Improved Localization Accuracy by LocNet for Faster R-CNN Based Text Detection in Natural Scene Images

Zhuoyao Zhong[a,b,*], Lei Sun[b], Qiang Huo[b]

[a]*School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China*
[b]*Microsoft Research Asia, Beijing 10080, China*

Abstract

Although Faster R-CNN based approaches have achieved promising results for text detection, their localization accuracy is not satisfactory in certain cases. In this paper, we address this problem and propose to use a LocNet to improve the localization accuracy of a Faster R-CNN based text detector. Given a proposal generated by the region proposal network (RPN), instead of predicting directly the bounding box coordinates of the concerned text instance, the proposal is enlarged to create a search region so that conditional probabilities to each row and column of this search region can be assigned, which are then used to infer accurately the concerned bounding box. Furthermore, we present a simple yet effective two-stage approach to converting the difficult multi-oriented text detection problem to a horizontal text detection problem, which extends our approach to robustly detect multi-oriented text with accurate bounding box localization. Experiments demonstrate that the proposed approach boosts the localization accuracy for Faster R-CNN based text detectors significantly. Consequently, our new text detector has achieved superior performance on both horizontal (ICDAR-2011, ICDAR-2013 and MULTILIGUL) and multi-oriented (MSRA-TD500, ICDAR-2015) text detection benchmark tasks.

Keywords: Text detection; Text localization accuracy; Faster R-CNN; LocNet; Natural scene images

1. Introduction

Text in natural scene images contains rich and valuable semantic information, which is beneficial to a variety of content-based visual applications, e.g., image and video retrieval, scene understanding and target geolocation. Consequently, text detection in natural scene images has gained increasing attention from document analysis

* Corresponding author. Tel.: +86 13570319907.
E-mail address: zhuoyao.zhong@gmail.com (Z. Zhong), lsun@microsoft.com (L. Sun), qianghuo@microsoft.com (Q. Huo).

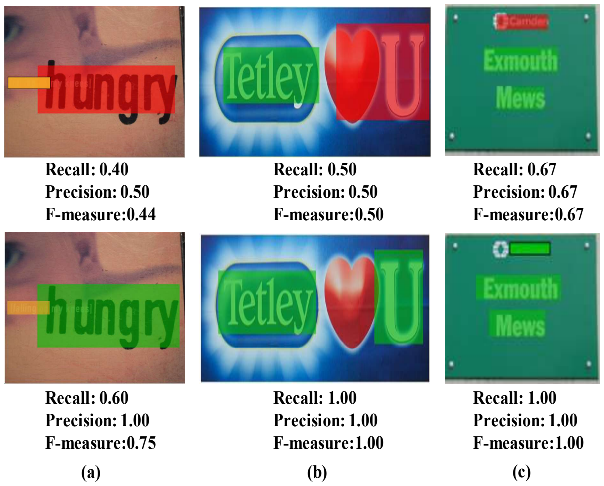


Fig. 1. Detection results of Faster R-CNN with bounding box regression module (1st row) and with LocNet based localization module (2nd row) on ICDAR-2013 dataset. Green and orange regions are correctly detected text regions, while red ones are wrongly detected text regions. Visulization results are captured from the online evaluation system (http://rrc.cvc.uab.es/?ch=2). (Best viewed in color)

and computer vision communities in recent years [1, 2, 3]. However, owing to extremely complex backgrounds, diverse text variabilities in colors, fonts, orientations, languages and scales, as well as highly interference factors like non-uniform illumination, low contrast, low resolution and occlusion, text detection in natural scene images remains a challenging and unsolved problem. Furthermore, the requirement on accurate bounding box prediction poses an additional challenge to this domain-specific task.

