supervise the prediction result. $\mathcal{L}_{text}$ and $\mathcal{L}_{kernel}$ are giaven by Eq. (3) and Eq. (4), respectively:

$$\mathcal{L}_{text} = 1 - \frac{2\sum_i P_{text}(i) G_{text}(i)}{\sum_i P_{text}(i)^2 + \sum_i G_{text}(i)^2} \tag{3}$$

$$\mathcal{L}_{kernel} = 1 - \frac{2\sum_i P_{kernel}(i) G_{kernel}(i)}{\sum_i P_{kernel}(i)^2 + \sum_i G_{kernel}(i)^2} \tag{4}$$

where $P_{text}(i)$ and $P_{kernel}(i)$ refer to the i^{th} pixel in the prediction masks for text regions and text kernel regions, respectively. Similarly, $G_{text}(i)$ and $G_{kernel}(i)$ refer to the i^{th} pixel in the ground-truth masks for text regions and text kernel regions, respectively. These four masks are binary images with the same size as the input image, in which text pixels in P_{text} and G_{text} or text kernel pixels in P_{kernel} and G_{kernel} are 1 and not-text pixels in P_{text} and G_{text} or not-text kernel pixels in P_{kernel} and G_{kernel} are 0.

We also follow [4] and use polygon clipping algorithm which computes the text kernel polygon contour points shrinkage offset to the text instance contour points provided by dataset. Then the offset is used to shrink every text instance polygon and get the text kernel polygon contour points.

IV. Experiments

In this section, we first introduce the benchmark datasets used in our experiments on scene text detection followed by implementation details. In order to verify the effectiveness of the modules in semantic segmentation, ablation studies are also carried out. Finally, the results of the proposed model on different benchmark datasets are compared against existing methods.

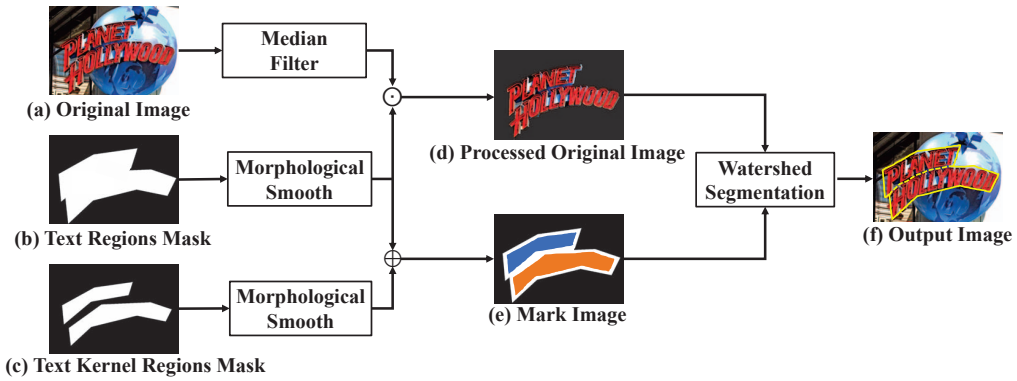

Fig. 5: The details of Rebuilding Text Instances module. The input original image and the two masks output by semantic segmentation stage are used to rebuild the text instance area in image by preprocess and watershed method. $\odot$ is hadamard products. $\oplus$ means mixing two masks and generating label numbers on mark image.

A. Datasets

Synthtext [17] is created by mixing natural images with artificially rendered text and contain 800,000 synthetic images. Following [5, 12], we use Synthtext as external dataset to pre-train our model.

CTW1500 [18] is a challenging dataset for curve scene text detection. It has 1000 training images and 500 testing images. Every text instance in this dataset is labelled by 14 polygon contour points.

TotalText [19] is a dataset for arbitrary shape scene text detection. This dataset consists of 1255 training images and 300 testing images.

MSRA-TD500[20] includes 300 training images and 200 test images with multi-direction text line instances. Since the training set of MSRA-TD500 is rather small, we follow the previous studies [4, 21] and add 400 images from HUST-TR400 [22] to the training set.

