DeepText: Detecting Text from the Wild with Multi-ASPP-Assembled DeepLab

***Abstract*—In this paper, we propose to address the issue of scene text detection in the manner of direct regression and successfully adopt an effective semantic segmentation model, DeepLab v3+ [1], for this application. In order to handle texts with arbitrary orientations and sizes, and improve the recall of small texts, we propose to extract features from multiple scales by inserting multiple Atrous Spatial Pyramid Pooling (ASPP) layers to the DeepLab after the feature maps with different resolutions. Then, we set multiple auxiliary IoU losses at the decoding stage and make auxiliary connections from the intermediate encoding layers to the decoder to assist network training and enhance the discrimination ability of lower encoding layers. Experiments conducted on the benchmark scene text dataset ICDAR2015 demonstrate the superior performance of our proposed solution over the state-of-the-art approaches.**

***Index Terms*—Scene text detection, DeepLab, multiple ASPP, auxiliary IoU losses, auxiliary connections**

I. Introduction

Scene text detection has attracted intensive attention in the community of computer vision because it is a primary step of scene text reading, and has a great application potential. Scene images are usually captured in wild scenarios under unconstrained conditions, e.g., with complex background, uneven illumination, skew, blur, perspective distortion and so forth, which introduce huge challenges to robust scene text detection. Moreover, challenges also arise due to arbitrary text appearances in practice with respect to their sizes, fonts, colors and orientations.

A scene text detector is used to predict the locations of text areas in input scene images. The existing solutions have been through a great revolution since 2015, when deep learning was introduced to this area, and have since made a great breakthrough. Conventional detection methods can be divided into sliding-window-based approaches and connected-component-based ones. Among them, a group of detectors based on Maximally Stable Extremal Regions (MSER) had achieved the state-of-the-art performance and had become the most popular solutions by 2015. Most of the conventional methods heavily depend on handcrafted features and single character classifiers, which has seriously limited models' flexibility and adaptability. A deep network integrates feature extraction modules and task processing modules into one unified framework, and trains both modules simultaneously in an end-to-end way. Therefore, ever since the deep-learning based models were introduced for scene text detection, they have surpassed conventional methods significantly and become the dominant solutions.

Deep learning based approaches usually tackle the scene text detection task from two perspectives, i.e., object detection and semantic segmentation. Text can be seen as specific targets in scene images, so it is reasonable to borrow ideas from general object detection field. This kind of methods usually select numerous anchors at the first step, followed by calculating text confidence and related offsets for individual anchors. Arbitrary orientations and aspect ratios lead to the bottleneck of this kind of methods because the amount and pre-defined configurations of anchors are limited and fixed. By contrast, the methods based on semantic segmentation are more robust, because, in these methods, bounding boxes' offsets and angles are predicted for individual pixels. It is notable that, in both categories of methods, the task-tackling layer at the end of the network usually consists of a classification module and a regression module. He [2] et al. classified regression methods into the direct methods, which calculated offsets from a given point, and the indirect methods, which calculated offsets for proposals or anchors. Therefore, in the literature, scene text detectors are also grouped into direct-regression-based approaches (that most semantic segmentation methods belong to) or indirect-regression-based approaches (that are mainly referred to the object-detection-based methods).

According to the literature, VGG [3] and ResNet [4] are the most widely used backbone at the encoder stage of the recent state-of-the-arts, no matter if the approaches are object-detection-based or semantic-segmentation-based. However, there are many other outstanding structures such as DeepLab [1], a model proved to be superior for semantic segmentation tasks, which have not been explored for scene text detection. Motivated by this, in this work, for the first time, we introduce DeepLab, and more specifically DeepLab v3+ [1], into scene text detection and study its limitations. Finally, we find that the system's recall is seriously influenced by the misdetection of small texts. To tackle this limitation, we modify DeepLab by inserting multiple ASPP layers to it so that more detailed and richer information can be extracted and leveraged. Furthermore, we also propose to utilize multiple auxiliary Intersection-Over-Union (IoU) loss and auxiliary connections to accelerate the training process and enhance the discrimination ability of lower encoding layers.

The rest of this paper is organized as follows. Section II summarizes the related works, Section III gives the details of our proposed network, and Section IV presents our experiments conducted on benchmark datasets. Conclusions are drawn in Section V.

