STELA: A Real-Time Scene Text Detector With Learned Anchor

LINJIE DENG[1], YANXIANG GONG[1], XINCHEN LU[1], YI LIN[2], ZHENG MA[1], AND MEI XIE[1]
[1]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610000, China

Corresponding author: Mei Xie (mxie@uestc.edu.cn)

ABSTRACT To achieve high coverage of target boxes, a normal strategy of conventional one-stage anchor-based detectors is to utilize multiple priors at each spatial position, especially in scene text detection tasks. In this work, we present a simple and intuitive method for multi-oriented text detection where each location of feature maps only associates with one reference box. The idea is inspired from the two-stage R-CNN framework that can estimate the location of objects with any shape by using learned proposals. The aim of our method is to integrate this mechanism into a one-stage detector and employ the learned anchor which is obtained through a regression operation to replace the original one into the final predictions. Based on RetinaNet, our method achieves competitive performances on several public benchmarks with a totally real-time efficiency (26.5*fps* at 800*p*), which surpasses all of anchor-based scene text detectors. In addition, with less attention on anchor design, we believe our method is easy to be applied on other analogous detection tasks. The code is publicly available at https://github.com/xhzdeng/stela.

INDEX TERMS Scene text detection, real-time detector, learned anchor.

I. INTRODUCTION

Text in scene usually conveys valuable semantic information. Thus, detecting text in natural images has recently attracted increasing attention in computer vision community cause perceiving information is a critical part of artificial general intelligence. It has been widely used in various applications such as multilingual translation, automotive assistance and image retrieval. Previous works [1], [2], [13], [17], [21], [22] have been dominated by sliding windows or connected component with hand-crafted feature, which divided the task into a sequence of distinct steps and utilized bottom-up strategy to search characters and words. Although these methods have shown their promising performances, they may be restricted to complex situations due to the diversity of text instances and undesirable image quality.

With the astonishing progress for object detection by exploring the powerful deep learning technology [34], recent methods take text as a specific object and extend the general object detection frameworks [11], [14], [20] to hypothesize word or text locations. Those approaches can be divided into two major groups: two-stage proposal-driven and one-stage proposal-free method. Although two-stage framework

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

[12], [24], [40] consistently achieves top accuracy on the public benchmarks [6]–[8], recent works [9], [23], [29], [35] based on one-stage frameworks also demonstrate yielding faster text detectors with comparable accuracy.

Unlike two-stage detector who can classify boxes at any position and shape by using learned proposals [14] and region pooling operation [26], one-stage detectors heavily rely on how densely the anchors cover the space of possible target locations [32]. A popular approach for achieving high coverage is to use multiple anchors to cover boxes of various scales and aspect ratios, especially in the tasks of scene text detection. TextBoxes++ [9] was based on SSD [11] and defined 7 specific aspect ratios (including 1, 2, 3, 5, 1/2, 1/3 and 1/5) for default boxes on each location of feature maps. In order to achieve multi-oriented text detection, DMPNet [29] added several rotated anchors, for a total of 12 (6 regular and 6 inclined) to find the best match to arbitrary-oriented text instance. Instead of choosing priors by hand, DeepTextSpotter [35] followed YOLOv2 [31] runs k-means clustering ($k = 14$) on the training set bounding boxes to automatically find suitable priors.

