International Journal on Document Analysis and Recognition (IJDAR) (2019) 22:315–327
https://doi.org/10.1007/s10032-019-00335-y

SPECIAL ISSUE PAPER

An anchor-free region proposal network for Faster R-CNN-based text detection approaches

Zhuoyao Zhong[1] · Lei Sun[2] · Qiang Huo[2]

Received: 14 November 2018 / Revised: 27 February 2019 / Accepted: 15 July 2019 / Published online: 25 July 2019

Abstract

The anchor mechanism of Faster R-CNN and SSD framework is considered not effective enough to scene text detection, which can be attributed to its Intersection-over-Union-based matching criterion between anchors and ground-truth boxes. In order to better enclose scene text instances of various shapes, it requires to design anchors of various scales, aspect ratios and even orientations manually, which makes anchor-based methods sophisticated and inefficient. In this paper, we propose a novel anchor-free region proposal network (AF-RPN) to replace the original anchor-based RPN in the Faster R-CNN framework to address the above problem. Compared with the anchor-based region proposal generation approaches (e.g., RPN, FPN–RPN, RRPN and FPN–RRPN), AF-RPN can get rid of complicated anchor design and achieves higher recall rate on both horizontal and multi-oriented text detection benchmark tasks. Owing to the high-quality text proposals, our Faster R-CNN-based two-stage text detection approach achieves the state-of-the-art results on ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013 text detection benchmark tasks by only using single-scale and single-model testing.

Keywords Scene text detection · Anchor · Anchor-free · Region proposal generation · Faster R-CNN

1 Introduction

Scene text detection has attracted considerable interests from computer vision and document analysis communities recently [1–4] owing to the increasing demands for many content-based visual intelligent applications, e.g., image and video retrieval, scene understanding and target geolocation. However, because of diverse text variabilities in colors, fonts, orientations, languages and scales, extremely complex and text-like backgrounds, as well as some distortions and artifacts caused by image capturing like non-uniform illumination, low contrast, low resolution and occlusion, text detection in natural scene images is still an unsolved problem.

Nowadays, with the astonishing development of deep learning, state-of-the-art convolutional neural network (CNN)-based object detection frameworks, such as Faster R-CNN [5] and SSD [6], have been widely used to address the text detection problem and outperform substantially traditional MSER- [7] or SWT-[8] based bottom-up text detection methods. However, Faster R-CNN and SSD are found to be not flexible enough for text detection because of their anchor (called default box in SSD) mechanism [9]. Anchors are used as reference boxes in both Faster R-CNN and SSD to predict the corresponding region proposals or target objects, and the label of each anchor is determined by its Intersection-over-Union (IoU) overlap with ground-truth bounding boxes [5]. If we want an object to be detected, there should be at least one anchor which has a high enough IoU overlap with this object. So, to achieve high recall, anchors with various scales and shapes should be designed to cover the scale and shape variabilities of objects in images. As scene text instances have wider variability in scales, aspect ratios and especially orientations than general objects, it requires much more complicated anchor design, i.e., more scales, aspect ratios and orientations [10–13], which makes anchor-

This work was done when Z. Zhong was an intern in Speech Group, Microsoft Research Asia, Beijing, China.

✉ Zhuoyao Zhong
zhuoyao.zhong@gmail.com

Lei Sun
lsun@microsoft.com

Qiang Huo
qianghuo@microsoft.com

[1] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

[2] Microsoft Research Asia, Beijing, China

based methods sophisticated and inefficient. Recently, the idea of DenseBox [14] is borrowed to overcome this problem in some text detection methods [9,15], which use a fully convolutional neural network (FCN) [16] to directly output the pixel-wise textness scores and bounding boxes of the concerned text instances through all locations and scales of an image. Although more flexible, the capabilities of these approaches are limited. For example, they cannot detect long or large text instances robustly, which occur very often in “Multilingual scene text detection” scenarios [4], as the maximal size of text instances that can be handled by the detector is limited by the receptive field (*RF*) size of the used convolutional feature maps.

To overcome the above problems, we propose incorporating the “anchor-free” idea of DenseBox into the Faster R-CNN framework. Specifically, we propose a novel anchor-free region proposal network (AF-RPN) to replace the original anchor-based RPN so that our Faster R-CNN-based text detector can possess high flexibility and high capability at the same time. As illustrated in Fig. 1, each pixel in a specific convolutional feature map can be mapped to a point (called a sliding point hereinafter) in the raw image. For each sliding point that locates in a text core region (points within the solid line oriented rectangle in Fig. 1b), AF-RPN directly predicts the offsets from it to the bounding box vertices of the concerned text instance (Fig. 1c). In this way, AF-RPN can generate high-quality inclined text proposals directly in an anchor-free manner, which can get rid of complicated hand-crafted anchor design. Moreover, the label definitions for sliding points in AF-RPN are much easier than IoU-based label definitions for anchors in the original RPN, where we only need to determine whether a sliding point is inside any ground-truth bounding box’s core region. Compared with DenseBox-based text detectors, Faster R-CNN-based text detectors can deal with long or large text instances much more effectively. This is because the ROI pooling algorithm in the second-stage Fast R-CNN can enlarge the *RF* size of pooled features for each proposal significantly, which can improve not only the bounding box regression precision of long or large text instances, but also the text/non-text classification accuracy. Furthermore, unlike DenseBox, we let AF-RPN extract text proposals from multi-scale feature maps of feature pyramid network (FPN) [17] in a scale-friendly manner so that AF-RPN can be more robust to large text scale variance. Thanks to this, our text detector can achieve superior text detection performance with only single-scale testing.

Extensive experiments demonstrate that, as a new region proposal generation approach, AF-RPN can achieve higher recall rate than the vanilla RPN [5] and FPN–RPN [17] on the large-scale COCO-Text [18] dataset and also outperforms the rotation region proposal network (RRPN) [12] and FPN–RRPN remarkably on the multi-oriented ICDAR-2015 dataset [3]. Owing to the high-quality text proposals, our Faster R-CNN-based two-stage text detection approach, i.e., AF-RPN + Fast R-CNN, achieves the state-of-the-art results on the ICDAR-2017 MLT [4], COCO-Text [18], ICDAR-2015 [3] and ICDAR-2013 [2] text detection benchmark tasks by only using single-scale and single-model (VGG16) testing.

The remainder of this paper is organized as follows: Previous related approaches are summarized in Sect. 2. The proposed text detection approach and the training strategy are described in detail in Sects. 3, 4 and 5, respectively. Section 6 presents our experimental results and discussions. Finally, the conclusion and future work are given in Sect. 7.

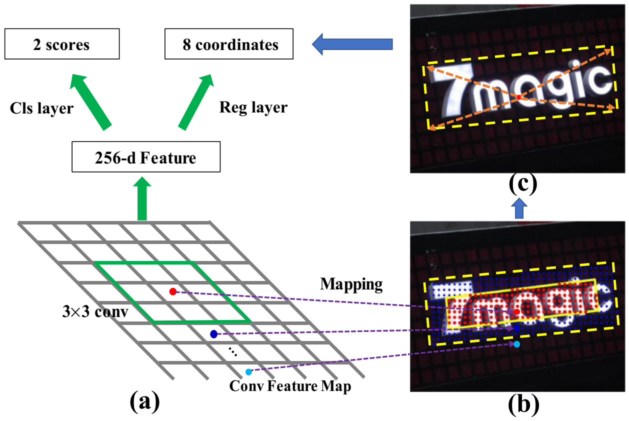

Fig. 1 **a** A detection module of AF-RPN, which can be considered as a sliding window detector like the vanilla RPN [5]; **b** mapping each pixel in the convolutional feature map to the corresponding sliding point in the raw image; examples of text (red), ignored (blue) and non-text (outside the text region) sliding points; **c** direct regression [9] from a given text sliding point to the four vertices of the concern ground-truth box

2 Related work

2.1 Scene text detection

Existing text detection methods can be roughly divided into two categories: bottom-up [8,19–28] and top-down methods [9–13,15,29–37].