II. Related works

Among the scene text detectors based on traditional approaches, MSER-based ones have achieved the best performance. Yin [5] et al. firstly extracted character candidates by employing MSER, followed by eliminating non-text areas with handcrafted features and a distance metric learning model. In order to deal with text lines with arbitrary orientations, a backward-forward algorithm was then designed in their work. This approach was state-of-the-art among traditional methods, but was surpassed by deep-learning based ones in 2015. Since 2015, almost all of the published scene text detection works have been deep-learning based. Therefore, in the following, we review only the scene text detectors based on deep-learning.

EAST proposed in [6] was a direct regression method that dealt with text detection in the semantic segmentation way. The network was based on a Fully Convolutional Network (FCN) and employed VGG as its backbone. For the regression module, the geometry maps could be either a five-channel RBOX (i.e., rotated box) representation or an eight-channel QUAD (i.e., quadrangle) representation. Additionally, to optimize the proposed network, the well-known IoU loss [7] was employed by EAST. Yao [8] et al. also cast the text detection task as a semantic segmentation problem and employed an FCN model to predict text regions, character regions and the linking orientations of adjacent characters at multiple scales. A series of post processing operations were then performed to obtain text lines with arbitrary orientations from the predicted dense maps. He [2] et al. also regarded text detection as a segmentation problem and performed a pixel-level prediction. Different from Yao's [8] work, direct regression was used in this work to predict the vertex coordinates of quadrilateral text regions. The backbone of the network was also FCN, and after the network prediction, a recalled Non-Maximum Suppression (NMS) was designed to remove redundant bounding boxes.

On the other hand, text can be seen as specific objects in images, so it is natural to adopt methods successfully applied for general object detection to locate text objects in images. Great efforts have been made in this direction and achieved promising results, such as TextBoxes [9], TextBoxes++ [10] and SegLink [11] etc. The object-detection-based methods usually detect a large amount of text proposals, and then calculate the offsets for the related text bounding boxes. Therefore, these methods basically are the indirect-regression approaches. TextBox [9] predicted the text presence confidence and offsets of 12 default boxes for individual locations from six feature maps. The offsets were designed towards the boxes' top-left coordinates, heights and widths. To deal with text with arbitrary orientations, TextBox++ [10] further improved TextBox by predicting the regression of offsets for oriented bounding boxes. Since the aspect ratios of words or text lines are usually larger than other objects, using pre-defined boxes with fixed aspect ratios, as in TextBox and TextBox++, may miss some long words or text lines. In [11], Shi [11] proposed the SegLink, where words or text lines were broken into segments. The SegLink network estimated the confidence scores and geometric offsets for a set of default boxes with respect to segments, instead of words or text lines. Meanwhile, the links of segments were also detected to determine whether two adjacent segments belonged to the same word or not. Following the idea of SegLink, Deng et al. [12] proposed the PixelLink, which also detected links to gather components of words or text lines. However, Deng et al. [12] held the idea that the regression of default boxes was not indispensable in text detection. Therefore, the proposed PixelLink predicted text confidence for individual pixels instead of segments. Both PixelLink and SegLink have the same network structure except that the PixelLink eliminated the regression module at the output layer.

On the other hand, Yang [13] proposed IncepText to detect text with arbitrary orientations in the view of instance-aware segmentation. The idea of IncepText was similar to Mask R-CNN [14], where a mask module was employed together with the classification and regression parts. Another highlight of this work was the use of deformable convolution layers [15] and deformable PSROI pooling layers [15]. A standard convolution sampled the pixels in a regular grid, while the deformable convolution shifted the locations of pixels in a regular grid according to the offsets learned from the input feature maps. Thus, the problem of arbitrary text orientation was able to solved to certain extent. The deformable PSROI pooling was a modification of PSROI pooling, in which the deformable operation was used for the same purpose as the deformable convolution.

Although semantic-segmentation-methods are more robust to text with arbitrary orientations, they may yield lower recalls due to the sparse features of small texts. The Pixel-Anchor network proposed in [16] took advantages of both semantic segmentation and object detection. ResNet-50 was employed as the backbone in this network, followed by a pixel-based module and an anchor-based model. The pixel-based module used ASPP at the feature map with a resolution of 1/16, and was assembled with a RBOX predictor and an attention heat map detector. Attention heat maps and the feature maps produced by ResNet-50 were also fed to the anchor-based module, together with some adaptive predictor layers that were used to adapt the anchor module to long anchors. As we all know, the problem of high false positive is a big challenge in scene text detection. This problem is often caused by lack of context information and inaccurate classification. The SPCNET proposed by Xie et al. [17] utilized a text context module and a re-scoring mechanism to tackle the high false positive problem and has improved detection performance significantly. Their text context module contained a pyramid attention module and a pyramid fusion module, and produced text segmentation as outputs. Then, by applying a re-scoring mechanism, the classification score and instance score were combined to prevent true positives from being filtered out.