Existing text detection methods can be roughly divided into two mainstream categories: bottom-up [4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and top-down methods [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]. Bottom-up methods are generally composed of three major steps [30], i.e., candidate text connected components (CCs) extraction (e.g., based on MSER [31] or SWT [5]), text/non-text classification and text-line grouping. One of the most popular bottom-up methods are the MSER based methods [32, 6], which won the first places in both ICDAR-2011 [1] and ICDAR-2013 [2] robust reading competitions. However, bottom-up methods still have some notable limitations. For example, some text in natural scene images cannot be extracted by the current candidate text CC extraction methods like MSER or SWT, which affects the recall rate of bottom-up methods severely [8]. 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 [19].

Nowadays, with the rapid development of deep learning, the convolutional neural network (CNN) based top-down approaches become more and more promising. [18, 19, 20] borrow the idea of semantic segmentation and employ a fully convolutional neural network (FCN) [33] 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 [18], so complex post-processing steps are needed to extract accurate bounding boxes of text-lines. Compared with FCN based methods, another group of methods [21, 22, 23, 24, 25, 26, 27, 28, 29], which apply popular CNN based object detection frameworks like R-CNN [34], YOLO [35], RPN [36], Faster R-CNN [36], SSD [37] and DenseBox [38], are more straightforward and detect text instances from images directly. All these methods rely on a crucial rectangular or quadrilateral bounding box regression module, which uses a regression function to directly predict the object bounding box coordinates. However, this module is considered as sub-optimal for bounding box prediction and may affect the localization accuracy of the text detector, which is because directly regressing the coordinates of the target bounding box is a difficult learning task that cannot yield accurate enough bounding box [39]. In this paper, we will take Faster R-CNN based approach for example and present a study to address this problem.

Faster R-CNN is the most representative and highest accuracy generic object detection method [40] and has also achieved promising results on text detection tasks [25, 41]. However, as illustrated in the first row of Fig. 1, its localization accuracy is unsatisfactory in certain cases, which is caused by partial text detection and excessive text detection. Partial text detection means that the detected bounding box (red region in 1st row of Fig. 1 (a)) partially covers the concerned text instance (without satisfying the default area recall threshold $t_r = 0.8$ [42]), while excessive text detection is that the detected bounding box (red region 1st row in Fig. 1 (b-c)) is too loose (without satisfying the default area precision threshold $t_P = 0.4$ [42]). The unsatisfactory text localization accuracy not only degrades the performance of text detection task, but also affects the performance

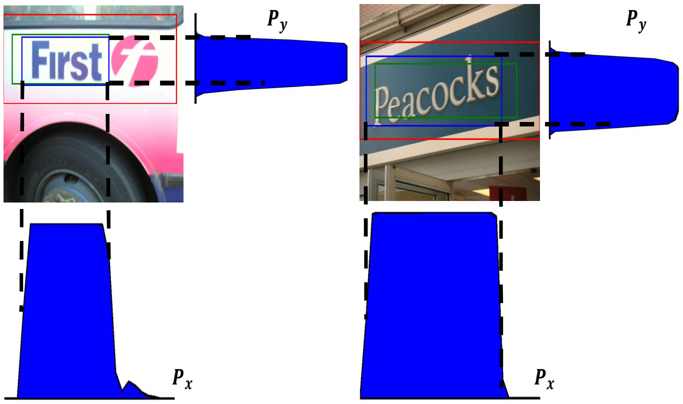


Fig. 2. Illustration of bounding box prediction process with a LocNet localization module. For each natural scene image, green, red and blue rectangles represent input proposal B, search region R and final predicted bounding box, respectively. We visualize p_x and p_y conditional probability vectors at the bottom and on the right of each image. By maximizing the likelihood of the In-Out element probabilities, we can accurately infer the bounding box location of the concerned text instances. (Best viewed in color)

of the succeeding text recognition task. Therefore, improving the text localization accuracy of these approaches is important and necessary.

To address the above problem, we propose to incorporate a LocNet based localization module [39, 43] into the Faster R-CNN framework to improve its localization accuracy. Specifically, given a proposal generated by the RPN [36], instead of predicting directly the coordinates of the concerned text instance, we firstly enlarge the proposal to create a search region and then assign conditional probabilities to each row and column of this region. These conditional probabilities provide useful and detailed information to measure how likely each row and column of the search region is inside the bounding box of the concerned text instance, based on which the bounding box location of the text instance can be inferred accurately. The illustration of bounding box prediction process with a LocNet localization module is depicted in Fig. 2.