Bottom-up methods. Bottom-up methods are generally composed of three major steps [38], i.e., candidate text connected component (CC) extraction (e.g., based on MSER [7] or SWT [8]), text/non-text classification and text line grouping. Bottom-up methods, especially MSER-based ones, were once the mainstream methods before the deep learning era and won the first places in both the ICDAR-2011 [1] and ICDAR-2013 [2] robust reading competitions. However, these methods have some notable limitations. For example, some text in natural scene images cannot be extracted by current candidate text CC extraction methods like MSER or SWT, which affects the recall rate severely [23]. Moreover, these methods usually generate a large number of non-text CCs, posing a big challenge to the succeeding text/non-text classification and text line grouping problems, which makes the corresponding solutions generally very complicated and less robust [35]. These methods have fallen behind CNN-based top-down approaches in terms of both accuracy and adaptability, especially when dealing with the more challenging "Incidental Scene Text" [3] and "Multilingual scene text detection" [4] scenarios.

Top-down methods. CNN-based top-down text detection approaches have become the mainstream recently. Based on the basic detection target, these methods can be further divided into three categories: pixel based, word/line based and segment based.

1. Pixel based. Pixel-based methods [31,32,39] borrow the idea of semantic segmentation and employ an FCN to make a pixel-level text/non-text prediction, which produces a text saliency map for text detection. However, only coarse text blocks can be detected from this saliency map [31], so complex post-processing steps are needed to extract accurate bounding boxes of text lines.

2. Word/line based. Word/line-based methods treat text as a specific object and leverage state-of-the-art object detection frameworks to detect words or text lines from images directly. Jaderberg et al. [33] adapted R-CNN [40] for text detection, while its performance was limited by the traditional region proposal generation methods. Gupta et al. [34] resembled the YOLO framework [41] and employed a fully convolutional regression network to perform text detection and bounding box regression at all locations and multiple scales of an image. Zhong et al. [10] and Liao et al. [11] employed the anchor-based Faster R-CNN [5] and SSD [6] frameworks to solve word-level horizontal text detection problem, respectively. In order to extend Faster R-CNN and SSD to multi-oriented text detection, Ma et al. [12] and Liu et al. [13] proposed quadrilateral anchors to hunt for inclined text proposals which could better fit the multi-oriented text instances. However, as mentioned above, these anchor-based methods are not effective and flexible enough for text detection, which lead to inferior performance. To overcome the inefficiency of anchor mechanism, Zhou et al. [15] and He et al. [9] borrowed the idea of DenseBox [14] and used a one-stage FCN to output pixel-wise textness scores as well as the quadrilateral bounding boxes through all locations and scales of an image. Although more flexible, they cannot handle long or large text instances effectively [15]. In this paper, to address the limitations of anchor mechanism and improve the capabilities of DenseBox-based approaches, we propose incorporating the "anchor-free" idea of DenseBox into the Faster R-CNN framework. Concretely, we propose a novel AF-RPN to replace the original anchor-based RPN so that our adapted Faster R-CNN-based text detector can be robust to not only multi-oriented text instances, but also long or large text instances. This is the main contribution of this paper.

3. Segment based. Instead of detecting whole words or text lines directly, segment-based methods use anchor-based or DenseBox-based object detection methods to detect text segments firstly, each of which contains a character (e.g., [37]) or part of a word/text line (e.g., [35,36]). Extracted text segments are then grouped into text lines with conventional heuristic text line grouping algorithms [35,37] or the learned linkage information [36,42]. Our proposed AF-RPN can be seamlessly leveraged by these methods.

2.2 Anchor mechanism in object detection

Anchor mechanism plays an important role in current state-of-the-art object detection and instance segmentation methods, e.g., Faster R-CNN [5], SSD [6], RetinaNet [43] and Mask R-CNN [44]. Formally, these anchor-based methods pre-define a set of anchors of different scales and shapes and use them as reference boxes to predict the corresponding region proposals or the target objects. Therefore, careful hand-crafted anchor design is critical to the performance of these anchor-based methods. However, for the domain-specific scene text detection task, since scene text instances have wider variability in scales, aspect ratios and especially orientations than general objects, it requires much more complicated anchor design, which makes these anchor-based methods sophisticated and inefficient. To overcome this pain point, we propose the AF-RPN, which can generate high-quality text proposals in an anchor-free manner by directly predicting the offsets from a given sliding point to the bounding box vertices of the concerned text instance.

3 Anchor-free region proposal network

Our proposed AF-RPN is composed of a backbone network and three scale-specific detection modules. The backbone network is responsible for computing a multi-scale convolutional feature pyramid over the full input image. Three detection modules are attached to different pyramid levels and designed to detect small, medium and large text instances, respectively. Each detection module contains a small network with two sibling output layers for text/non-text classification and quadrilateral bounding box regression, respectively. A schematic view of our AF-RPN architecture is depicted in Fig. 2, and details are described in the following subsections.

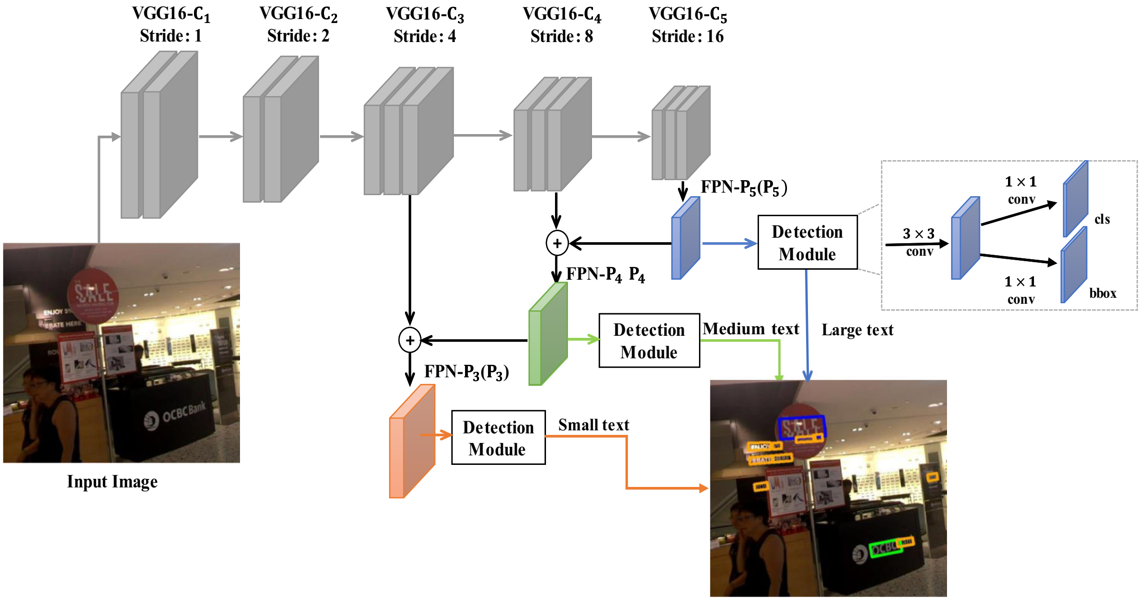

Fig. 2 Architecture of the proposed AF-RPN, which consists of an FPN backbone network [17] and three scale-specific detection modules (Fig. 1a) for small, medium and large text detection, respectively. Visualization of text proposals after score thresholding and non-maximum suppression (NMS)

3.1 Network architecture of AF-RPN

We adopt FPN [17] as the backbone network for AF-RPN. In brief, FPN enhances a standard convolutional network with a top-down pathway and lateral connections to construct a rich and multi-scale feature pyramid from a single-resolution input image. Each level of the pyramid can be effectively used for detecting objects of scales within a specific range. We build FPN on top of the conventional VGG16 architecture [45] instead of ResNets [46] for fairer comparison with other methods. Here, we construct a feature pyramid with three levels, i.e., P_3, P_4 and P_5, whose strides are 4, 8, 16, respectively. All feature pyramid levels have $C = 256$ channels. We refer readers to [17] for further implementation details.