Given the anchor design of the above detectors, a natural question to ask is: could we decrease the number of anchors and maintain similar accuracy? This changing will bring twofold benefit: reducing manual attention on anchors
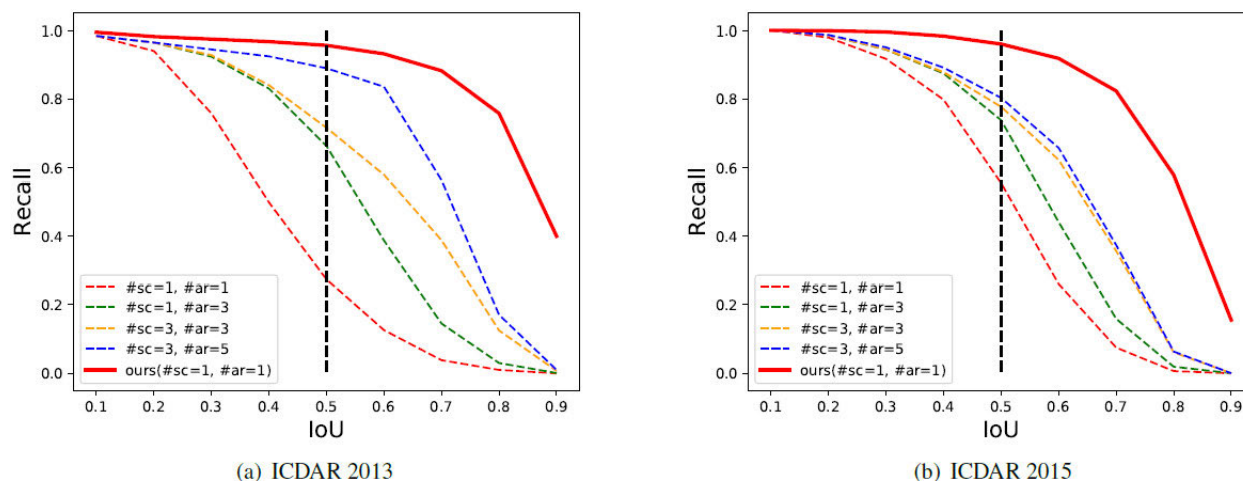
FIGURE 1. The recall rates of different anchor designs on ICDAR 2013(a) and 2015(b). "#sc" means number of scales, "#ar" means number of aspect ratios. The black line represents the IoU threshold (0.5) which is usually used in training stage to discriminate foreground and background. The other dashed lines with colors represent different anchor designs with various number of scales and aspect ratios. The solid line represent the recall rates of learned anchor which proposed in this work. The performances of each design will be shown in Section.III-B.

and improving efficiency at inference stage. First, the shapes and scales of anchors have to be predefined for different tasks, and this must be careful because a wrong design may harm the performance of detection [30]. Second, most anchors correspond to false candidates which are irrelevant to the targets, and meanwhile a large number of anchors can lead to significant computational cost when the network involves heavy heads. Besides, although not mentioned in many papers, the anchor generation usually needs to cost a certain amount of time.

Being attracted to simple network architecture and high computational efficiency, in this work, we investigate the issue of anchor design within one-stage detector which we mentioned above for multi-oriented text detection. In one-stage methods, the optimization target in training and the prediction reference in testing are both based on the coverage between original anchors and target boxes. Then, the quality of those prior boxes has a critical impact on the performances of a detector. Normally, as the number of anchors increases, the coverage of targets increases, but it will still be saturated in some situations, as shown in Figure 1. Therefore, we need to find a better way to choose priors that make it easier for the network to learn to predict better detection. Inspired from the learned proposal mechanism [14] in the two-stage R-CNN framework, we intend to utilize the learned anchor which is obtained through a regression operation to replace the original one into the final predictions. It is worth noting that unlike region proposal network (RPN) in two-stage detector which can reduce the number of possible locations down to one or two thousands, we still maintain the original quantity of anchors and keep the rest parts of one-stage detector's architecture. To validate its effectiveness, we adopt the state-of-the-art RetinaNet [32] as our baseline model and present a simple and intuitive text detector named STELA (**S**cene **TE**xt Detector with **L**earned **A**nchor), in which each

location of feature maps only associates with one anchor. Following the standard evaluation protocols in each benchmark, our method achieves comparable performances with an F-measure 0.887 on ICDAR 2013 [6], 0.833 on ICDAR 2015 [7] and 0.715 on ICDAR 2017 MLT [8]. Besides, our method is a totally real-time scene text detector with $26.5fps$ at $800p$, which surpasses all of anchor-based methods. At last, with less attention on anchor design, we believe our method is easy to be applied on other analogous detection tasks. Also, all of our training and testing code is publicly available at https://github.com/xhzdeng/stela now.

II. METHODOLOGY

A. ONE-STAGE OBJECT DETECTION

In this section, we first review the one-stage detection pipeline. OverFeat [36] is one of the first modern one-stage object detector based on deep neural networks. More recent SSD [11] and YOLOv2 [31] have renewed interest in one-stage methods. The key idea of them is to associate a set of pre-defined anchors which are centered at each location of feature maps and make final predictions based on those reference boxes [27]. As shown in Figure 2.a, it basically contains a backbone network for feature extraction over the entire image and two parallel sub-networks following, one for predicting the probability distribution over multiple categories of each anchor and another for regression the offset from each positive candidate to a nearby ground-truth box, if one exists.

