Feature Fusion for Scene Text Detection

Zhen Zhu, Minghui Liao, Baoguang Shi, Xiang Bai*
School of Electronic Information and Communications
Huazhong University of Science and Technology, Wuhan, China 430074
Email: zzhu@hust.edu.cn, mhliao@hust.edu.cn, shibaoguang@gmail.com, xbai@hust.edu.cn

***Abstract*—A significant challenge in scene text detection is the large variation in text sizes. In particular, small text are usually hard to detect. This paper presents an accurate oriented text detector based on Faster R-CNN. We observe that Faster R-CNN is suitable for general object detection but inadequate for scene text detection due to the large variation in text size. We apply feature fusion both in RPN and Fast R-CNN to alleviate this problem and furthermore, enhance model's ability to detect relatively small text. Our text detector achieves comparable results to those state of the art methods on ICDAR 2015 and MSRA-TD500, showing its advantage and applicability.**

***Keywords*-Oriented; Faster R-CNN; Feature fusion.**

I. Introduction

Reading text in the wild develops rapidly recently, driven by a lot of applications such as [1], [2], [3]. It usually consists of two steps including detection and recognition. Scene text detection, localizing text in natural images, is so significant that it is the first bottleneck of a text-reading system.

Scene text detection is a challenging problem due to the large variation in text size. Two-stage detectors [4], [5] adopt a proposal generator, such as region proposal network, to produce sparse candidates. Then, features of the proposals are extracted from a selected layer for classification and regression. Benefiting from the more accurate proposals generated by the region proposal network, two-stage detectors usually achieve higher performance. However, they use the selected feature layer to extract features regardless of the size, which limits their capability of handling object with large variation in size. Small objects are negligible in deep layers, which make it difficult to detect small objects, while large objects are hard to be extracted in shallow layers because of the limited receptive field. Thus, we propose a feature fusion strategy to handle the problem of large variation in text size by fusing the deep layers and the shallow layers.

The main contributions of this paper are summarized as follows: (1) We study the cause that Faster R-CNN is not robust enough to handle large variation in text sizes and propose a method to alleviate this problem. (2) Our proposed method achieves comparable performance on two benchmarks.

*Corresponding author.

II. Related work

A. Scene text detection

Recently, numerous works [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] have been proposed for scene text detection. According to the representation of the detection output, these methods can be roughly divided into two categories: (1) Horizontal text detectors [8], [9], [10], [12], [13] assume that the locations of text in the images can be expressed as horizontal rectangle bounding boxes. (2) Oriented text detectors [6], [11], [14], [15], [16], [17], [18], [22] give more precise bounding boxes to localize the text, such as oriented rectangle bounding boxes, or more precisely, quadrilateral bounding boxes. Our proposed method is an oriented text detector which provides quadrilateral bounding boxes to localize text.

B. Feature fusion

The main idea of dense sampling in methods such as SSD [30] is that different stages of feature maps focus on objects of different scales. Early stages focus on small objects and later stages focus on large objects. Candidate object proposals are densely sampled from multi-stages of feature maps in the network.

Compared to methods with dense sampling of object locations such as YOLO [31], YOLOv2 [32] and SSD [30], we observed that Faster R-CNN has much higher precision but lower recall in scene text detection, which can be attributed to the sparse sampling behaviour of RPN. As known, the original RPN [5] generates proposals based on one single stage of feature maps. In this way, proposals are generated regardless of the text sizes. Therefore, capability in handling the large variation of text size of Faster R-CNN is quite limited.

One possible solution to solve this problem is to fuse the early features and later features together as is discussed in Feature pyramid networks (FPN) [33] in detail. FPN provides an advanced feature fusion architecture via building pyramid feature hierarchy with top-down pathways and skip connections among feature maps. FPN make predictions on each level and verify the accuracy and robustness of their method. However, the implementation of complete integration between FPN and Faster R-CNN is unavailable, so for simplicity, We propose a simple yet well-desgined feature fusion method that can achieve good results as well.