Three scale-specific detection modules are attached to P_3, P_4 and P_5, respectively. Similar to RPN [5], each detection module can be considered as a sliding window detector, which uses a small network to perform text/non-text classification and quadrilateral bounding box regression in each 3×3 sliding window on a single-scale pyramid level (Fig. 1a). As depicted in the right part of Fig. 2, the small network is implemented as a 3×3 convolutional layer followed by two sibling 1×1 convolutional layers, which are used for predicting textness score and bounding box coordinates, respectively. We propose a scale-friendly learning method to learn three detection modules that are designed to detect small, medium and large text instances, respectively. This can effectively relieve the learning difficulties in the text/non-text classification and bounding box regression of each detection module, thus making AF-RPN be able to deal with large text scale variance more robustly. The details of scale division are described in Sect. 3.2, and the ground-truth definition of AF-RPN is elaborated in Sect. 3.3.

3.2 Scale-friendly learning

In the training stage, we assign text instances to the three detection modules of AF-RPN based on the spatial sizes of their features on the corresponding pyramid levels. We design a series of controlled experiments on the large-scale COCO-Text dataset and observe that, when the scales (i.e., shorter sides) of the features for text instances on a pyramid level are less than 3 pixels (px), the performance of the corresponding detection module degrades dramatically. Experiments are described as follows: First, we train a detection module attached to P_4 (DM-P_4) to specially detect text instances whose scales are less than 48 px in the resized images. To achieve this, only ground-truth bounding boxes on the training set of COCO-Text within this scale range are selected for training, and others are ignored. Then we select the top-300 scoring detection results to compute the recall rate at an IoU threshold of 0.5 on the validation set of COCO-Text. The results are listed in the first part of Table 1. It can be seen that DM-P_4 achieves a high recall rate of 93.05% and 97.37% in the text scale range of [24 px, 36 px] and [36 px, 48 px], while the recall rate in the text scale range of [1 px, 12 px] and [12 px, 24 px] is degraded to 30.34% and 76.00%, respectively. It should be noted that there are lots of small text instances whose scales are less than 24 px in the training set. So this degraded performance must be caused by their insuf-

Table 1 Recall rate in each text scale range on COCO-Text for two scale-specific detection modules (DM-P_4 and DM-P_3) of AF-RPN

Detection module	Text scale range (px)			
	[1, 12] (%)	[12, 24] (%)	[24, 36] (%)	[36, 48] (%)
DM-P_4	30.34	76.00	93.05	97.37
DM-P_3	47.75	91.36	–	–

Testing images are resized such that their short sides have 800 px

ficient features on P_4, whose scales have less than 3 px. To further confirm this assumption, we train another detection module attached to P_3 (DM-P_3) to detect small text instances whose scales are less than 24 px with the similar training strategy. As shown in the second part of Table 1, DM-P_3 significantly improves the recall rate in the text scale range of [12 px, 24 px] from 76.00 to 91.36%. However, DM-P_3 still struggles with extreme text instances whose scales are less than 12 px because the scales of their features on P_3 are still less than 3 px.

Therefore, based on these observations, when assigning a text instance to a pyramid level, we ensure that the scale of its features on this pyramid level has no less than 3 px. As the strides of P_3, P_4 and P_5 are 4, 8, 16 px, the scales of text instances assigned to them should have no less than 12, 24 and 48 px, respectively. Consequently, we classify text instances into three groups according to their scales (shorter side lengths), i.e., small text ($<$ 24 px), medium text (24 px-48 px) and large text ($>$ 48 px).

3.3 Label generation

Text instances in text detection tasks are usually labeled in word level with quadrilateral or axis-aligned bounding boxes. To ease implementation, for quadrilateral bounding boxes, we use their minimum enclosing boxes (oriented rectangles) as new ground-truth bounding boxes (dashed lines in Fig. 1b). It is inevitable that some surrounding backgrounds can be included in the ground-truth bounding boxes when they are not tight enough. To reduce the influence of background noise on text/non-text classification, following [9,15], we shrink the short and long sides of each ground-truth rectangle by the scaling factors of 0.5 and 0.8, respectively, to create the corresponding core text region (solid lines in Fig. 1a), and only sliding points within core regions are taken as positive. Sliding points outside core regions but inside ground-truth rectangles are assigned a “DON’T CARE” label and are ignored during training (Fig. 1b). Sliding points outside all ground-truth rectangles are taken as negative. For each positive sliding point, we predict the coordinates of its bounding box directly. Let $p_t = (x_t, y_t)$ denote a positive sliding point, which is located in a ground-truth rectangle G. Let $\{p_i = (x_i, y_i)|i \in \{1, 2, 3, 4\}\}$ denote the vertices of G. Then the coordinate offsets from p_t tp G’s vertices can be denoted as $\{\triangle_i = (\triangle_{x_i}, \triangle_{y_i})|i \in \{1, 2, 3, 4\}\}$, where $\triangle_{x_i} = (x_i - x_t)$ and $\triangle_{y_i} = (y_i - y_t)$ (Fig. 1c). Considering the fact that the numerical ranges of $\triangle_{x_i}$ and $\triangle_{y_i}$ could be very large, we normalize them as follows: $\triangle_{x_i} = (x_i - x_t)/norm$, $\triangle_{y_i} = (y_i - y_t)/norm$, where $norm$ represents the normalization term. If p_t is on P_3 or P_4, $norm$ is set as the upper bound of the corresponding scale range, i.e., $norm = 24$ or $norm = 48$. If p_t is on P_5, $norm$ is set as a proportion of the RF of P_5 (related to 3 × 3 units), i.e., $norm = \alpha RF_{P_5}$, where $\alpha = 0.5$.

4 Faster R-CNN with AF-RPN

Given an input image, we first use AF-RPN to perform the first-stage text detection, after which the top-N_1 scoring detection results of each detection module of AF-RPN are selected to construct a proposal set $\{P\}$. Then, we use the standard non-maximum suppression (NMS) algorithm with an IoU threshold of 0.7 to remove redundant proposals in $\{P\}$ and select the top-N_2 scoring proposals. Both N_1 and N_2 are set to 2000 in the training stage, and 300 in the testing stage. Next, we adopt the same scale division criterion illustrated in Sect. 3.2 to classify text proposals into small, medium and large text proposal groups, which are assigned to pyramid levels P_3, P_4 and P_5, respectively.

In the second stage, similar to AF-RPN, three individual Fast R-CNN detectors, which do not share parameters, are attached to pyramid levels P_3, P_4 and P_5, respectively. For each proposal, we adopt RoI Align algorithm [44] to extract 7 × 7 features from its assigned pyramid level and attach two 2048-d fully connected (*fc*) layers (each followed by ReLU) before the final text/non-text classification and bounding box regression layers. A schematic view of the Fast R-CNN module is depicted in Fig. 3.

5 Training

5.1 Loss function

Multi-task loss for AF-RPN. There are two sibling output layers for each scale-specific detection module, i.e., a text/non-text classification layer and a quadrilateral bounding box regression layer. The multi-task loss function for each detection module is denoted as follows:

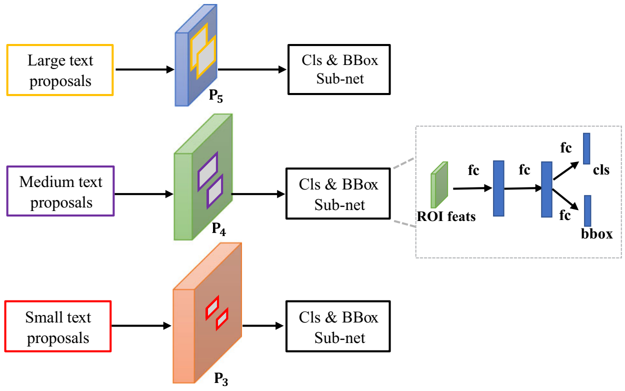

Fig. 3 A schematic view of the Fast R-CNN module based on an FPN backbone network [17]. Text proposals generated by AF-RPN are classified into small, medium and large text proposal groups according to the scale division criterion illustrated in Sect. 3.2. Three individual Fast R-CNN detectors are attached to P_3, P_4 and P_5 and designed to deal with small, medium and large text proposals, respectively. The design of each Fast R-CNN detector's head is very simple, which just includes two 2048-d fc layers before the final predictions

$$L(c, c^*, t, t^*) = \lambda_{\text{cls}} L_{\text{cls}}(c, c^*) + \lambda_{\text{loc}} L_{\text{loc}}(t, t^*), \tag{1}$$

where c and c^* are predicted and ground-truth labels for each sliding point, respectively, $L_{\text{cls}}(c, c^*)$ is a softmax loss for classification tasks, t and t^* represent the predicted and ground-truth 8-dimensional normalized coordinate offsets from p_t to G, $L_{\text{loc}}(t, t^*)$ is a smooth-L_1 loss [47] for regression tasks, λ_{cls} and λ_{loc} are two loss-balancing parameters, and we set $\lambda_{\text{cls}} = 1$ and $\lambda_{\text{loc}} = 3$.