Comparing with two-stage R-CNN (Figure 2.b) methods, one-stage detectors skip the region proposal generation step and give final predictions (classification and regression) based on original anchors directly. However, its detection accuracy is usually behind that of two-stage approaches, one of the main reasons is they must process a much larger set of candidate object locations regularly sampled across an image.
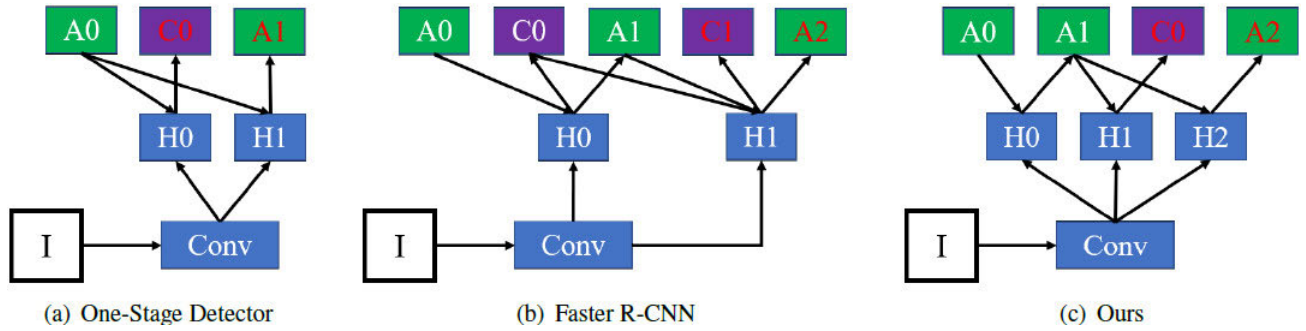
FIGURE 2. The architectures of different frameworks. "I" input image, "Conv" backbone convolutions, "H" network head, "C" classification. "A0" are the original anchors in all architectures, "A1" in (b) and (c) represent the selected proposals and learned anchors respectively. "A" and "C" in red color represent the final outputs of each detector.

The extreme foreground-background class imbalance problem will encounter during training phase and hamper the resulting performance. More recently, RetinaNet [32] proposed focal loss (FL) to address the class imbalance problem that one-stage detectors are able to match the accuracy of existing two-stage ones. The focal loss is modified from standard cross entropy (CE) loss:

$$CE(p, y) = \begin{cases} -log(p) & \text{if } y = 1 \\ -log(1 - p) & \text{otherwise} \end{cases} \quad (1)$$

In the above $y = 1$ specifies the ground-truth class and $p \in [0, 1]$ is the probability. Normally, we define p_t:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \quad (2)$$

and rewrite $CE(p, y) = -log(p_t)$. Then, the classification loss is defined as:

$$L_{cls} = FL(p_t) = -\alpha_t (1 - p_t)^\gamma log(p_t) \quad (3)$$

where α_t is a *balanced* weighting factor and γ is a *focusing* parameter. It applies a modulating term to the cross entropy loss in order to focus learning on hard examples and down-weight the numerous easy negatives. In our implementation, we follow the original focal loss that set $\alpha_t = 0.25$ and $\gamma = 2.0$.

B. ROTATED BOUNDING BOX REGRESSION
As depicted in [3], using rectangular bounding boxes to localize multi-oriented text may result in redundant background noise and unnecessary overlap. Thus, we adopt rotated rectangular boxes to match arbitrary-oriented text instances. Each bounding box b is represented by a five tuple $b = (x, y, w, h, \theta)$, where x, y are the center point, w, h are width and height, θ is the angle to horizontal. The task of the regression operation is to predict the distance of each item from a positive anchor to the nearby ground-truth. Normally, to encourage a regression invariant to scale and location, the distance vector $\Delta = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\theta)$ is defined by:

$$\delta_x = (g_x - b_x)/b_w, \quad \delta_y = (g_y - b_y)/b_h \quad (4)$$

$$\delta_w = log(g_w/b_w), \quad \delta_h = log(g_h/b_h) \quad (5)$$
$$\delta_\theta = tan(g_\theta) - tan(b_\theta) \quad (6)$$

where b and g represent a bounding box and its target ground-truth respectively. The regression task loss L_{loc} is calculated by regression target Δ_t and predicted tuple Δ_p

$$L_{loc} = smooth_{L_1}(\Delta_t - \Delta_p) \quad (7)$$

where $smooth_{L_1}$ is a robust L_1 loss defined in [26]. Usually, for improving the effectiveness of multi-task learning, Δ is normalized by its mean and variance. In our experiments, the mean is set to (0, 0, 0, 0, 0) and the variance is set to (0.1, 0.1, 0.2, 0.2, 0.1).

C. LEARNED ANCHOR
Normally, the detector needs to search the true positives from thousands anchors and adjust the shapes and locations to make them tighter on the targets. It is difficult to determine if a bounding box b is a positive candidate cause it usually includes an object and some amount of the background. In practice, this is solved by the IoU metric between box b and most nearby ground-truth g. Commonly, the threshold μ is a constant set to 0.5. If the IoU is above the threshold μ, bounding box b is considered to be an example of positive.

$$y = \begin{cases} 1, & IoU(b, g) \geq \mu \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Also, $y = 1$ specifies the ground-truth class. It is worth noting that conventional IoU based on rectangular boxes is unsatisfactory for our task, thus we modify it to compute the overlaps for rotated rectangles. Given all this, the optimization target in training is determined by the overlaps between original anchors with ground-truth boxes.

However, the original anchors with fixed scales and aspect ratios which are pre-defined manually may not be the optimal designs. Compared with one-stage detector, we argue that the most important part of proposal scheme in two-stage is that the selected proposals are chosen by learning. That makes two-stage method able to reduce the search space of targets, and meanwhile optimize the quality of candidates.

FIGURE 3. The view of different boxes. The blue, red and green boxes represent original anchor, learned anchor and final output box respectively. It is worth noting that the learned anchor (red) shares the same central point with the original one (blue). Viewing digitally with zooming is recommended.

Inspired by this, we intend to integrate this mechanism into the one-stage detectors. We simply add an extra regression branch for anchor refining and utilize learned one into the final classification and regression, as shown in Figure 2.c.

Especially, the regression targets of learned anchor is not arbitrary. As refer in [30], one of general rules for a reasonable anchor design is alignment. To use convolutional features as anchor representations, the center of an anchor needs to be well aligned with feature map pixels. Towards this end, we only regress the offsets within $\Delta' = (\delta'_w, \delta'_h, \delta'_\theta)$, and this will keep anchors still align with feature map, as depicted in Figure 3. Following regression task, the anchor refining loss L_{ref} is defined as:

$$L_{ref} = smooth_{L_1}(\Delta'_t - \Delta'_p) \qquad (9)$$

Unlike two-stage R-CNN method filter anchors with an objectness score, we only adjust the shape of each anchor and keep the quantity of anchors here. Evaluated on public benchmarks, we find comparing with original ones, the coverage of targets will be given a huge enhancement after anchor refining stage, as shown in Figure 1.

D. NETWORK ARCHITECTURE
For the trade-off between efficiency and accuracy, all of our experiments are implemented on RetinaNet [32] with ResNet-50 [5] as backbone, though other networks are still applicable. We also adopt the Feature Pyramid Network (FPN) from [25] to construct a rich, multi-scale feature pyramid from a single resolution input image. The FPN consists of levels P_3 to P_7 feature maps, and the corresponding base anchor sizes from 16^2 to 256^2 for detecting small text instances (32^2 to 512^2 in source implementation). In original RetinaNet, the two sub-networks (heads) are deeper with 5 convolutional layers. For improving the running speed, we decrease the number of layers from 5 to 2 for streamlining the heads. This may result in a slight accuracy loss, but will give us a real-time text detector in return. Based on the above definitions, the model is trained to simultaneously minimize the losses on anchor refining, final regression and classification. Overall, the loss function is a weighted sum of three losses

$$L = \lambda_{ref} L_{ref} + \lambda_{loc} L_{loc} + \lambda_{cls} L_{cls} \qquad (10)$$

TABLE 1. The impact of the different anchor designs. "#sc" number of scales, "#ar" number of aspect ratios. "la" means learned anchor. All input images are resize to 800 pixels.