B. Implementation Details

We use the ResNet pre-trained on ImageNet as our backbone. All the networks are optimized by using Adam. There are two ways to train model, i.e., training from scratch or fine-tuning on pre-trained model. In this study, pre-trained model is firstly trained on SynthText for 1 epoch with learning rate of 0.001. We train model with batch size 16 on 1 GPU for 1000 epochs and early stop when F-score not improve in 200 recent epochs. The shrink ratio of the kernels is set to 0.7 for all datasets. Open operation with 3×3 rectangle box is adopted in morphological smooth. The same evaluation method as [12] is used to evaluate in this study. The evaluation metrics are precision, recall, and F-measure.

C. Ablation Studies

The ablation study experiments on TotalText dataset without external data and results are shown in Table I.

The effectiveness of backbone. To evaluate the influence of the backbone, the backbone is changed from Resnet18 to Resnet50. According to results (Table I No.1 and No.5), F-measure decreases by 0.34%, indicating that network depth is not positively correlated with experimental results.

TABLE I: The results of models with different structures on TotalText."P", "R" and "F" represent the precision, recall, and F-measure respectively.

No.	Model structure	TotalText		
		P	R	F
1	Resnet50+DFF+ERF	87.99	80.76	84.22
2	Resnet18	86.64	79.86	83.12
3	Resnet18+DFF	85.97	**80.83**	83.32
4	Resnet18+ERF	87.46	79.85	83.48
5	Resnet18+DFF+ERF	**88.69**	80.8	**84.56**

In order to demonstrate the effectiveness of DFF and ERF modules, only the backbone is deployed in the semantic segmentation stage and the experimental results are recorded in Table I No.2. The model structure of Resnet18+DFF+ERF (No.5) is the default model structure in this study. These two groups of results are regarded as the baselines for comparison.

The effectiveness of Resnet18+DFF. The ERF module is removed from default model structure and the output feature maps from DFF are regarded as the final output feature maps of the semantic segmentation stage. The results are recorded in Table I No.3 and the F-measure is 0.2% higher than F-measure of model only deployed backbone (Table I No.2).

The effectiveness of Resnet18+ERF. Here, the DFF module is removed from default model structure and the output feature maps from backbone are regarded as the input feature maps of ERF. the results show that F-measure of Resnet18+ERF (Table I No.4) is 0.36% higher than F-measure of model only deployed backbone (Table I No.2).

The effectiveness of Resnet18+DFF+ERF. From the experimental results in Table I No.3, No.4 and No.5, we can see that default model structure has the highest F-measure with 1.24% higher than model without ERF and 1.08% higher than model without DFF.