The total loss of AF-RPN is the sum of the losses of three scale-specific detection modules.

Multi-task loss for Fast R-CNN. The loss function for each individual Fast R-CNN detector is the same as Eq. (1). Only the parameters λ_{cls} and λ_{loc} are set differently. Here, we set $\lambda_{\text{cls}} = 1$ and $\lambda_{\text{loc}} = 1$. Moreover, compared with AF-RPN, there are some differences in coordinate offsets normalization for the quadrilateral bounding box regression task. Let P be an input proposal and (P_x, P_y, P_w, P_h) be the center coordinates, height and width of its axis-aligned bounding box, respectively. We directly use P_w and P_h to normalize the coordinate offsets from (P_x, P_y) to G's vertices as follows: $\triangle_{x_i} = (x_i - P_x)/P_w$, $\triangle_{y_i} = (y_i - P_y)/P_h$, where $i \in \{1, 2, 3, 4\}$. The total loss of Fast R-CNN is the sum of losses of three individual Fast R-CNN detectors.

5.2 Training details

In each training iteration of AF-RPN, we sample a mini-batch of 128 positive and 128 negative sliding points for each detection module. Similarly, for Fast R-CNN, we sample a mini-batch of 64 positive and 64 negative text proposals for each individual Fast R-CNN detector. A proposal is assigned a positive label if it has an IoU over 0.5 with any ground-truth bounding box, and a negative label if its IoU overlap is less than 0.3 for all ground-truth bounding boxes. For efficiency, the IoU overlaps between proposals and ground-truth boxes are calculated using their axis-aligned rectangular bounding boxes. Note that each ground-truth bounding box is assigned to only one detection module of AF-RPN or one Fast R-CNN detector according to the text scale division criterion illustrated in Sect. 3.2 and ignored by other two in the training stage.

6 Experiments

6.1 Datasets and evaluation protocols

To evaluate the performance of the proposed approach, we conduct experiments on four standard text detection benchmark tasks, including ICDAR-2017 MLT [4], COCO-Text [18], ICDAR-2015 [3] and ICDAR-2013 [2]. Text instances are labeled in word level with quadrilateral bounding boxes in the former two datasets and axis-aligned rectangular bounding boxes in the ICDAR-2013 dataset. Two kinds of bounding boxes are labeled in COCO-Text. ICDAR-2017 MLT is built for the Multilingual scene text detection and script identification challenge in the ICDAR-2017 Robust Reading Competition, which includes 9 languages: Chinese, Japanese, Korean, English, French, Arabic, Italian, German and Indian. It contains 7200, 1800 and 9000 images for training, validation and testing, respectively. COCO-Text is a large-scale dataset with 43,686 training, 10,000 validation and 10,000 testing images, which is another text detection challenge in the ICDAR-2017 Robust Reading Competition [48]. ICDAR-2015 is built for the Incidental Scene Text challenge in the ICDAR-2015 Robust Reading Competition, which contains 1000 and 500 images for training and testing. ICDAR-2013 is a horizontal text detection dataset, with 229 images for training and 233 for testing.

The standard performance metrics for text detection are precision, recall and F-measure rates. To make our results comparable to others, we use the online official evaluation tools to evaluate the performance of our approach on the ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013 testing sets. We use recall rate as an evaluation metric to compare the performance of different region proposal generation approaches on the COCO-Text validation set and ICDAR-2015 testing set.

6.2 Implementation details

The weights of VGG16 related layers in the FPN backbone network are initialized by using a pre-trained VGG16 model for the ImageNet classification task [45]. The weights of the new layers for FPN, AF-RPN and Fast R-CNN are initialized by using random weights with a Gaussian distribution of mean 0 and standard deviation 0.01. The training process can be separated into two stages: In the first stage, we train the AF-RPN model until convergence. Then in the second stage, we use this well-trained AF-RPN model to initialize the Faster R-CNN model, which is then jointly fine-tuned in an end-to-end manner. All the models are optimized by the standard SGD algorithm with a momentum of 0.9 and weight decay of 0.0005.

The number of training iterations and adjustment strategy of learning rate depend on the size of different datasets. Specifically, for ICDAR-2017 MLT, we use the training and validation data, i.e., a total of 9000 images for training. Both AF-RPN and Faster R-CNN models are trained for 400 K iterations with an initial learning rate of 0.001, which is then divided by 10 at 180 K and 360 K iterations. As the training sets of ICDAR-2015 and ICDAR-2013 are too small, some previous methods usually use some larger datasets like VGG SynthText [34] to pre-train their models. To make our experimental results reproducible, we follow [49] to use the model trained on ICDAR-2017 MLT as our pre-trained model, which is then fine-tuned on the training set of ICDAR-2015 and ICDAR-2013, respectively. All models for these two datasets are trained for 50 K iterations with an initial learning rate of 0.0005, which is divided by 5 at 20 K and 40 K iterations. For COCO-Text, we train models on its training set and report region proposal generation results and text detection results on its validation set and testing set, respectively. All models are trained for 500 K iterations with an initial learning rate of 0.001, which is then divided by 10 at 200 K and 400 K iterations.

We implement our approach based on Detectron [50], and experiments are conducted on a workstation with 4 Nvidia P100 GPUs. We adopt a multi-scale training strategy. The scale S is defined as the length of the shorter side of an image. In each training iteration, a selected training image is individually rescaled by randomly sampling a scale S from the set $\{448, 608, 768, 928, 1088\}$.

6.3 Region proposal quality evaluation

6.3.1 Comparison with prior arts

We compare our proposed AF-RPN to RPN and FPN–RPN for the rectangular proposal generation task on COCO-Text firstly and then compare AF-RPN to RRPN and FPN–RRPN for the quadrilateral proposal generation task on ICDAR-2015. We evaluate the recall rates $R_{\#}^{0.5}$ and $R_{\#}^{0.75}$ at a single IoU threshold of 0.5 and 0.75, respectively, while we also evaluate the average recall rate $AR_{\#}^{0.5:0.05:0.95}$ at multiple IoU thresholds between 0.50 and 0.95 with an interval of 0.05, when using a given fixed number (#) of text proposals. We report results for 50, 100 and 300 proposals per image (300 proposals are used for Fast R-CNN in testing stage). The scale S for all testing images is set as 800 for all experiments in this section.

Rectangular proposal. For fair comparison, we design a complicated set of anchors for RPN and FPN–RPN following [10]. Specifically, for RPN, we use 3 scales $\{32, 64, 128\}$ and 6 aspect ratios $\{0.2, 0.5, 0.8, 1.0, 1.2, 1.5\}$, i.e., 18 anchors, at each sliding position on C_5. For FPN–RPN, we use 6 aspect ratios and a single scale in $\{32, 64, 128\}$, i.e., 6 anchors, at each position on each pyramid level in $\{P_3, P_4, P_5\}$. In the training stage of RPN and FPN–RPN, an anchor is assigned a positive label if it has the highest IoU for a given ground-truth box or an IoU over 0.5 with any ground-truth box, and a negative label if it has an IoU less than 0.1 for all ground-truth boxes as in [10]. The training strategies are kept the same as AF-RPN.