#sc	#ar	la	ICDAR 2013	ICDAR 2015	FPS
1	1		0.447	0.621	**27.3**
1	3		0.789	0.742	26.3
3	3		0.788	0.742	24.1
3	5		0.863	0.753	22.3
1	1	✓	**0.887**	**0.833**	26.5

TABLE 2. The impact of the number of stages. "#stage" means the number of anchor refining stages.

#stages	ICDAR 2013	ICDAR 2015	FPS
1	0.887	**0.833**	**26.5**
2	**0.889**	0.830	25.1
3	0.879	0.827	23.7

where $\lambda_{ref}, \lambda_{loc}, \lambda_{cls}$ are user constants indicating the relative strength of each component defined above. In order to keep the balance of different loss types, we set them to 0.5, 0.5, 1.0 respectively.

III. EXPERIMENTS
A. IMPLEMENTATION DETAILS
The backbone of network is initialized by the model trained on ImageNet [15] for classification task and other layers are initialized by following [32]. The network is trained with Adam [39] optimizer. Restricted by the hardware, the batch size is set to 4 and the initial learning rate is set to 10^{-4}. We randomly pick up 100,000 images from SynthText [4] to pretrain the network for 5 epochs, and collect real data from ICDAR 2013 [6], 2015 [7] and 2017 [8] to finetune a final model for 25 epochs. The learning rate is decayed to 10^{-5} after 15 epochs of finetuning. We use the multi-scale training scheme that randomly resize the input size between 480 and 800. Random flipping is also used for data augmentation.

Specially, in order to capture more regression target candidates, we set the IoU threshold to 0.3 in anchor refining training. In the inference stage, a confidence threshold with 0.3 and a non-maximum suppression threshold with 0.3 are applied to yield the final outputs. The proposed method is

TABLE 3. Results on ICDAR 2013 Focused Scene Text. "*" means multi-scale test. "–" means no report in their papers.

	ICDAR Standard			DetEval			FPS
	Recall	Precision	F-measure	Recall	Precision	F-measure	
TextBoxes++ [9]	0.740	0.860	0.800	0.740	0.880	0.810	11.6
CTPN [18]	0.730	0.930	0.820	0.820	0.930	0.880	7.1
FCRN [4]	0.764	**0.938**	0.842	0.755	0.920	0.830	0.8
DeepText [24]	0.830	0.870	0.850	–	–	–	0.6
SegLink [16]	–	–	–	0.830	0.877	0.853	20.6
Lyu et al. [33]	0.794	0.933	0.858	–	–	–	10.4
SSTD [23]	0.860	0.880	0.870	0.860	0.890	0.880	7.7
EAST [19]	0.828	0.926	0.874	–	–	–	16.8
CRPN [3]	0.839	0.919	0.876	0.840	0.921	0.879	9.1
FOTS [19]	–	–	0.882	–	–	0.883	23.9
Ours	0.849	0.927	0.887	0.851	0.933	0.890	**26.5**
Lyu et al.* [33]	0.844	0.920	0.880	–	–	–	1.0
TextBoxes++* [9]	0.840	0.910	0.880	0.860	0.920	0.890	2.3
RRD* [10]	0.860	0.920	0.890	–	–	–	–
Ours*	**0.896**	0.937	**0.916**	**0.893**	**0.938**	**0.915**	10.5

implemented by using PyTorch[1] and all experiments are carried out on a standard PC with Intel i7-6800k and a single NVIDIA TITAN Xp.

B. ABLATION STUDY

To investigate the effectiveness of our method, we conduct several ablation studies. Each model is evaluated on ICDAR 2013 [6] and 2015 [7] benchmarks.