Given the robustness of direct regression to texts with arbitrary orientations and sizes, in this work, we address the issue of scene text detection in the way of semantic segmentation. The well-known DeepLab model is adopted and

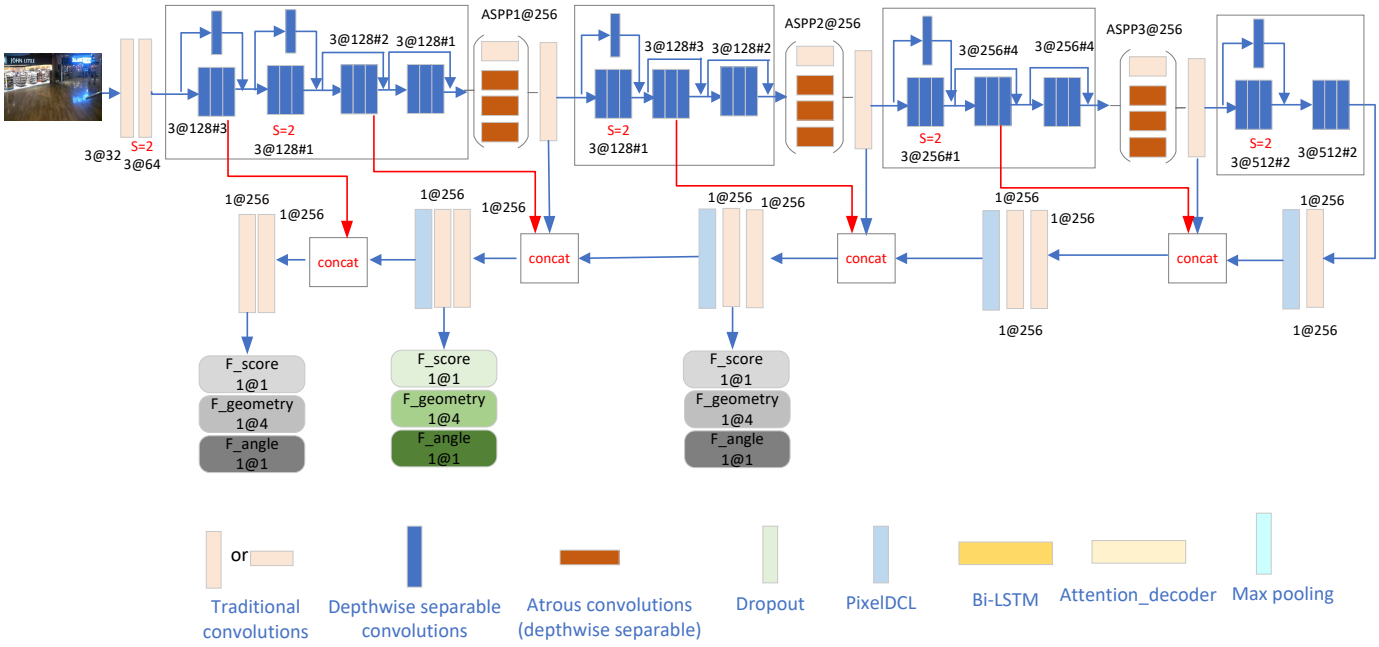

Fig. 1. The structure of the proposed network.

modified to improve the detection performance, especially the recall of small texts. More details are presented as below.

III. The Proposed Method

The overview of proposed network is shown in Fig. 1. As we can see, the DeepLab v3+ [1] is used as the backbone with some modifications on the structure and basic configurations. The original output layer is replaced with a regression module and a classification module, and the decoder branch is equipped with a smaller upsampling factor. Besides, multiple ASPP layer is exploited to improve the recall of small texts, and multiple auxiliary losses and connections are introduced to assist network training and enhance the discrimination ability of lower encoding layers.