Although above approach can achieve promising results on horizontal text detection task, it cannot deal with multi-oriented text detection problem effectively. This is because when using axis-aligned rectangles to represent the bounding boxes of oriented text instances as the original Faster R-CNN, the bounding boxes of nearby oriented text instances will be highly overlapped (Fig. 3(a)). If the Intersection-over-Union (IoU) overlap

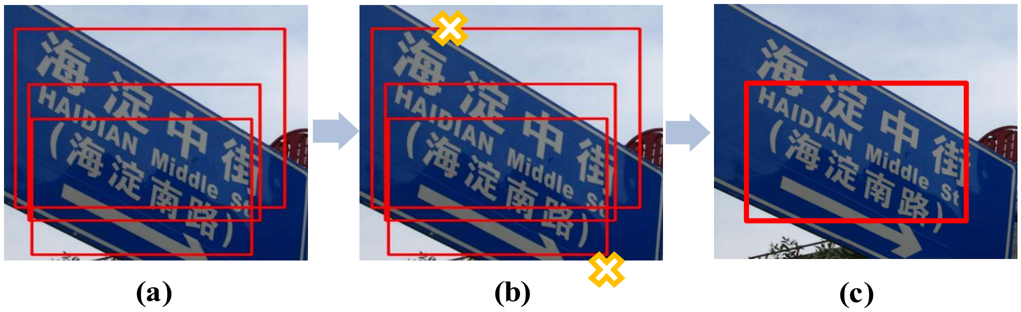


Fig. 3. Performing NMS algorithm on nearby oriented text instances. (a): nearby oriented text instances; (b) NMS algorithm performing, (c): results after NMS.

between two detected bounding boxes is larger than a pre-defined threshold, one of them will be suppressed by the non-maximum suppression (NMS) algorithm [36] (Fig. 3(b)), which will affect the recall rate of both RPN and Fast R-CNN for multi-oriented text detection (Fig. 3(c)). To solve this problem, we propose a simple yet effective two-stage approach to converting the difficult multi-oriented text detection problem to an easier horizontal text detection problem, so that the advantages of LocNet based localization module can also be taken to improve the localization accuracy of multi-oriented text detection. Specifically, the pipeline of the proposed two-stage multi-oriented detection approach is described in Fig. 4. In the first stage, we extract candidate text-line regions and estimate their orientations at the same time. Next, nearby oriented text regions are clustered into a text-block region which is then enlarged and de-rotated based on the estimated orientation. In this way, the text-lines in this region will become horizontal. In the second stage, we perform text-line-level or word-level text detection for each de-rotated text-block region and map the coordinates of detected bounding boxes back to the original image to get final detection results.

Experiments demonstrate that the proposed approach improves the localization accuracy of Faster R-CNN based text detection method significantly. Owing to this improvement, our proposed approach achieves superior performance not only on horizontal text detection benchmarks (ICDAR-2011, ICDAR-2013, MULTILINGUAL), but also on multi-oriented text detection benchmarks (MSRA-TD500, ICDAR-2015). Moreover, even if our text detection model is not trained with multilingual or handwritten text images, we