1) ANCHOR DESIGN

We first investigate the impacts of different anchor designs on performances, including accuracy and efficiency. The baseline models are directly extended from RetinaNet by simply changing the regression strategy introduced in Section.II-B. The aspect ratios of anchors on single location of feature maps are simply selected from {0.25, 0.5, 1, 2, 4}. Also, the scales are chosen from ($2^{k/3}$, $k < 3$). As shown in Table.1, with the anchor increasing, the F-measure improved from 0.447 to 0.863 on ICDAR 2013, but no significant improvement (0.621 to 0.753) on ICDAR 2015. Analyzing from Figure 1, we argue that the coverage of targets gets saturated on ICDAR 2015, but not on ICDAR 2013. That proves once again the most important design factor in a one-stage detector is how densely it covers the space of target boxes.

Attaching anchor refining operation on the first baseline, the resulting model obtains a huge improvement (shown in Figure 1) in recall rates on both benchmarks, with great progresses on accuracy from 0.447 to 0.887 on ICDAR 2013 and 0.621 to 0.833 on ICDAR 2015. This strongly demonstrates the effectiveness of our approach. In addition, we also assess other baselines with anchor refining, there is only slower running speed, but no obvious improvement.

2) NUMBER OF STAGES

Like Cascade R-CNN [37], we add more stages of anchor refining to compare the influences. We also increase the IoU threshold of each refining stage by following Cascade R-CNN. The results are summarized in Table.2. Increasing

[1] https://pytorch.org/

TABLE 4. Results on ICDAR 2015 Incidental Scene Text. All results of works are reported with single testing scale. "–" means no report in their papers.

	Recall	Precision	F-measure	FPS
DMPNet [29]	0.680	0.730	0.710	–
SegLink [16]	0.768	0.731	0.750	8.9
SSTD [23]	0.730	0.800	0.770	7.7
EAST [19]	0.735	0.836	0.782	13.2
RRPN [12]	0.770	0.840	0.800	3.0
TextBoxes++ [9]	0.707	**0.941**	0.807	3.6
Lyu et al. [33]	0.767	0.872	0.817	11.6
RRD [10]	0.790	0.856	0.822	6.5
FOTS RT [38]	0.798	0.856	0.828	22.6
CRPN [3]	**0.807**	0.887	**0.845**	5.0
Ours	0.786	0.887	0.833	**26.5**

more refining stages will not lead to significant improvement, or even accuracy decrease. Besides, adding the number of stages will affect the running speed. Therefore, one refining stage is the best choice for our method.

C. COMPARISON TO STATE OF THE ART

We evaluate our method on several public benchmarks and compare to recent state-of-the-art methods. Figure 4 shows some detection results from each dataset.