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.60

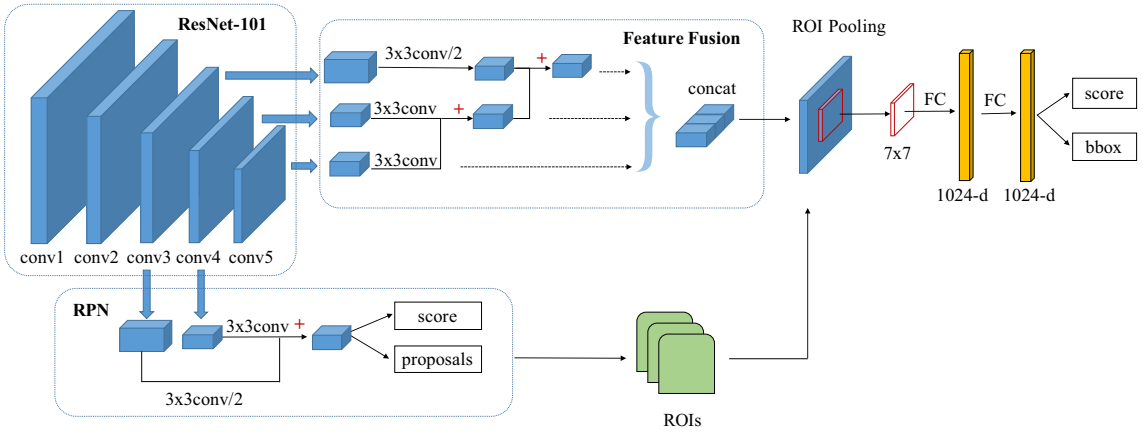

Figure 1. Architecture of the proposed text detection network. "conv" means **conv**olutional layer and "FC" means **F**ully **C**onnected layer. "+" means ***add*** operation upon two different feature maps. "/2" means the feature maps are scaled down by $\frac{1}{2}$ with "stride=2" convolutions. "concat" means **concat**enation of feature maps. Note that feature maps after *conv4* and *conv5* hold the same size for the dilation trick described in Sec.III-A.

III. Methods

A. Network architecture

The architecture of our model is depicted in Fig.1. The main framework is Faster R-CNN [5] and the incarnation is based on ResNet-101 [34] pre-trained on ImageNet [35]. Faster R-CNN is composed by ***RPN*** and ***Fast R-CNN*** (***R-CNN*** for short). RPN is used to generate possible candidate text localizations, namely RoIs (**R**egion **o**f **I**nterest) and R-CNN utilizes these RoIs through RoI Pooling layer as displayed in Fig.1 to regress accurate bounding-boxes. As described in [36], we remove the average pooling layer and the final 1000-class Fully Connected layer (***fc*** for short) in ResNet-101 and modify all convolutional filters with "stride=2" in *Conv5* to "stride=1" while changing those filters with "hole algorithm" [37], [38] to compensate for the reduced stride. In this case, feature maps after conv4 and conv5 hold the same size. And we add 2 *fc* layers following RoI pooling to transform feature maps of into vectors with fixed size. Finally, the model makes classification and regresses final oriented bounding boxes. Note that RPN and R-CNN share features in stages *Conv1*$\sim$*Conv4*.

B. Oriented text detection

Text in natural images is often oriented. However, general object detectors based on CNN are not specially designed to predict oriented bounding boxes. In order to remedy this, we integrate Faster R-CNN to support predicting oriented bounding boxes. The detailed method is described below.

The two procedures in Faster R-CNN can be viewed as roughly detecting object's location (RPN) and then precisely regressing the bounding boxes (R-CNN). So, in our implementation, we only alter R-CNN to regress oriented bounding boxes for precise regression and keep RPN to generate RoIs described as $R = (x_{min}, y_{min}, x_{max}, y_{max})$, which remains the same as original Faster R-CNN.

C. Feature fusion in RPN

In practice, we observe that in scene text detection, Faster R-CNN usually gets higher precision but lower recall as the values of F-measure are maintained the same. It's intuitive to think by improving the quantity and quality of generated proposals, the recall can be enhanced. However, this idea is proved to be not helping as depicted in the ablation study part of Fast R-CNN [4].

Taking inspiration from FPN [33] and SSD [30] etc., We propose a different method to provide RPN with feature maps of different stages. As demonstrated in Fig.1, we combine feature maps after *conv3* and *conv4* together through a 3×3 convolution with "stride=2" on top of *conv3* and a 3×3 convolution on top of *conv4*. Then the generated two feature maps are added up to predict RoI classification scores and regress bounding-boxes of RoIs.