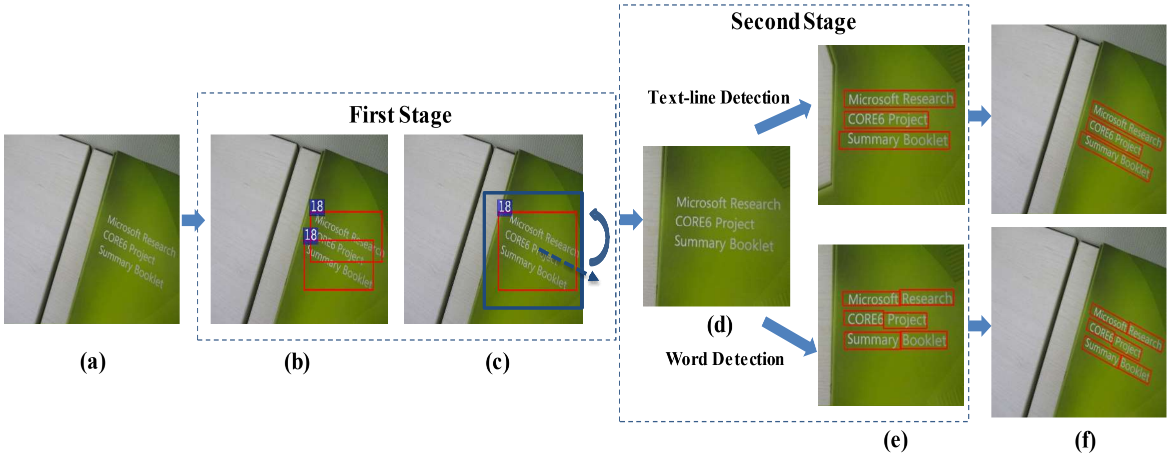


Fig. 4. Pipeline of the proposed two-stage multi-oriented text detection approach: (a) An input image; (b) Candidate text-line region extraction and corresponding orientation estimation; (c) Text-block generation; (d) Enlarging the bounding box of each oriented text-block and de-rotating this enlarged region; (e) Second-stage horizontal text-line or word detection; (f) Final detection results.

surprisingly find it perform well in such scenarios, which reflects the strong generalization ability of our proposed approach.

Although some contents of our approach have been published in [44], which concentrated solely on the horizontal text detection, this paper extends in the following aspects: (1) a simple yet effective two-stage multi-oriented text detection approach is proposed; (2) more technical details and discussions of our approach are presented; (3) more comprehensive related work is reviewed; (4) more experimental results are given. The remainder of this paper is organized as follows. Previous related approaches are summarized in Section 2. The proposed text detection approach and the training strategy are described in detail in Section 3 and Section 4, respectively. Section 5 presents our experimental results and analysis. Finally, the conclusion and discussion are given in Section 6.

2. Related work

2.1. Horizontal text detection

In this part, we focus on horizontal text detection methods built on CNN based object detection methods. Following R-CNN framework, Jaderberg et al. [21] combined complementary region proposal generation methods to generate word region proposals, and then used CNN to filter out non-text proposals and refine the bounding boxes of text proposals. Gupta et al. [22] employed a fully-convolutional regression network, which resembles the YOLO framework [35], to perform text detection and bounding box regression at all locations and multiple scales in an image. Motivated by RPN [36], Tian et al. [23] split a text-line into a sequence of fine-scale text proposals and developed a vertical anchor mechanism to jointly predict text/non-text score and vertical location of each fixed-width proposal with connectionist text proposal network (CTPN). Zhong et al. [41] and Liao et al. [24] adopted modified Faster R-CNN and SSD framework [37] to perform word-level text detection, respectively.