ICDAR 2013 [6] dataset consists of 229 training and 233 testing images which were captured by user explicitly detecting the focus of the camera on the text content of interest. It is a standard benchmark for evaluating horizontal or nearly horizontal text detection. In this benchmark, we set the scale of input images to 800 for single-scale testing. We also evaluate on multi-scale testing which the scales are set to 320, 480, 640 and 800. As depicted in Table. 3, the proposed method outperforms all anchor based methods including DeepText [24], FCRN [4] and CTPN [18], which are mainly designed for nearly horizontal text detection. For single-scale testing, our method achieves a totally real-time running speed at 26.5 fps. Even the multi-scale, our method also runs at a speed of 10.5 fps. Compared with recent methods [3], [10], [33], [38], our method is comparable with accuracy and efficiency.
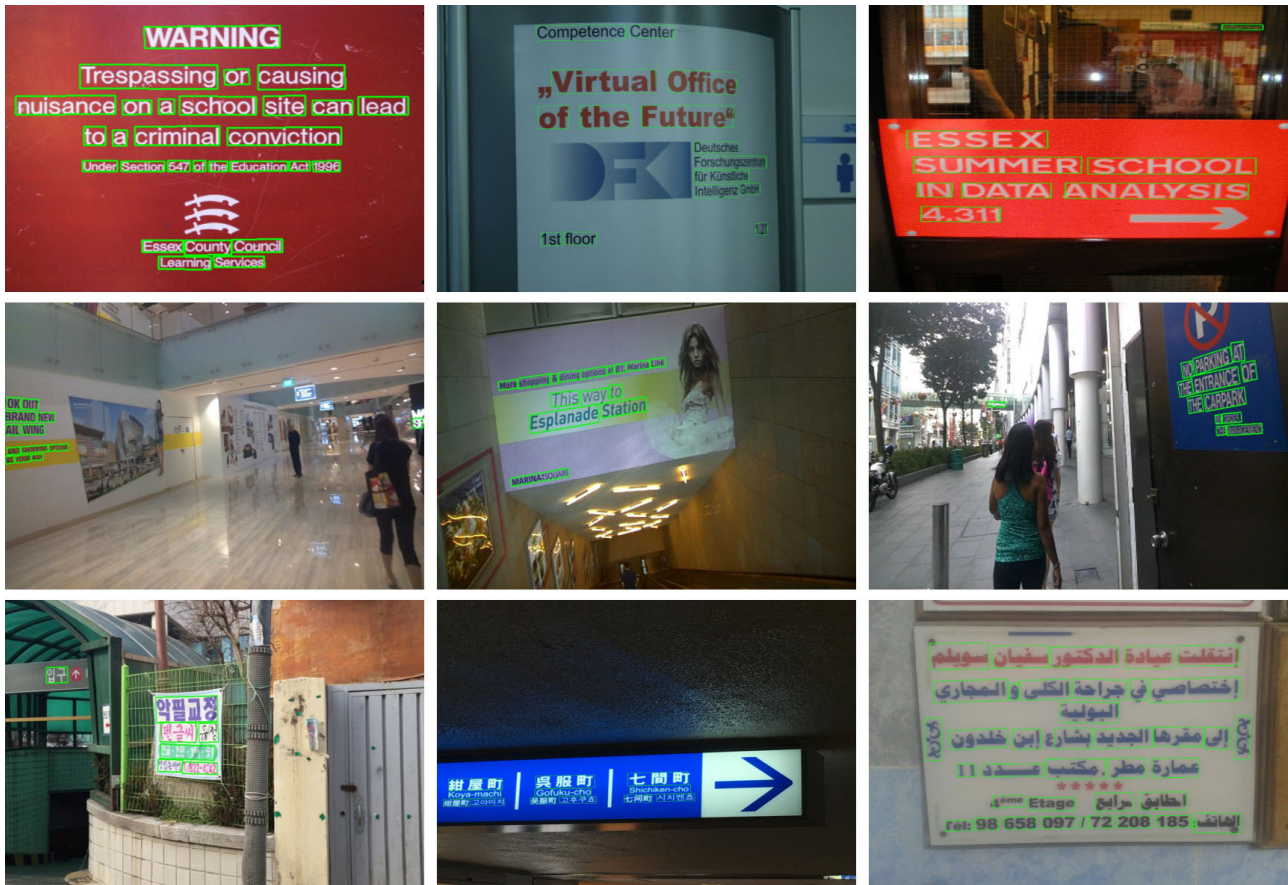
FIGURE 4. Selected results from the public benchmarks. Viewing digitally with zooming is recommended.

TABLE 5. Results on ICDAR 2017 Multi-lingual Scene Text Detection. All results of works are reported with single testing scale. "†" means the result is obtained from the ICDAR 2017 MLT leaderboard.

	Recall	Precision	F-measure
SARI_FDU_RRPN_v1†	0.555	0.711	0.623
Sensetime OCR†	**0.694**	0.569	0.625
SCUT_DLVClab1†	0.545	0.802	0.648
Lyu et al. [33]	0.556	**0.838**	0.668
FOTS [38]	0.575	0.809	0.672
Border [38]	0.621	0.777	0.690
SPCNET [28]	0.669	0.734	0.700
Ours	0.655	0.787	**0.715**

ICDAR 2015 [7] benchmark was released during the ICDAR 2015 Robust Reading Competition. It provides 1000 training and 500 testing images which were collected without taking any specific prior attention. It was designed for multi-oriented text detection, so all images are annotated with word-level quadrangles. To evaluate the adaptability of our learned anchor, we still set the input size to 800 pixels. As shown in Table.4, our method achieves an F-measure of 0.833, which also surpasses all of the anchor-based methods [9], [12], [16], [23], [29], including one-stage and two-stage frameworks. Compared with other approaches [19], [38] which utilize a deep regression network that directly predict text region, our method still keep an absolute lead in running speed.