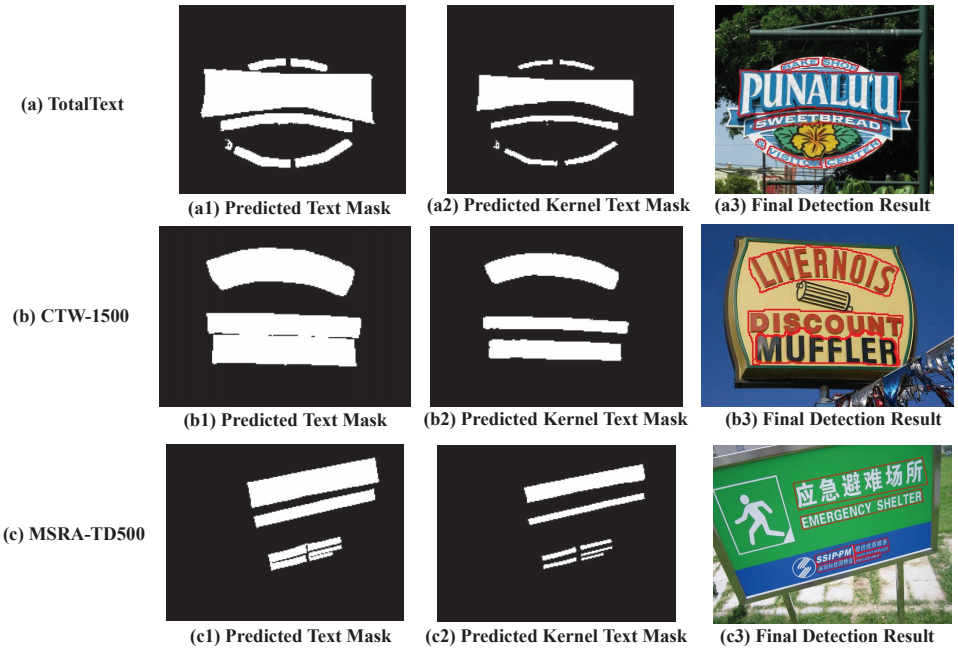

Fig. 6: The visual detection results on TotalText, CTW1500, and MSRA-TD500.

According to the above experimental results, after adding only the DFF or the ERF modules, the F-measure only improves 0.2-0.3% compared to the original backbone. While, the F-measure increases by 1-1.4% if both of them are added. Therefore, it is necessary to deepen feature fusion and expand receptive field at the same time in semantic segmentation stage.

D. Comparisons with Other Methods

In this section, the results of fine-tuning experiment are compared with the results of training from scratch firstly to illustrate the influence of external data. We then compare our model with previous methods on different datasets. Since F-measure is a comprehensive score of precision and recall, this study pays more attention to F-measure when comparing results.

TABLE II: Finetuning result "P", "R" and "F" represent the precision, recall and F-measure respectively. "Ext." indicates external data.

Dataset	Ext.	P	R	F
TotalText	×	88.69	80.8	84.56
	√	89.12	80.22	84.43
MSRA-TD500	×	87.75	79.65	83.51
	√	85.34	83.02	84.17
CTW1500	×	86.32	77.51	81.68
	√	87.93	78.38	82.89

Table II shows the experimental results of the default model structure of WSS-Net on three different datasets with two different training methods. On TotalText, the precision, recall, and F-measure with and without external data are all relatively close, and the gap is within 1%. On MSRA-TD500, the precision with fine-tuning is 2.41% lower than that without fine-tuning. But for recall, the model with fine-tuning is 3.37% higher than that without fine-tuning. And the F-measure with fine-tuning is only 0.66% higher than that without fine-tuning. On CTW1500, the precision, recall and F-measure with fine-tuning are 1.61, 0.87, and 1.21% higher than those without fine-tuning respectively. Under the same external data and training epoch, the F-measure of the experimental model does not improve by 2-5% like previous models [5, 12]. Therefore, only the results obtained without fine-tuning by external data will be compared with the results of other models. We also present an example of visual detection results as shown in Fig. 6.

TABLE III: The results on TotalText. "P", "R" and "F" represent the precision, recall and F-measure respectively. "Ext." indicates external data.

Method	Ext.	Venue	TotalText		
			P	R	F
PAN[12]	×	19ICCV	88	79.4	83.5
PSENet[4]	√	19CVPR	84	78	80.9
ABCNet[15]	√	20CVPR	87.9	81.3	84.5
TextRay[23]	√	20MM	83.5	77.9	80.6
STKM [24]	×	21CVPR	86.32	78.36	82.15
PCR [21]	×	21CVPR	86.4	81.5	83.9
Chang et al.[25]	√	21IROS	86.1	80.7	83.3
Xue et al. [26]	×	22PR	86.1	**82.6**	84.4
Wang et al.-R18 [27]	×	22IP	85.8	77	81.1
WSS-Net	×	-	**88.69**	80.8	**84.56**

To evaluate the ability of WSS-Net for arbitrary shape

text detection, we compare the results with others model on TotalText by default model structure firstly. Visual detection results on TotalText are shown in Fig. 6 (a). The short edge of image is set to 640. As shown in Table III, the precision, recall, and F-measure of our method are 88.69%, 80.8% and 84.56% respectively. As can be seen, the precision and F-measure of our method is the highest compared to other state-of-the-arts. In particular, it is noted that F-measure of our method without external data is 3.96% higher than TextRay[23] which training with external data. This demonstrates that our proposed method is effective in arbitrary-shaped text detection.