In this way, RPN generate proposals by taking feature maps of early stages into consideration. Note that we do not fuse the features on top of *conv1* and *conv2*. One reason is conv1 and conv2 features may not be semantically strong features. Another reason is that during training, parameters in these stages are not updating, namely the parameters are merely drawn from well-trained models pre-trained from ImageNet. Then these features may not be helpful for scene text classification.

D. Feature fusion in R-CNN

Small text is commonly seen in natural scenes. These text hugely influence the performance evaluation of text detectors. However, it's commonly known that in high level features, information or feature representation of small objects is usually missing. The problem appears common in deep neural networks (DNN).

To make high level features abundant with strong semantics and high resolutions, a common way is to create a feature pyramid via a top-down pathway and skip connections between different stages. In R-CNN, as displayed in Fig.1, we fuse the features of *conv3*, *conv4* and *conv5* to get final high-resolution features containing

strong semantics. Feature maps after *conv3* have twice size as large as feature maps after *conv4* and *conv5*. In order to sum and concatenate them together, we set the stride of the convolution after *conv3* to 2 as the same setting in RPN. After convolutions, the feature maps are sumed up, then fed into concatenate operation to form the final fused feature maps.

Note that, the ***add*** operation can be replaced with ***concat*** operation as well. But in experiments, we found that ***add*** operation accelerate the speed to reach convergence and provides a huge performance improvement.

E. Loss function

For RPN, We adopt the same loss function as Faster R-CNN [5].

For R-CNN, the loss function is slightly changed to make model capable to predict oriented bounding boxes based on the form depicted in Fast R-CNN [4] and Faster R-CNN [5].

Let $t = (t_{x1}, t_{y1}, t_{x2}, t_{y2}, t_{x3}, t_{y3}, t_{x4}, t_{y4})$ be the ground-truth bounding-box regression offsets. To get the ground-truth regression offsets, we denote RoIs as $R = (x_{min}, y_{min}, x_{max}, y_{max})$. For a more detailed interpretation, RoIs can be denoted as $R = [(x_i, y_i), i = 1, 2, 3, 4]$, where $x_1 = x_4 = x_{min}, x_2 = x_3 = x_{max}, y_1 = y_2 = y_{min}, y_3 = y_4 = y_{max}$. In R-CNN procedure, each RoI is attached to a ground truth oriented bounding box written as $G = [(g_{xi}, g_{yi}), i = 1, 2, 3, 4]$. Then R-CNN's bounding-box regression offsets t is calculated by following equations,

$$\begin{aligned} t_{xi} &= (g_{xi} - x_i)/w, \\ t_{yi} &= (g_{yi} - y_i)/h, \end{aligned} \tag{1}$$

where $i = 1, 2, 3, 4$, $w = x_{max} - x_{min}$, and $h = y_{max} - y_{min}$.

We use K to represent the number of ground-truth classes. As we focus on scene text detection, K equals to 1 in all our implementation. Let $p = (p_0, ..., p_K)$ be the probabilities over $K + 1$ classes which is computed by a softmax over $K + 1$ outputs of a fully connected layer, v be the bounding-box regression offsets generated by a sibling layer (see Fig.1 for intuitive understanding) and u be the ground-truth class(text). Then the final multi-task loss function L on each RoI can be denoted as the following equation:

$$L(p, u, v, t) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(v, t), \tag{2}$$

in which $L_{cls}(p, u) = -\log p_u$ is the log loss for true class u and t is the ground-truth bounding-box regression offsets generated by Equation.(1). L_{loc} here means loss for bounding-box regression,

$$L_{loc}(t, v) = \sum_{i \epsilon \{x_i \ or \ y_i, i=1,2,3,4\}} smooth_{L_1}(t_i, v_i), \tag{3}$$

in which $smooth_{L_1}(t_i, v_i)$ is firstly adopted in [4]. In Equation.2, λ is set to 1 in all experiments and $[u \geq 1]$ means only the RoIs of class text are incorporated to the calculation of bounding-box regression loss.

F. Non-maximum suppression

Non-maximum suppression(NMS) is the only post-process operation taken in all experiments. With NMS, we can filter out some bounding boxes with large IoU(Intersection over Union) among each other or low prediction scores. IoU between two oriented bounding boxes are calculated as:

$$\frac{S_{U(o_1,o_2)}}{S_{o_1} + S_{o_2}},$$

where $S_{U(o_1,o_2)}$ implies the area of intersection between two oriented bounding boxes, S_{o_1} and S_{o_2} mean the area of corresponding oriented bounding box.