ICDAR 2017 MLT [8] is a large scale multi-lingual text dataset, which includes 7200 training, 1800 validation and 9000 testing images with 9 languages. It was proposed for verifying the generalization ability of each method. Therefore, it is more difficult than previous ICDAR challenges. Due to a larger number of small text instances in this dataset, we enlarge the scale of testing image by 2 times to 1600 pixels and our method achieves 0.655, 0.787, 0.715 in recall, precision and F-measure by using the online evaluation system provided officially, as shown in Table.5. The presented results demonstrate that our method is capable of applying practically in multi-lingual text detection.

IV. CONCLUSION AND FUTURE WORK

In this work, we propose a simple and intuitive method based on RetinaNet for multi-oriented text detection where each location of feature maps associate with only one anchor. The aim of our method is to integrate the learning mechanism from two-stage R-CNN framework into the one-stage detector and utilize the learned anchor to replace the original one into the final predictions. Experimental results on public benchmarks confirm that the proposed method is capable of achieving comparable performance with state-of-the-art methods. Besides, it is a total real-time scene text detector.

In the future, we are interested in integrating the detector with a text recognizer to consist an end-to-end text reading system. In addition, we also plan to evaluate it on other detection tasks to prove the universality of our approach.

REFERENCES

[1] M. Buta, L. Neumann, and J. Matas, "FASText: Efficient unconstrained scene text detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1206–1214.

[2] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.

[3] L. Deng, Y. Gong, Y. Lin, J. Shuai, X. Tu, Y. Zhang, Z. Ma, and M. Xie, "Detecting multi-oriented text with corner-based region proposals," *Neurocomputing*, vol. 334, pp. 134–142, Mar. 2019.

[4] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2315–2324.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[6] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[7] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, F. Shafait, S. Uchida, E. Valveny, and S. Lu, "ICDAR 2015 competition on robust reading," in *Proc. IEEE 13th Int. Conf. Document Anal. Recognit.*, Aug. 2015, pp. 1156–1160.

[8] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khlif, M. M. Luqman, J. Burie, C. Liu, and J.-M. Ogier, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification," in *Proc. IEEE 14th Int. Conf. Document Anal. Recognit.*, Aug. 2017, pp. 1454–1459.

[9] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[10] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5909–5918.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[12] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[13] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1872–1885, Sep. 2016.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[16] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2550–2558.

[17] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4651–4659.

[18] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 56–72.

[19] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5551–5560.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[21] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[22] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.

[23] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3066–3074.

[24] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A unified framework for text proposal generation and text detection in natural images," 2015, *arXiv:1605.07314*. [Online]. Available: https://arxiv.org/abs/1605.07314

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 936–944.

[26] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2015, pp. 1440–1448.

[27] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Consistent optimization for single-shot object detection," 2019, *arXiv:1901.06563*. [Online]. Available: https://arxiv.org/abs/1901.06563

[28] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," 2018, *arXiv:1811.08605*. [Online]. Available: https://arxiv.org/abs/1811.08605

[29] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Mar. 2017, pp. 3454–3461.

[30] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," 2019, *arXiv:1901.03278*. [Online]. Available: https://arxiv.org/abs/1901.03278

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6517–6525.

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[33] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[35] M. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2223–2231.

[36] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229

[37] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[38] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[40] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R^{2}CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: https://arxiv.org/abs/1706.09579

LINJIE DENG received the M.S. degree from Sichuan University, in 2015. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include the application of computer vision and machine learning.
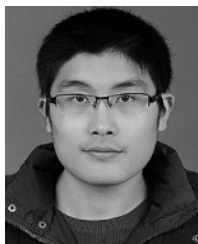
YANXIANG GONG is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include pattern recognition, computer vision, and deep learning.

XINCHEN LU is currently pursuing the M.S. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include object recognition and tracking.

YI LIN received the Ph.D. degree from Sichuan University, Chengdu, China, in 2019. He is currently an Assistant Professor with Sichuan University. He is also a Visiting Scholar with the University of Wisconsin–Madison. His research interests include air traffic flow management, machine learning, and deep learning applications on air traffic management.

ZHENG MA is currently a Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include convex optimization, computer vision, and image processing.

MEI XIE received the Ph.D. degree in signal and information processing (SIP) from UESTC, in 1997, and the postdoctoral degree from the University of Hong Kong and University of Texas, from 1997 and 1999, respectively. She is currently a Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). Her research interests include pattern recognition, computer vision, and artificial intelligence.

• • •