TABLE IV: The results on CTW1500. "P", "R" and "F" represent the precision, recall and F-measure respectively. "Ext." indicates external data.

Method	Ext.	Venue	CTW1500		
			P	R	F
PAN[12]	×	19ICCV	84.6	77.7	81
CSE [28]	×	19CVPR	81.1	76	78.4
ABCNet[15]	√	20CVPR	84.4	78.5	81.4
TextRay [23]	√	20MM	82.8	**80.4**	81.6
PolarText[29]	√	21EUC	83.5	78.8	81
Chang et al.[25]	√	21IROS	83	76	79.3
STKM [24]	×	21CVPR	85.08	78.23	81.51
Wang et al.-R18 [27]	×	22IP	84.6	77.7	81
WSS-Net	×	-	**86.32**	77.51	**81.68**

CTW1500 is another dataset which can show ability of our model for arbitrary shape text detection. As with TotalText, the short edge of image is set to 640 for test. The visual detection results on CTW1500 are shown in Fig. 6 (b). As shown in Table IV, the precision, recall and F-measure of our method are 86.32%, 77.51%, and 81.68% respectively. The precision and F-measure of the proposed model is also the highest. In addition, F-measure of our model without external data is 0.68% higher than PolarText[29] which training with external data.

TABLE V: The results on MSRA-TD500. "P", "R" and "F" represent the precision, recall and F-measure respectively. "Ext." indicates external data.

Method	Ext.	Venue	MSRA-TD500		
			P	R	F
CRAFT [30]	√	19CVPR	**88.2**	78.2	82.9
PAN[12]	×	19ICCV	80.7	77.3	78.9
SAE [31]	√	19CVPR	84.2	**81.7**	82.9
Chen et al.[32]	×	20ICPR	79.5	78.7	79.1
Mask-TextSpotter-v3[33]	√	20ECCV	90.7	77.5	83.5
Wang et al.[34]	×	21IPCCC	87.6	74.4	80.5
SRM-Exp[35]	×	21NC	84.23	80.76	82.46
JMNET[36]	×	22NC	84.7	80	82.3
WSS-Net	×	-	87.75	79.65	**83.51**

To test the robustness of detecting long straight text instance, we evaluate the proposed method on MSRA-TD500 dataset. The short edge of test image is set to 736. Visual detection results on MSRA-TD500 are shown in Fig. 6 (c). From Table V, we can see that our method achieves the highest F-measure by 83.51% compared to the others. It indicates that the proposed model is also robust for long straight text detection.

V. Conclusion

In this paper, we have proposed a novel scene text detector based on semantic segmentation and watershed segmentation. In the first stage, we deploy a deepening feature fusion module and a expanding receptive field module after the backbone network, which can benefit semantic segmentation. In the second stage, the watershed segmentation module uses the semantic segmentation results output by the first stage as priori knowledge to rebuild text instances in image. These two stages both improves the performance of the proposed model for arbitrary-shaped text detector. Extensive experiments on Total-Text, CTW1500, and MSRA-TD500 demonstrated the advantages when compared to previous scene text detectors. For future work, we plan to consider the text instance contours rebuilt by watershed segmentation module to improve the recall further.

References

[1] Z. Hong, Y. Petillot, D. Lane, Y. Miao, and S. Wang, "Textplace: Visual place recognition and topological localization through reading scene texts," in *ICCV*, 2019, pp. 2861–2870.

[2] C. Craye, D. Filliat, and J.-F. Goudou, "Environment exploration for object-based visual saliency learning," in *2016 ICRA*. IEEE, 2016, pp. 2303–2309.

[3] A. U. Dey, S. K. Ghosh, E. Valveny, and G. Harit, "Beyond visual semantics: Exploring the role of scene text in image understanding," *Pattern Recognition Letters*, vol. 149, pp. 164–171, 2021.