IV. Experiments

A. Datasets

MSRA-TD500 [6]: It contains 500 natural images taken by pocket cameras from indoor and outdoor scenes, with 300 images for training and 200 for testing. The languages of text in the images are mainly Chinese and English. Therefore, annotations are provided in terms of text line bounding boxes. This dataset has many text instances per image and hugely variant text orientations. We conduct experiments on this dataset to prove our models' powerful ability to handle long text.

ICDAR 2015 Incidental Text (IC15) [39]: It comes from the Challenge 4 of ICDAR 2015 Robust Reading Competition. It has 1,500 images in total, 1,000 for training and 500 for testing. Images in this dataset were taken by Google Glasses in an incidental way. Therefore, the image resolution is not satisfactorily high and text in images exhibits in many orientations. Furthermore, compared to MSRA-TD500, this dataset contains more small text instances and has more scenes that contain scene text with large variation in size. Annotations are provided in terms of word bounding boxes. We conduct experiments on this dataset to verify the effectiveness of our feature fusion strategy for small text and text with large variation in size.

B. Implementation details

Training: In all experiments, our model is optimized with SGD algorithm. Momentum and weight decay are set to 0.9 and 5×10^{-4} respectively. Learning rate is initially set to 1×10^{-3}, and and are multiplied by $\frac{1}{10}$ after training for 12, 35 and 55 epochs. We train models with the shorter side of input size setting to 720 while keeping the longer side below 1280 without changing aspect ratios of images for both IC15 and MSRA-TD500. Throughout the training procedure, we apply OHEM (Online hard example mining) [40] strategy. For experiments conducted on IC15, we initialize corresponding layers with ResNet-101 models pretrained from ImageNet and other new layers are randomly initialized with a Gaussian distribution. For experiments conducted on MSRA-TD500, we fine tune the model adequately trained on IC15. All the experiments are carried out on a PC with 4 Titan Xp GPUs.

Testing: We apply single scale testing and multi-scale testing for IC15 and only single scale testing for MSRA-TD500. As for single scale testing, we input images with the same input size as training. For multi-scale testing, the shorter side of input image is enlarged by (1.0, 1.2, 1.4) times of 720, namely (720, 864, 1008). The maximum length of the longer side is changed accordingly with image aspect ratio remaining the same.

Data augmentation: Neither of IC15 and MSRA-TD500 has enough training data to reach model's capacity to the maximum limits. In order to obtain more training data, we rotate the images in both datasets with the following angles (-90°, -75°, -60°, -45°, -30°, -15°, 15°, 30°, 45°, 60°, 75°, 90°). The ground-truth annotations are changed accordingly. By rotating images, we get 13,000 training images of IC15 and 3,900 training images of MSRA-TD500, respectively.

Anchor ratios and scales: Reasonable design of anchor ratios and scales gives training efficiency and performance gains. We studied aspect ratios and sizes of text bounding-boxes in IC15 and MSRA-TD500. As a result, we get 7 anchor ratios as the following (0.15, 0.2, 0.3, 0.5, 1, 1.5, 2.5) and 5 anchor scales, namely (4, 8, 16, 32, 64). We keep this setting the same in all experiments on IC15 and MSRA-TD500.

Table I
EVALUATION RESULTS ON MSRA-TD500.

MSRA-TD500			
Methods	Recall	Precision	F-measure
Huang et al. [28]	0.68	0.74	0.71
Zhang et al. [11]	0.67	0.83	0.74
He et al. [16]	0.70	0.77	0.74
Yao et al. [41]	0.75	0.77	0.76
Zhou et al. [14]	0.67	0.87	0.76
Shi et al. [15]	0.70	0.86	0.77
Baseline	0.72	0.74	0.73
Baseline+featfusion	0.75	0.78	0.76

Table II
EVALUATION RESULTS ON IC15. **MS** MEANS MULTI-SCALE TESTING.