Structure of Our Proposed Network: DeepLab v3+ [1] is an efficient semantic segmentation model that was developed on the base of DeepLab v1 [18], DeepLab v2 [19] and DeepLab v3 [20]. Advanced techniques such as atrous convolution, depthwise separable convolution, and Xception [21] are used in DeepLab v3+ and have achieved an accurate pixel-level prediction. Atrous convolution and ASPP are able to adjust filter's receptive fields so as to capture multi-scale information by configuring different atrous rates. Depthwise separable convolution is a powerful tool to relieve the computational burden, while Xception is an efficient encoder-decoder structure that can achieve high performance on both classification and semantic segmentation. To take the advantages of the above mentioned benefits, we adopt DeepLab v3+ as our base model and make some modifications, as explained below, to adapt it to the scene text detection task.

Details of the overall structure and related configuration are described in Fig. 1, where the smallest resolution at the encoder stage is 32, and the recovered resolution at the decoder stage is 2. Moreover, $a@b\#c$ means the current block (each block has 3 depthwise separable convolution layers) is repeated for c times, the kernel size of this block is $a \times a$ and there are b channels at each layer of this block. $S{=}2$ means the stride is 2 at a specific layer or the last layer of specific block. Note that, due to the limited GPU resource, denser output feature maps are not considered in our model, and the size of channel configuration of our model is much smaller compared to that of [1].

Output Layer: The DeepLab v3+ [1] originally proposed for semantic segmentation has a pixel-level prediction module in its output layer, where confidence maps with respect to individual object classes are produced. This prediction layer works well for the semantic segmentation purpose, but is not suitable for our scene text detection task. In order to locate text in images, we also need to predict the offsets from individual pixels to the related bounding boxes. Therefore, in this work, we replace the original output layer with a classification module and a regression module.

Concretely, a score map is generated to evaluate pixels' confidence of being text, and five RBOX geometry maps are generated to perform a direct regression, as shown in Fig. 2. For an individual location (X, Y), the values at the five RBOX geometry maps represent the distance to the four boundaries of the corresponding rotated box and the rotation angle of the corresponding box, respectively. During the testing stage, we restore the corresponding bounding box according to the prediction results, and eliminate the redundant boxes with the NMS algorithm.

Smaller Upsampling Factor: In [20], the features are bilinearly upsampled with a factor of 16 in the decoder stage,

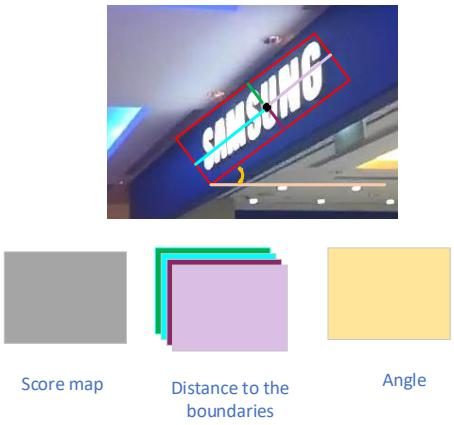

Fig. 2. Output feature maps of proposed network

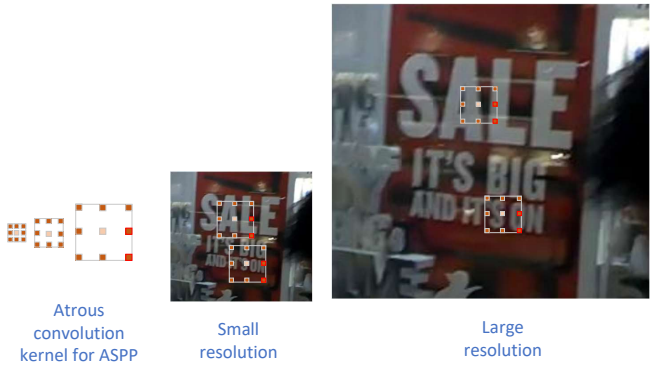

Fig. 3. Feature extraction by ASPP with the same atrous rates for text with various scales