The results are listed in Table 2a–c. It can be seen that our proposed AF-RPN outperforms RPN and FPN–RPN substantially in all evaluation metrics on COCO-Text, which demonstrates the effectiveness of our proposed AF-RPN. When the evaluated number of proposals drops from 300 to 50, the improvements are much more significant.

Quadrilateral proposal. In order to hunt for inclined text proposals, following [12], we design a set of rotated anchors to achieve RRPN and FPN–RRPN. For RRPN, we use the above 3 scales, 6 aspect ratios and the additional 6 orientations $\{-\pi/6, 0, \pi/6, \pi/3, \pi/2, 2\pi/3\}$, i.e., 108 rotated anchors, at each sliding position on C_5. Similarly, for FPN–RRPN, we apply 6 aspect ratios, 6 orientations and a single scale, i.e., 36 anchors, at each position on each pyramid level. During training, we assign a positive label to a rotated anchor if it has the highest Skewed IoU [12] for a given ground-truth box or a Skewed IoU over 0.5 with any ground-truth box, while the intersection orientation with the matched ground-truth is less than $\pi/12$. A rotated anchor is assigned a negative label if it has a Skewed IoU less than 0.1 for all ground-truth boxes.

As shown in Table 2g–i, AF-RPN achieves 86.5%, 90.4% and 94.1% in $R_{50}^{0.5}$, $R_{100}^{0.75}$ and $AR_{300}^{0.5:0.05:0.95}$ on ICDAR-2015, respectively, outperforming RRPN and FPN–RRPN by a large margin.

6.3.2 Ablation study on using different pyramid levels

We conduct a series of ablation experiments by using different pyramid levels to evaluate rectangular proposal quality on COCO-Text. As shown in Table 2c–f, the performance of the AF-RPN model that uses all pyramid levels (P_3, P_4 and P_5) is much better than that uses only one (P_3) or two (P_3 and P_4 or P_4 and P_5) pyramid levels. Furthermore, we also calculate the recall rate $R_{300}^{0.5}$ of the AF-RPN model that

Table 2 Region proposal quality evaluation on COCO-Text and ICDAR-2015 (%)

Method	Feature	#Anchor / #sp (k)	$R^{0.5}_{50}$	$R^{0.75}_{50}$	$AR^{0.5:0.05:0.95}_{50}$	$R^{0.5}_{100}$	$R^{0.75}_{100}$	$AR^{0.5:0.05:0.95}_{100}$	$R^{0.5}_{300}$	$R^{0.75}_{300}$	$AR^{0.5:0.05:0.95}_{300}$
Rectangular proposal quality evaluation on COCO-text											
(a) RPN	C_5	45	70.5	27.3	33.6	80.1	32.8	39.2	88.3	38.5	44.5
(b) FPN–RPN	$\{P_k\}$	315	69.5	30.3	34.9	79.6	38.8	41.8	90.0	50.2	50.0
(c) AF-RPN	$\{P_k\}$	52.5	**78.0**	**37.5**	**40.7**	**86.1**	**43.8**	**46.2**	**92.3**	**51.3**	**51.5**
(d) AF-RPN	P_3	40	76.0	33.8	38.4	85.3	39.9	44.2	89.1	44.7	47.5
(e) AF-RPN	$\{P_3, P_4\}$	50	77.1	35.0	39.2	85.3	40.9	44.5	91.3	46.9	49.0
(f) AF-RPN	$\{P_4, P_5\}$	12.5	76.3	33.6	38.3	82.9	37.8	42.3	86.9	41.3	45.2
Quadrilateral proposal quality evaluation on ICDAR-2015											
(g) RRPN	C_5	270	76.8	27.5	36.2	82.2	31.5	39.8	88.8	35.5	43.5
(h) FPN–RRPN	$\{P_k\}$	1890	81.9	42.7	44.0	86.7	47.7	47.6	90.3	52.5	50.7
(i) AF-RPN	$\{P_k\}$	52.5	**86.5**	**48.3**	**47.8**	**90.4**	**52.1**	**50.6**	**94.1**	**55.1**	**53.1**

The column “feature” denotes the feature maps on which the prediction layers are attached. The column “#anchor / #sp” represents the number of anchors or sliding points used during inference for anchor-based region proposal networks and our proposed AF-RPN, respectively

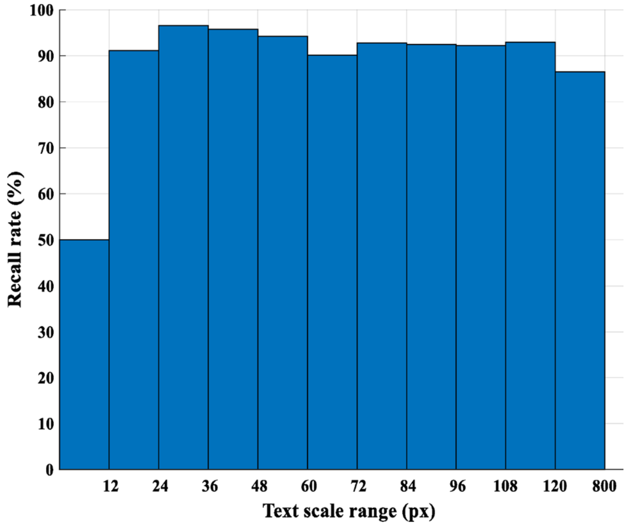

Fig. 4 Recall rate ($R^{0.5}_{300}$) of the AF-RPN model that uses all pyramid levels in each text scale range on COCO-Text

uses all pyramid levels in each text scale range. As shown in Fig. 4, the proposed AF-RPN can deal with large text scale variance robustly, while it cannot perform equally well for those extremely small text instances (< 12 px) because of their low-resolution features (< 3 px) on P_3 as analyzed in Sect. 3.2.

6.4 Text detection performance evaluation

In this section, we evaluate our proposed Faster R-CNN-based text detection approach on ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013. We use the top-300 scoring text proposals generated by AF-RPN for the succeeding Fast R-CNN. Detection results from different Fast R-CNN detectors are aggregated with Skewed NMS [12]. All the experiments are based on single-model and single-scale testing. The scale of testing image S is set as 1440, 800, 800 and 512 for ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013, respectively. We compare the performance of our approach with other most competitive results on these four benchmark tasks. For fair comparisons, we report all results without using recognition information.

Table 3 Comparison with prior arts on ICDAR-2017 MLT (%)

Method	Recall	Precision	F-measure
linkage-ER-Flow† [4]	25.59	44.48	32.49
TH-DL† [4]	34.78	67.75	45.97
SARI_FDU_RRPN_v2† [4]	55.37	67.07	60.66
SARI_FDU_RRPN_v1† [4]	55.50	71.17	62.37
Sensetime OCR† [4]	69.43	56.93	62.56
SCUT_DLVClab1† [4]	54.54	80.28	64.96
He et al. [9] + MS	57.94	76.69	66.01
Liu et al. [49]	57.45	79.48	66.69
Lyu et al. [51]	55.60	**83.80**	66.80
Lyu et al. [51] + MS	**70.60**	74.30	72.40
Proposed	66.67	79.49	**72.52**

†Indicates the ICDAR-2017 MLT competition results. MS stands for using multi-scale testing

ICDAR-2017 MLT. We collect competition results as well as recent results reported in the published literatures on this benchmark for comprehensive comparisons. As shown in Table 3, our approach outperforms the top-1 competition result remarkably by improving F-measure from 64.96 to 72.52%. Furthermore, our approach improves the most competing method [51] by 5.72% in F-measure when the single-scale testing is used in both methods. Although [51] has applied a multi-scale testing strategy for extremely push-