ICDAR2015			
Methods	Recall	Precision	F-measure
Tian et al. [12]	0.520	0.740	0.610
Shi et al. [15]	0.768	0.731	0.750
Liu et al. [17]	0.682	0.732	0.706
Zhou et al. [14]	0.735	0.836	0.782
Zhou et al. +MS [14]	0.783	0.833	0.807
Hu et al. +MS [19]	0.770	0.793	0.782
He et al. +MS [16]	0.800	0.820	0.810
Baseline	0.629	0.842	0.720
Baseline+featfusion	0.626	0.797	0.701
Baseline+MS	0.706	0.820	0.759
Baseline+featfusion+MS	0.733	0.824	0.776

C. Ablation study

We apply several variants of our model to verify the effectiveness of feature fusion. The tested variants of our model are summarized as follows:
Baseline: architecture based on Faster R-CNN with the integration of oriented bounding-box regression;
Baseline+featfusion: architecture with feature fusion as demonstrated in Fig.1;

1) Feature fusion helps small text detection?: Through the quantitative results shown in Tab.II, our proposed *Baseline+featfusion* model with multi-scale testing has larger F-measure and recall when compared to the *Baseline* model, which verifies the feature-fused architecture helps correctly locating text, especially small text. We visualize some results obtained from IC15 as demonstrated in Fig.3. The results can also proves the model with feature fusion is more capable of detecting small text and giving correct text locations even in very complicated and text-crowded scenes.

2) Feature fusion helps handling large variation in text sizes?: As demonstrated in Tab.I, compared to *Baseline* models, the final recall is improved around 3% with no precision dropping. In Tab.II, similar phenomenon is shown with around 3% improvement of recall and 4% of precision. From the results demonstrated on Fig.3, it can be seen that models with feature fusion can generate more correct proposals with large variation in text sizes. In fact, we assume that the improvement is mainly obtained from the more accurate small text detection.

3) Comparison between single scale test and multi-scale test: From Tab.II, single scale test results of the *Baseline+featfusion* model are lower than the *Baseline* model. We assume the reason should be strong adaptation of the *Baseline+featfusion* model for multi-scale input while degradation for single scale input. This result verifies that our model is robust to multi-scale inputs which contains text with large variation in size.

D. Results on Scene Text Benchmarks

We compare quantitative results on both datasets with other state of the art methods, as depicted in Tab.I and Tab.II. Although we haven't achieved state of the art results, we think it is reasonable because our model is simple yet proved to be useful.

MSRA-TD500: MSRA-TD500 has lots of long and large text which can strongly test model's ability in detecting text with large sizes and aspect ratios. In Tab.I, our proposed method achieves the competitive result on MSRA-TD500, which proves that our model is not only mighty for small text detection, but also robust to detect long and large text.

IC15: In Tab.II, our proposed model achieves the comparable results to state of the art methods on IC15 with F-measure reaching 0.776. The results show that our model is applicable to scene text with large variation in size. Furthermore, our model is robust and accurate in text detection of complicated and various scenes with small amounts of training data (only 1,000 training images actually). We argue that with more training data, our model's performance can be further improved.

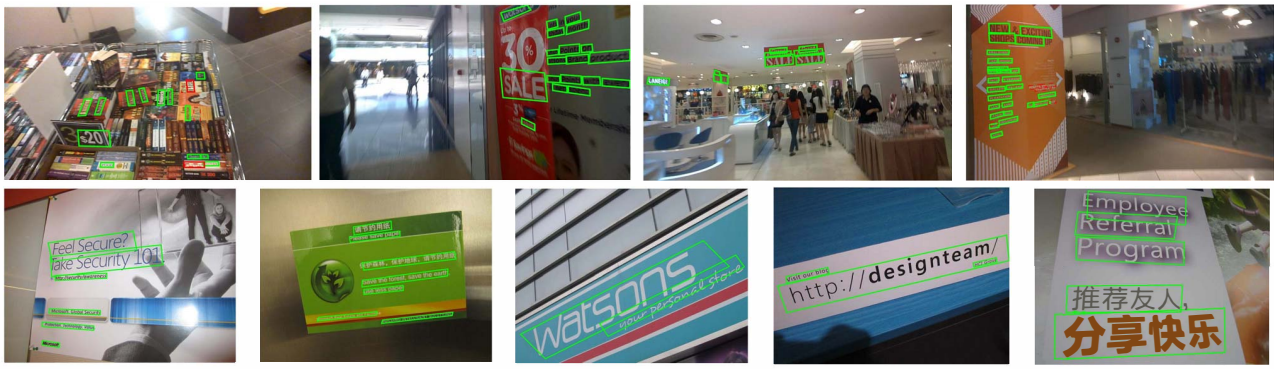

Figure 2. Some visual results on IC15(first row) and MSRA-TD500(second row).