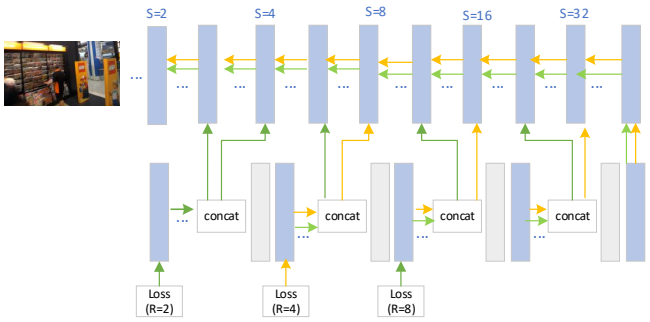

Fig. 4. Back propagation after using auxiliary losses and connections. The yellow arrows indicate back propagation paths without auxiliary losses and connections, while the green arrows represent additional paths after using auxiliary losses and connections.

which is claimed [1] to have failed when recovering segmentation details. Therefore, in DeepLab v3+ [1], the upsampling operation is performed with a factor of 4. In our case, detecting small text areas requires more detailed information and more refined features recovered, so we upsample feature maps with a smaller factor of 2 at the decoder stage, and then concatenate them with the low-level features from the encoder stage.

Multiple ASPP Layers: Small texts present frequently in scene images and detection accuracy of such texts has a great impact on the overall performance. To better deal with these text objects, we further improve the network architecture by inserting multiple ASPP layers to our network after the feature maps with different resolutions.

The original DeepLab v3+ [1] assembles only one ASPP layer after the feature maps with the smallest resolution (at the end of the encoder). As shown in Fig. 3, this operation is helpful for extracting wide range contextual information for large texts. However, when it comes to small texts, the extracted features become too coarse, and much detailed information is missed. By contrast, if a ASPP layer with the same atrous rates is applied on the feature maps with a large resolution, the extracted features would be more refined for small texts, but the contextual information contained might be too little for large texts. To take both small texts and large texts into consideration, we propose to insert multiple ASPP layers to the DeepLab after the feature maps with different resolutions. As shown in Fig. 1, we assemble three ASPP layers after the feature maps with resolutions of 4, 8, 16, respectively. In an individual ASPP layer, a traditional convolution layer with 1×1 kernel and three atrous convolutional layers with atrous rates of 6, 12 and 18 are assembled in parallel. Then, outputs of these four layers are concatenated, followed by a 1×1 traditional convolutional layer that is used to reduce the overall channels of feature maps.

Multiple Auxiliary Losses and Connections: To optimize our proposed network, the IoU loss [7], as defined in 1, is employed in our work. The IoU loss is originally proposed for object detection. Compared with the widely used L2 loss that optimizes the four values of distance independently, the IoU loss is invariant against different scales of objects. Given the predicted bounding boxes R^* and the ground truth bounding boxes R (their related orientations are denoted by θ^* and θ, respectively), the IoU loss minimizes the difference between their intersection area and their union area. In our case, the IoU loss is calculated for individual pixels, and the predicted bounding box R^* is derived from the five geometry maps produced by the output layer.

$$\begin{aligned} Loss_{IoU} &= Loss_{area} + Loss_{angle} \\ Loss_{area} &= -logIoU(R, R^*) = -log\frac{|R \cap R^*|}{|R \cup R^*|} \\ Loss_{angle} &= \lambda * (1 - cos(\theta - \theta^*)) \end{aligned} \tag{1}$$

where

$$|R \cup R^*| = |R| + |R^*| - |R \cap R^*| \tag{2}$$

Subsequently, to assist the training of the proposed network and promote the convergence speed, we propose to employ multiple auxiliary IoU losses and connections at the decoder module, which is expected to be able to enhance the gradient signals during the back propagation procedure. The existing scene text detectors usually calculate the loss once on the final decoded feature maps. For example, EAST [6] calculated the loss on the feature maps with $1/4$ resolution and

TABLE I
COMPARISON WITH PRIOR ARTS ON ICDAR2015

Methods	Additional data	Recall	Precision	F-measure
EAST [6]	ImageNet	73.47	83.57	78.20
SegLink [11]	SynthText	76.80	73.10	75.00
RRPN [22]	ImageNet, SVT	73.23	82.17	77.44
R^2CNN [23]	ImageNet etc.	74.29	76.42	75.34
TextBoxes++ [10]	SynthText	76.70	87.20	81.70
PixelLink [12]	✗	82.00	85.50	83.70
TextSpotter [24]	SynthText	81.20	85.80	83.40
DeepLab_small	✗	77.03	86.21	81.36
DeepLab	✗	77.80	87.49	82.36
DeepLab_MASPP	✗	81.08	87.57	84.20
Ours	✗	81.13	88.27	84.55