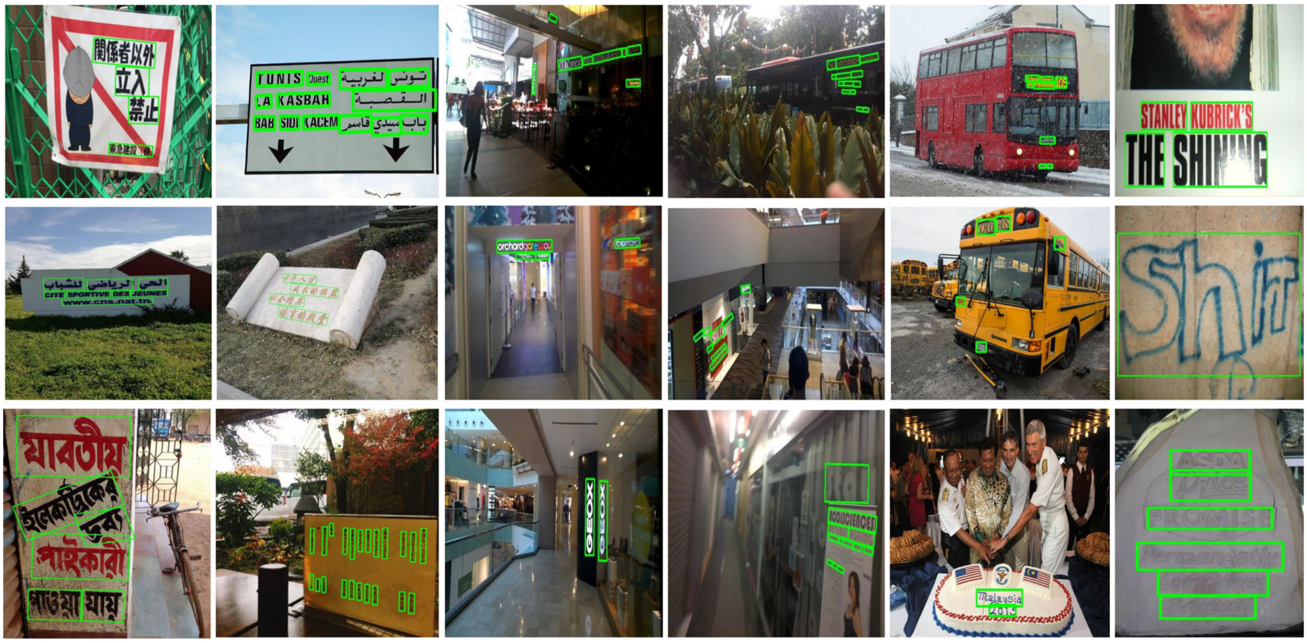

Fig. 5 Detection results of our approach: first–second col: ICDAR-2017 MLT; third–fourth col: ICDAR-2015; fifth col: COCO-Text; sixth col: ICDAR-2013

ing performance from 66.80 to 72.40%, their result is still inferior to our single-scale testing one. Considering that ICDAR-2017 MLT is a large-scale, extremely challenging, and the first Multilingual text detection dataset, the superior performance achieved by our proposed approach can demonstrate the advantage of our approach.

COCO-Text. As shown in Table 4, our approach outperforms the closest method [51] substantially by improving F-measure from 62.60 to 65.07% when the IoU threshold is set as 0.5. Furthermore, when the evaluation criterion becomes stricter, i.e., IoU threshold is set as 0.75, our approach still achieves the best result of 36.45% in F-measure.

ICDAR-2015. On the challenging ICDAR-2015 task, as shown in Table 5, our approach achieves the best result of 82.96%, 90.02% and 86.34% in recall, precision and F-measure, respectively, outperforming other recently published CNN-based approaches substantially.

ICDAR-2013. We also evaluate our approach on ICDAR-2013, which is a popular dataset for horizontal text detection. As shown in Table 6, although ICDAR-2013 has been well tuned by many previous methods, our approach still achieves the best result of 91.92% in F-measure.

Inference time. Based on our current implementation, our text detector has an inference time of 0.40 s, 0.18 s, 0.20 s and 0.16 s per image when using a single P100 GPU for $S = 1440$, $S = 800$, $S = 800$ and $S = 512$ on ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013, respectively.

Table 4 Comparison with prior arts on COCO-Text (%)

Method	Recall	Precision	F-measure
IoU = 0.5 criterion			
HappyCCL [48]	64.82	44.88	53.04
UM [48]	**65.47**	47.58	55.11
Lyu et al. [51]	52.90	72.50	61.10
Liao et al. [52] + MS	57.00	64.00	61.00
Text_Detection_DL [48]	61.81	60.90	61.35
Lyu et al. [51] + MS	62.20	62.90	62.60
Proposed	58.34	**73.56**	**65.07**
IoU = 0.75 criterion			
HappyCCL [48]	27.89	19.31	22.82
Text_Detection_DL [48]	25.54	25.16	25.35
UM [48]	31.21	22.68	26.27
Lyu et al. [51]	30.00	40.00	34.60
Lyu et al. [51] + MS	**34.80**	35.10	34.90
Liao et al. [52] + MS	34.00	38.00	36.00
Proposed	32.68	**41.21**	**36.45**

R, P and F stand for recall, precision and F-measure, respectively. MS means using multi-scale testing

Qualitative results. The superior performance achieved on the above four datasets demonstrates the effectiveness and robustness of our Faster R-CNN-based text detection approach. As shown in Fig. 5, our text detector can detect scene text regions under various challenging conditions, such as multiple languages, low resolution, non-uniform illumination, large aspect ratios as well as varying orientations.

Table 5 Comparison with prior arts on ICDAR-2015 (%)

Method	Recall	Precision	F-measure
1st ICDAR' 2015 [3]	36.74	77.46	49.84
Liu et al. [13]	68.22	73.23	70.64
Shi et al. [36]	76.80	73.10	75.00
Ma et al. [12]	73.23	82.17	77.44
Han et al. [37] + MS	77.03	79.33	78.16
Zhou et al. [15] (VGG16)	72.75	80.46	76.41
Zhou et al. [15] (PVANET2x + MS)	78.33	83.27	80.72
He et al. [9] + MS	80.00	82.00	81.00
Deng et al. [42]	82.00	85.50	83.70
Lyu et al. [51] + MS	79.70	89.50	84.30
Liu et al. [49]	82.04	88.84	85.31
Proposed	**82.96**	**90.02**	**86.34**

MS stands for using multi-scale testing

6.4.1 Component evaluation

In this section, we conduct a series of ablation experiments to evaluate the effectiveness of each component of our approach on ICDAR-2017 MLT and ICDAR-2015. All models are trained with the same hyper-parameters for fair comparison, and all the results are based on single-scale and single-model testing.

Effectiveness of multi-scale predictions with scale-friendly learning. Other than being used as a region proposal generator, our proposed AF-RPN itself can be used as a one-stage text detector. Here, we evaluate the text detection performance of AF-RPN as well as its variants. We first train an AF-RPN model that uses only one pyramid level (P_3), which can be considered as a re-implementation of EAST [15] with a VGG16-FPN backbone network. As shown in the first part of Table 7, this model obtains an F-measure of 54.48% and 81.27% on ICDAR-2017 MLT and ICDAR-2015, respectively. Note that in the original implementation of EAST, the shrinking ratios for the short and long sides of each ground-truth bounding box are both set as 0.7. We follow this setting and retrain the above AF-RPN model. As shown in the second part of Table 7, the text detection results are comparable on these two datasets, which demonstrates that our AF-RPN model is insensitive to these two shrinking ratios to some extent. Furthermore, we train another AF-RPN model that uses all pyramid levels but does not apply the proposed scale-friendly learning strategy. We can observe that,

Table 6 Comparison with prior arts on ICDAR-2013 (%)

Method	Recall	Precision	F-measure
1st ICDAR' 2013 [2]	69.28	88.80	77.83
Gupta et al. [34]	75.50	92.00	83.00
Zhong et al. [10]	83.00	87.00	85.00
Liao et al. [11] + MS	83.00	89.00	85.89
Zhou et al. [15]	82.67	92.64	87.37
He et al. [9] + MS	86.16	89.26	87.68
Lyu et al. [51] + MS	84.40	92.20	88.00
Deng et al. [42] + MS	87.50	88.60	88.10
Han et al. [37] + MS	87.53	93.34	90.34
Proposed	**90.06**	**93.84**	**91.92**