Figure 3. Some visual results on IC15. **Left**: results of *Baseline* model. **Right**: results of the final proposed *Baseline+featfusion* model.

E. Limitations

Although our model is robust to text with long aspect ratios and small sizes, RPN is not powerful enough to give more accurate RoIs compared to other state of the art methods for the comparatively lower recall. The reasons may not only be attributed to lack of enough training data, but also the barely functional design of RPN.

Compared to other state of the art methods, our single scale test results are not satisfying. We believe by elaborately designing the architecture, this problem can be alleviated.

V. Conclusion

We proposed a novel scene text detector based on Faster R-CNN and achieved competitive results on standard benchmarks. We integrated feature fusion in RPN and Fast R-CNN to help detecting text with large variation in size and moreover, improving the detection performance of small text. We verified the effectiveness of our method and we assume this strategy can be used to other models and extended to general object detectors.

In the future, we will look into the architecture design of RPN to produce more accurate proposals. We are also curious about the methods of achieving competitive results with less training data.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No.61733007 and 61573160), and the Program for HUST Academic Frontier Youth Team.

References

[1] X. Rong, C. Yi, and Y. Tian, "Recognizing text-based traffic guide panels with cascaded localization network," in *Proc. ECCV workshop*, 2016.

[2] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proc. ICCV*, 2013.

[3] H. Rajput, T. Som, and S. Kar, "An automated vehicle license plate recognition system," *IEEE Computer*, vol. 48, no. 8, pp. 56–61, 2015.

[4] R. B. Girshick, "Fast R-CNN," in *Proc. ICCV*, 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015.

[6] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. CVPR*, 2012.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[8] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. CVPR*, 2015.

[9] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *IJCV*, vol. 116, no. 1, pp. 1–20, 2016.

[10] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. CVPR*, 2016.

[11] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. CVPR*, 2016.

[12] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*, 2016.

[13] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proc. AAAI*, 2017.

[14] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," in *Proc. CVPR*, 2017.

[15] B. Shi, X. Bai, and S. J. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. CVPR*, 2017.

[16] W. He, X. Zhang, F. Yin, and C. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. ICCV*, 2017.

[17] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. CVPR*, 2017.

[18] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. ICCV*, 2017.

[19] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," in *Proc. ICCV*, 2017.

[20] Y. Wu and P. Natarajan, "Self-organized text detection with minimal post-processing via border learning," in *Proc. ICCV*, 2017.

[21] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proc. ICCV*, 2017.

[22] S. Tian, S. Lu, and C. Li, "Wetext: Scene text detection under weak supervision," in *Proc. ICCV*, 2017.

[23] M. Busta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. ICCV*, 2017.

[24] S. Qin and R. Manduchi, "A fast and robust text spotter," in *Proc. WACV*, 2016, pp. 1–8.

[25] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan, "Text flow: A unified text detection system in natural scene images," in *ICCV*, 2015.

[26] D. He, X. Yang, W. Huang, Z. Zhou, D. Kifer, and C. L. Giles, "Aggregating local context for accurate scene text detection," in *Proc. ACCV*, 2016, pp. 280–296.

[27] X. Rong, C. Yi, and Y. Tian, "Unambiguous text localization and retrieval for cluttered scenes," in *Proc. CVPR*, 2017, pp. 3279–3287.

[28] W. Huang, D. He, X. Yang, Z. Zhou, D. Kifer, and C. L. Giles, "Detecting arbitrary oriented text in the wild with a visual attention model," in *Proc. ACM MM*, 2016, pp. 551–555.

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

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed, "SSD: single shot multibox detector," in *Proc. ECCV*, 2016.

[31] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, 2016.

[32] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[33] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[35] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009.

[36] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Proc. NIPS*, 2016, pp. 379–387.

[37] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 640–651, 2017.

[38] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.

[39] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proc. ICDAR*, 2015.

[40] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. CVPR*, 2016, pp. 761–769.

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