PixelLink [12] did on the feature maps with $1/2$ resolution. Moreover, in these models, up-sampled features of the decoder module are often concatenated with the low level feature maps that have the same resolution from only one layer. These strategies make the learning of low level weights slow and the learned features less discriminative. In this work, to enhance the discrimination power of low encoder layers and speed up the convergence, we calculate the IoU loss three times on the feature maps with resolution $1/2$, $1/4$ and $1/8$, respectively, and make auxiliary connections from multiple intermediate encoder layers, as shown in Fig. 1. Note that, in the inference stage, we only perform prediction at the feature maps of $1/4$ resolution to save time. Fig. 4 describes the back propagation details. As we can see, the gradients are enhanced by the auxiliary losses and connections.

IV. EXPERIMENTS

To demonstrate the effectiveness of our proposed detector, we test our proposed solution on the benchmark dataset ICDAR2015 and compare it with the state-of-the-art approaches.

A. Datasets

The ICDAR2015 dataset was proposed for the Incidental Scene Text Reading Competition of ICDAR 2015 [25]. Images in this dataset are taken by Google Glasses without limitation on text position, image quality and view point. This dataset is very challenging because text instances could be small, blur and multi-oriented. There are 1000 training images and 500 test images in this dataset, and all of the text regions are labeled with word level quadrangles. We also include 229 training images from the ICDAR2013 dataset in our training set. Therefore, in our experiments, we totally have 1229 training images. Performance of the proposed method is evaluated on the 500 ICDAR2015 test samples.

B. Implementation Details

To optimize the proposed network, the Adam optimizer with an initial learning rate of 1e-4 is used. The learning rate is decayed exponentially with a decay rate of 0.94 and a decay step of 10000. The proposed model is implemented with the Tensorflow framework, and our batch size is set to 4 due to the limitation of GPU memory, instead of 8 used in some other literatures.

C. Evaluation of the Proposed Detector

To demonstrate the effectiveness of our proposed detector, we compare the performance with those of state-of-the-arts. Table I gives details of the comparison results.

As we all know, training data has a great impact on detection performance, so we include additional training sets when other training samples in the datasets are used in addition to the ICDAR2013 and ICDAR2015 datasets. The results tested on multiple scales can always be better than those tested on a single scale. Since many methods only report their results on a single scale, to be fair, we only list single scale results for all of the methods in Tabel I. When multiple settings are tested for certain models, we report their best ones. For example, the model named EAST tests seven settings in [6], but we only take their best performance achieved by PVANET2x on a single scale. Additionally, if the compared method is an end-to-end method, we take their detection-branch-only results in Table I, such as Mask TextSpotter. ICDAR 2015 does not provide any offline evaluation tool or ground truth for the test set. Therefore, we directly submit our prediction results to the online platform (http://rrc.cvc.uab.es/?ch=4com=evaluationtask=1) and take the platform's evaluation results.

From Table I, we can see that the proposed method achieves the best performance among all of the listed detectors with a F-measure of 84.55%. Notably, all of the listed detectors pretrain their models using additional datasets such as ImageNet, SynthText etc., except for PixelLink and ours. To demonstrate the effectiveness of our modification, we also carry out experiments with the original DeepLapv3+ [1] structure, indicated by DeepLab in Table I. DeepLab_small has the same structure and layer setting as DeepLab, but the channels in each layer is shrunk from 256 to 128 (layers with 256 channels in Fig. 1) and from 512 to 256 (layers with 512 channels in Fig. 1). DeepLab and DeepLab_small use the same loss function, data pre-processing strategies, learning rate and optimizer as EAST. The only difference is that EAST uses VGG as the backbone. Clearly, DeepLabv3+ is a better backbone than VGG because the performance is elevated from 78.20% to 82.36%(for DeepLab). Even we use a smaller setting for DeepLab_small, the F-measure is 3.16% higher. Furthermore, the performance of DeepLab_small is 1% lower

than that of DeepLab, so we can conclude that greater setting is good for the improvement of model's performance. Therefore, when compared the performances of different models, both of the structure and the network scale should be taken into consideration. Unfortunately, due to the limitation of our GPU memory, we cannot implement our model with a bigger setting and compare the performance with the methods like IncepText, which has 1024, 2048 and 1024 channels in convolution stage-4, convolution stage-5 and decoder stage respectively, and achieves a performance of 85.3% when a single scale is used.