MS stands for using multi-scale testing

Table 7 Component evaluation on ICDAR-2017 MLT and ICDAR-2015 (%)

Feature	Shrinking ratio	Scale-friendly learning?	Fast R-CNN?	ICDAR-2017 MLT			ICDAR-2015		
				R	P	F	R	P	F
P_3	0.8 × 0.5	–	×	51.41	57.94	54.48	78.96	84.15	81.47
			✓	62.72	78.91	69.89	82.14	89.65	85.73
P_3	0.7 × 0.7	–	×	51.77	57.36	54.42	79.59	83.02	81.27
			✓	62.72	78.76	69.83	82.43	88.94	85.56
$\{P_k\}$	0.8 × 0.5	×	×	52.53	57.13	54.74	79.73	83.55	81.59
			✓	63.60	78.26	70.17	82.43	89.35	85.75
$\{P_k\}$	0.8 × 0.5	✓	×	59.58	64.97	62.16	79.30	88.55	83.67
			✓	**66.67**	**79.49**	**72.52**	**82.96**	**90.02**	**86.34**

R, P and F stand for recall, precision and F-measure, respectively

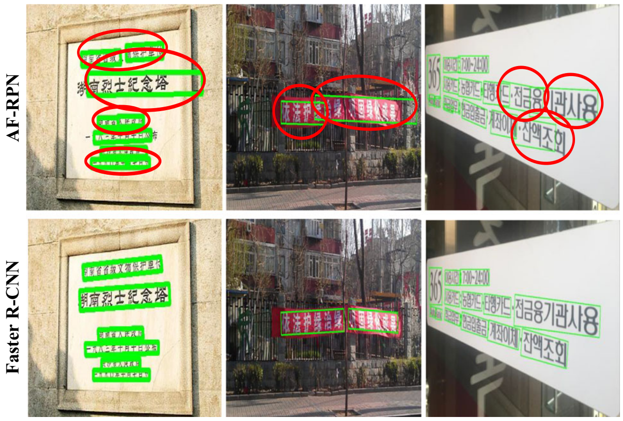

Fig. 6 Qualitative detection results of AF-RPN and our Faster R-CNN-based text detector. Red ellipses represent wrongly detected text caused by unsatisfactory localization accuracy of AF-RPN. It can be seen that our Faster R-CNN-based text detector can effectively improve text localization accuracy (best viewed in color)

although using multi-scale predictions on all pyramid levels, the results of this AF-RPN model on ICDAR-2017 MLT and ICDAR-2015 are just marginally better than that of AF-RPN only attached on P_3 (the third part of Table 7). This indicates that directly extending single-scale prediction to multi-scale predictions cannot achieve satisfactory enough performance. To address this issue, we propose a scale-friendly learning strategy to let each detection module handle text instances of scales within an appropriate range. In this way, the learning difficulty of textness score prediction and bounding box regression problems for each detection module can be effectively relieved. Therefore, the whole AF-RPN model can deal with large text scale variance more robustly, leading to a 2.20% F-measure improvement on ICDAR-2015 and more gain of 7.42% on ICDAR-2017 MLT (the fourth part of Table 7).

Effectiveness of Fast R-CNN. One of the main contributions of our paper is that we propose to use AF-RPN as a new region proposal network for the Faster R-CNN framework. Here, we compare the text detection performance of AF-RPN (its variants) with the Faster R-CNN-based two-stage text detector to figure out the influence of the second-stage Fast R-CNN. As shown in Table 7, no matter using single-scale or multi-scale predictions, the Faster R-CNN-based two-stage text detector outperforms the corresponding AF-RPN-based one-stage text detector remarkably by improving F-measure by more than 2.67% on ICDAR-2015 and more than 10.36% on ICDAR-2017 MLT, respectively. Note that the gains on ICDAR-2017 MLT are more significant because there are many long text lines in Chinese, Japanese and Korean, whose localization accuracy can be effectively improved by Fast R-CNN. The large improvements in both precision and recall rates demonstrate the effectiveness of the second-stage Fast R-CNN detector. Some qualitative comparison results are presented in Fig. 6.

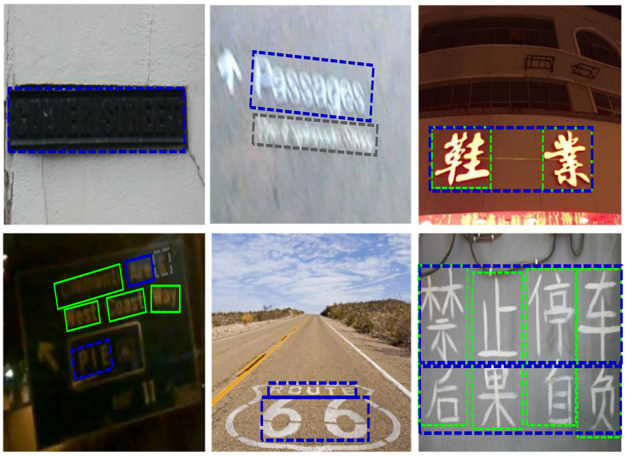

Fig. 7 Some failure examples. Blue dashed boxes: missed ground-truth bounding boxes. Gray dashed boxes: ignored ground-truth bounding boxes. Green solid boxes are correctly detected bounding boxes, while green dashed ones are wrongly detected bounding boxes (best viewed in color)

6.5 Discussion

6.5.1 Comparisons with relevant methods

In this section, we compare our approach with other most relevant scene text detection methods for better understanding the superiority of our approach.

Comparisons with anchor-based Faster R-CNN methods. Zhong et al. [10] employed Faster R-CNN with complicated and text-specific anchor designs to perform horizontal text detection. To extend Faster R-CNN to multi-oriented text detection, Ma et al. [12] introduced a rotated anchor strategy. However, owing to the inefficiency of anchor mechanism for text detection, the performances of these anchor-based methods are obviously inferior to our anchor-free approach, i.e., 85.00% [10] versus 91.92% in F-measure on ICDAR-2013, and 77.44% [12] versus 86.34% in F-measure on ICDAR-2015, respectively.

Comparisons with DenseBox-based methods. Although DenseBox-based text detection methods [9,15] also make use of the "anchor-free" concept, their capability is limited for large or long text, which could be a common issue for one-stage text detectors. Our Faster R-CNN-based two-stage approach, i.e., AF-RPN + Fast R-CNN, can overcome this limitation effectively as demonstrated in Sect. 6.4.1. Therefore, compared with [9], our approach improves the F-measure by 5.34% on ICDAR-2015 and achieves more gain of 6.51% on ICDAR-2017 MLT. More comprehensive comparisons with EAST [15] can be seen in the first part of Sect. 6.4.1.

6.5.2 Limitations of our approach

Although our proposed approach shows superior capability in most scenarios as demonstrated by the above experimental results, it still has some limitations. First, our approach cannot work equally well in certain cases, such as very low contrast, serious blur and complex layouts like large character spacing and ambiguous alignment. Some failure cases are depicted in Fig. 7. Second, our approach still struggles with extremely small text instances whose shorter sides are less than 12 px in resized images. A possible solution is to introduce the pyramid level P_2 with a stride of 2 for detecting these extremely small text. But this would lead to high computation. So, more researches are needed to address this challenging problem. Moreover, our approach cannot robustly deal with curved text instances. But the proposed AF-RPN can be seamlessly incorporated into the recent Mask R-CNN framework [44] that can effectively handle curved text instances from a perspective of instance segmentation, which would be our future work.