[4] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *CVPR*, 2019, pp. 9336–9345.

[5] Z. Hu, X. Wu, and J. Yang, "Tcatd: text contour attention for scene text detection," in *2020 25th ICPR*. IEEE, 2021, pp. 1083–1088.

[6] Q. Wang, Y. Zheng, and M. Betke, "A method for detecting text of arbitrary shapes in natural scenes that improves text spotting," in *CVPR Workshops*, 2020, pp. 540–541.

[7] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[8] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *CVPR*, 2021, pp. 3123–3131.

[9] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proceedings of the AAAI conference on*

artificial intelligence, vol. 33, no. 01, 2019, pp. 9038–9045.

[10] D. Jiang, S. Zhang, Y. Huang, Q. Zou, X. Zhang, M. Pu, and J. Liu, "Detecting dense text in natural images," *IET Computer Vision*, vol. 14, no. 8, pp. 597–604, 2020.

[11] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *CVPR*, 2017, pp. 5551–5560.

[12] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *ICCV*, 2019, pp. 8440–8449.

[13] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 06, pp. 583–598, 1991.

[14] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask r-cnn with pyramid attention network for scene text detection," in *2019 WACV*. IEEE, 2019, pp. 764–772.

[15] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "Abcnet: Real-time scene text spotting with adaptive bezier-curve network," in *CVPR*, 2020, pp. 9809–9818.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[17] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *CVPR*, 2016, pp. 2315–2324.

[18] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.

[19] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *2017 14th ICDAR*, vol. 1. IEEE, 2017, pp. 935–942.

[20] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *CVPR*. IEEE, 2012, pp. 1083–1090.

[21] P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in *CVPR*, 2021, pp. 7393–7402.

[22] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[23] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 111–119.

[24] Q. Wan, H. Ji, and L. Shen, "Self-attention based text knowledge mining for text detection," in *CVPR*, 2021, pp. 5983–5992.

[25] H.-C. Chang, H.-J. Chen, Y.-C. Shen, H.-H. Shuai, and W.-H. Cheng, "Re-attention is all you need: Memory-efficient scene text detection via re-attention on uncertain regions," in *2021 IROS*. IEEE, pp. 452–459.

[26] C. Xue, S. Lu, and S. Hoi, "Detection and rectification of arbitrary shaped scene texts by using text keypoints and links," *Pattern Recognition*, vol. 124, p. 108494, 2022.

[27] F. Wang, X. Xu, Y. Chen, and X. Li, "Fuzzy semantics for arbitrary-shaped scene text detection," *IEEE Transactions on Image Processing*, pp. 1–1, 2022.

[28] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh, "Towards robust curve text detection with conditional spatial expansion," in *CVPR*, 2019, pp. 7269–7278.

[29] Q. Kong, Y. Wu, and S. Wan, "Polartext: Single-stage scene text detection with polar representation," in *2021 IEEE 19th EUC*. IEEE, 2021, pp. 1–8.

[30] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *CVPR*, 2019, pp. 9365–9374.

[31] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *CVPR*, 2019, pp. 4234–4243.

[32] Y. Chen, W. Wang, Y. Zhou, F. Yang, D. Yang, and W. Wang, "Self-training for domain adaptive scene text detection," in *2020 25th ICPR*. IEEE, 2021, pp. 850–857.

[33] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *ECCV*. Springer, 2020, pp. 706–722.

[34] H. Wang, X. Xiao, Y. Hui, Z. Yin, and N. Cheng, "Two-layer federated learning for scene text detection," in *2021 IPCCC*. IEEE, 2021, pp. 1–6.

[35] M. Liang, J.-B. Hou, X. Zhu, C. Yang, J. Qin, and X.-C. Yin, "Multi-orientation scene text detection with scale-guided regression," *Neurocomputing*, vol. 461, pp. 310–318, 2021.

[36] "Jmnet: Arbitrary-shaped scene text detection using multi-space perception," *Neurocomputing*, 2022.