The method named as DeepLab_MASPP in Table I has the same settings (number of layers and channels in each layer) as the one named as DeepLab, except that DeepLab_MASPP utilizes multiple ASPP layers in the encoder stage and up-samples feature maps with a factor of 2 at the decoder stage. Apparently, when MASPP and smaller up-sample factors are used, the performance can be significantly improved because more smaller text regions are recalled (the recall is improved from 77.80% to 81.08%). Finally, after employing multiple auxiliary IoU losses and auxiliary connections, we obtain a detection performance of 84.55%, which is slight better than that of DeepLab_MASPP. However, DeepLab_MASPP gets the best results of 84.20% at the iteration of 1154k (batch size is set to 4), while after using auxiliary losses and connections, the best results of 84.55% is obtained at the iteration of 734k (batch size is also set to 4). It is evidenced that auxiliary losses and connections are able to greatly assist the training of deep networks in the text detection task, and the discrimination of lower encoding layers can also be enhanced.

V. Conclusion

A powerful backbone is essential to deep networks in the field of computer vision. In this paper, we have firstly introduced the well-known DeepLab structure for the scene text detection task, and achieved promising performance. When detecting text from scene images, encoding the wider range contextual information and detailed information from different scales is able to improve models' robustness to arbitrary text sizes and orientations. Toward this end, we have modified the original DeepLab structure by inserting multiple ASPP layers to the network after feature maps with different resolutions. Additionally, multiple auxiliary IoU losses and connections have been employed to assist the network training and enhance the discrimination ability of lower encoder layers. Experimental results on ICDAR2015 have shown that the performance has been significantly improved by applying proposed modifications.

Acknowledgment

This work was sponsored by xxx.

References

[1] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *the European Conference on Computer Vision*, 2018.

[2] W. He, X. Zhang, F. Yin, and C. Liu, "Deep direct regression for multi-oriented scene text detection," in *IEEE International Conference on Computer Vision*, 2017.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *International Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[5] X. Yin, X. Yin, K. Huang, and H. Hao, "Robust text detection in natural scene images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[6] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[7] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *2016 ACM on Multimedia Conference*, 2016, pp. 516–520.

[8] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," *CoRR*, vol. arXiv preprint arXiv: 1606.09002, 2016.

[9] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: a fast text detector with a single deep neural network," in *AAAI*, 2017, pp. 4161–4167.

[10] M. Liao, B. Shi, and X. Bai, "Textboxes++: a single shot oriented scene text detector," *IEEE Transaction on Image Processing*, vol. 27, pp. 3676–3690, 2018.

[11] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *International Conference on Computer Vision and Pattern Recognition*, 2017.

[12] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: detecting scene text via instance segmentation," in *AAAI*, 2018, pp. 6773–6780.

[13] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "Inceptext: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection," in *IJCAI*, 2018, pp. 1071–1077.

[14] K. He, G. Gkioxari, P. Dollar, and R. Grishick, "Mask r-cnn," *CoRR*, vol. arXiv preprint arXiv: 1703.06870, 2017.

[15] J. Dai, H. Qi, Y. Xiong, and Y. Li, "Deformable convolutional networks," *CoRR*, vol. arXiv preprint arXiv: 1703.06211, 2017.

[16] Y. Li, Y. Yu, Z. Li, Y. Lin, M. Xu, J. Li, and X. Zhou, "Pixel-anchor: a fast oriented scene text detector with combined networks," *CoRR*.

[17] E. Xie, Y. Zang, S. Shao, G. Xu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *AAAI*, 2019.

[18] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[19] ——, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution and fully connected crfs," *IEEE Transaction on pattern analysis machine intelligence*, vol. 40, pp. 834–848, 2017.

[20] L. Chen, G. Papandreous, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. arXiv preprint arXiv: 1706.05587, 2017.

[21] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[22] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *CoRR*, vol. arXiv preprint arXiv: 1703.01086, 2017.

[23] Y. Jiang, X. Wang, S. Yang *et al.*, "R^2cnn: rotational region cnn for orientation robust scene text detection," *CoRR*, vol. arXiv preprint arXiv: 1706.09579v2, 2017.

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

[25] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov *et al.*, "Icdar 2015 competition on robust reading," in *International Conference on Document Analysis and Recognition*, 2015, pp. 1156–1160.