7 Conclusion and future work

In this paper, we present AF-RPN as an anchor-free and scale-friendly region proposal network for the Faster R-CNN framework. Comprehensive comparisons with RPN, FPN–RPN, RRPN and FPN–RRPN on COCO-Text and ICDAR-2015 datasets demonstrate the superior performance of our proposed AF-RPN used as a new region proposal network. Owing to the high-quality text proposals, our Faster R-CNN-based text detector, i.e., AF-RPN + Fast R-CNN, achieves state-of-the-art results on the ICDAR-2017 MLT, COCO-Text, ICDAR-2015 and ICDAR-2013 text detection benchmark tasks by only using single-scale and single-model testing. Future direction is to explore the effectiveness of the proposed AF-RPN in other detection tasks, such as generic object detection, face detection and pedestrian detection.

References

1. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: reading text in scene images. In: ICDAR, pp. 1491–1496 (2011)
2. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Gomez, L., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., de las Heras, L.P.: ICDAR 2013 robust reading competition. In: ICDAR, pp. 1484–1493 (2013)
3. Karatzas, D., Gomez, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S.-J., Shafait, F., Uchida, S., Valveny, E.: ICDAR 2015 robust reading competition. In: ICDAR, pp. 1156–1160 (2015)
4. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z.-B., Pal, U., Rigaud, C., Chazalon, J., Khlif, W., Luqman, M.M., Burie, J.C., Liu, C.-L., Ogier, J.M.: ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification—RRC-MLT. In: ICDAR, pp. 1454–1459 (2017)
5. Ren, S.-Q., He, K.-M., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. PAMI **39**(6), 1137–1149 (2017)
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single shot multiBox detector. In: ECCV (2016)
7. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 384–393 (2002)
8. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR, pp. 2963–2970 (2010)
9. He, W.-H., Zhang, X.-Y., Yin, F., Liu, C.-L.: Deep direct regression for multi-oriented scene text detection. In: ICCV, pp. 745–753 (2017)
10. Zhong, Z.-Y., Jin, L.-W., Huang, S.-P.: DeepText: a new approach for proposal generation and text detection in natural images. In: ICASSP, pp. 1208–1212 (2017)
11. Liao, M.-H., Shi, B.-G., Bai, X., Wang, X.-G., Liu, W.-Y.: TextBoxes: a fast text detector with a single deep neural network. In: AAAI, pp. 4164–4167 (2016)
12. Ma, J.-Q., Shao, W.-Y., Ye, H., Wang, L., Wang, H., Zheng, Y.-B., Xue, X.-Y.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans. Multimed. **20**(11), 3111–3122 (2018)
13. Liu, Y.-L., Jin, L.-W.: Deep matching prior network toward tighter multi-oriented text detection. In: CVPR, pp. 1962–1969 (2017)
14. Huang, L.-C., Yang, Y., Deng, T.-F., Yu, Y.-N.: Densebox: unifying landmark localization with end to end object detection. Preprint (2015). arXiv:1509.04874
15. Zhou, X.-Y., Yao, C., Wen, H., Wang, Y.-Z., Zhou, S.-C., He, W.-R., Liang, J.-J.: EAST: An efficient and accurate scene text detector. In: CVPR, pp. 5551–5560 (2017)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. PAMI **39**(4), 640–651 (2017)
17. Lin, T.-Y., Dollár, P., Girshick, R.B., He, K.-M., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR, pp. 2117–2125 (2017)
18. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: COCO-Text: dataset and benchmark for text detection and recognition in natural images. Preprint (2016). arXiv:1601.07140
19. Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: ACCV, pp. 770–783 (2010)
20. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: CVPR, pp. 3538–3545 (2012)
21. Yin, X.-C., Yin, X.-W., Huang, K.-Z., Hao, H.-W.: Robust text detection in natural scene images. IEEE Trans. PAMI **36**(5), 970–983 (2014)
22. Huang, W.-L., Qiao, Y., Tang, X.-O.: Robust scene text detection with convolutional neural networks induced MSER trees. In: ECCV, pp. 497–511 (2014)
23. Sun, L., Huo, Q., Jia, W., Chen, K.: A robust approach for text detection from natural scene images. Pattern Recogn. **48**(9), 2906–2920 (2015)
24. Yin, X.-C., Pei, W.-Y., Zhang, J., Hao, H.-W.: Multi-orientation scene text detection with adaptive clustering. IEEE Trans. PAMI **37**(9), 1930–1937 (2015)

25. Lu, S.-J., Chen, T., Tian, S.-X., Lim, J.-H., Tan, C.-L.: Scene text extraction based on edges and support vector regression. IJDAR **18**(2), 125–135 (2015)
26. Gomez, L., Karatzas, D.: A fast hierarchical method for multi-script and arbitrary oriented scene text extraction. IJDAR **19**(4), 335–349 (2016)
27. Fabrizio, J., Robert-Seidowsky, M., Dubuisson, S., Calarasanu, S., Boissel, R.: TextCatcher: a method to detect curved and challenging text in natural scenes. IJDAR **19**(2), 99–117 (2016)
28. Gomez, L., Karatzas, D.: TextProposals: a text-specific selective search algorithm for word spotting in the wild. Pattern Recogn. **70**, 60–74 (2017)
29. Wang, T., Wu, D.-J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: ICPR, pp. 3304–3308 (2012)
30. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: ECCV, pp. 512–528 (2014)
31. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: CVPR, pp. 4159–4167 (2016)
32. Yao, C., Bai, X., Sang, N., Zhou, X.-Y., Zhou, S.-C., Cao, Z.-M.: Scene text detection via holistic, multi-channel prediction. Preprint (2016). arXiv:1606.09002
33. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. IJCV **116**(1), 1–20 (2016)
34. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localization in natural images. In: CVPR, pp. 2315–2324 (2016)
35. Tian, Z., Huang, W.-L., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: ECCV, pp. 56–72 (2016)
36. Shi, B.-G., Bai, X., Belongiey, S.: Detecting oriented text in natural images by linking segments. In: CVPR, pp. 2550–2558 (2017)
37. Hu, H., Zhang, C.-Q., Luo, Y.-X., Wang, Y.-Z., Han, J.-Y., Ding, E.: WordSup: exploiting word annotations for character based text detection. In: ICCV, pp. 4940–4949 (2017)
38. Jung, K., Kim, K., Jain, A.: Text information extraction in images and video: a survey. Pattern Recogn. **37**(5), 977–997 (2004)
39. Renton, G., Soullard, Y., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Fully convolutional network with dilated convolutions for handwritten text line segmentation. IJDAR **21**(3), 177–186 (2018)
40. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
41. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
42. Deng, D., Liu, H.-F., Li, X.-L., Cai, D.: Pixellink: detecting scene text via instance segmentation. In: AAAI (2018)
43. Lin, T.-Y., Goyal, P., Girshick, R.B., He, K.-M., Dollár, P.: Focal loss for dense object detection. In: ICCV, pp. 2980–2988 (2017)
44. He, K.-M., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV, pp. 2980–2988 (2017)
45. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
46. He, K.-M., Zhang, X.-Y., Ren, S.-Q., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
47. Girshick, R.B.: Fast R-CNN. In: ICCV (2015)
48. Gomez, R., Shi, B.-G., Gomez, L., Neumann, L., Veit, A., Matas, J., Belongie, S., Karatzas, D.: ICDAR2017 robust reading challenge on COCO-Text. In: ICDAR, pp. 1435–1443 (2017)
49. Liu, X.-B., Liang, D., Yan, S., Chen, D.-G., Qiao, Y., Yan, J.-J.: FOTS: fast oriented text spotting with a unified network. In: CVPR, pp. 5676–5685 (2018)
50. Girshick, R.B., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.-M.: Detectron (2018). https://github.com/facebookresearch/detectron
51. Lyu, P.-Y., Yao, C., Wu, W.-H., Yan, S.-C., Bai, X.: Multi-Oriented scene text detection via corner localization and region segmentation. In: CVPR, pp. 7553–7563 (2018)
52. Liao, M.-H., Zhu, Z., Shi, B.-G., Xia, G.-S., Bai, X.: Rotation-sensitive regression for oriented scene text detection. In: CVPR, pp. 5909–5918 (2018)