2.2. Multi-oriented text detection

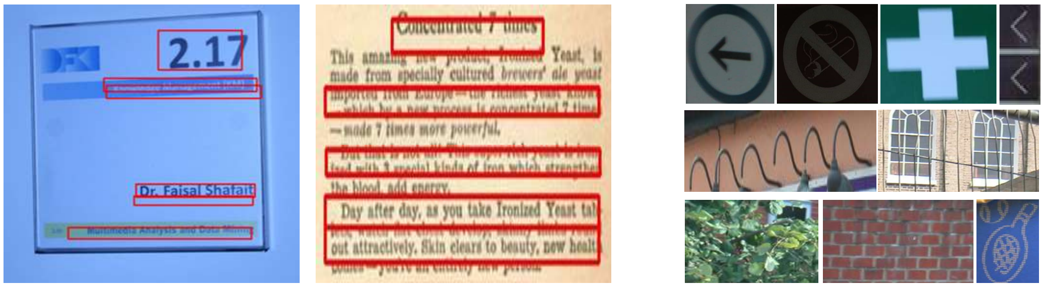
Yao et al. [45] employed SWT to extract candidate text components, then proposed a bottom-up grouping and top-down pruning approach to generating text-lines and filter out non-text components. Yin et al. [46] and Kang et al. [47] used MSER as candidate text CCs, which are then grouped into text-lines with adaptive hierarchical clustering and higher-order correlation clustering algorithms, respectively. After that, non-text lines are filtered out by classifiers. [18, 19, 20] employed FCN to extract text-blocks firstly, from which text-lines were then segmented with post-processing algorithms. [25] applied rotation region proposal networks (RRPN) to generate inclined text proposals and proposed rotation region-of-interest (RROI) pooling layer to project inclined text proposals to the feature map and extract fixed-length feature descriptor for the following text region classier. Similar to [23], [26] decomposed text into two locally elements, e.g. segments and links. A segment covers a part of a word or text-line with oriented bounding box, while a link connects two adjacent segments. Final detections are the combinations of segments that are connected by links. [27] used some predefined quadrilateral sliding windows to hunt for high-quality proposals which can match the multi-oriented text instances with high overlapping area. [28, 29] borrowed the idea of DenseBox [38] and applied CNN to directly predict offsets from bounding box vertexes to points in region of interest. Though some promising results have been achieved, all these approaches contain the sub-optimal bounding box regression module [39].

2.3. Bounding box localization

Most of recent work on text detection [21, 22, 23, 24, 25, 26, 27, 28, 29], or object detection [34, 35, 36, 37, 38] treats bounding box localization problem as a regression problem. A bounding box regression module, given a candidate box that is loosely localized the concerned object, directly predicts the coordinates of the target bounding box. Though the regression ability of this module is improved by the powerful representation learning ability of convolutional neural network recently, its localization accuracy is still unsatisfactory in certain cases. For example, the performance of object detection models including the above regression module degrades a lot under the much stricter COCO-style evaluation [48]. To solve this problem, Gidaris et al. proposed an iterative localization method [49], which iteratively predicts the coordinates of bounding boxes by a regression module. Although this method could improve localization accuracy to some extent, it cannot overcome the limitation of the bounding box regression module inherently. Recently, [39] propose a new LocNet based localization module, which formulates bounding box localization problem as a dense classification problem. This approach firstly enlarges the given candidate box to create a search region and then assign conditional probabilities to each row and column of this search region, which are used to infer accurately the concerned bounding box. This method achieves promising results on general object detection tasks owing to the significant localization accuracy improvement. For text detection task, text localization accuracy is a more important but under-researched problem. In this paper, we attempt to address this problem and propose to incorporate above LocNet based localization module into the Faster R-CNN based text detectors to improve their localization accuracy.

3. Our text detection approach

In this section, we firstly figure out the specific challenges for Faster R-CNN in text detection and then introduce our improved Faster R-CNN based text detection approach. Based on the improved Faster R-CNN based text detector, we illustrate how to use a LocNet based localization module to improve its localization accuracy for better text detection performance.



(a) **(b)**

Fig. 5. (a) Failure cases of Faster R-CNN on small text detection; (b) Text-like backgrounds with very similar texture as text.

3.1. Our improved Faster R-CNN based text detection approach

Faster R-CNN is composed of two core modules: 1) Region Proposal Network (RPN), which proposes rectangular regions of interest (ROIs) from input images; and 2) a Fast R-CNN detector, which classifies extracted region proposals and refines the corresponding object bounding boxes. With the alternating optimization [36] or the approximate joint training [50], RPN and Fast R-CNN can be trained to share convolutional features. Both RPN and Fast R-CNN extract or pool feature from the last full-image shared convolutional layer (e.g., “Conv5_3” for VGG16). In our experiments, the standard VGG16 network [51] is used as a base network architecture.

Theoretically, Faster R-CNN can be used directly to address the horizontal text detection as the rectangular bounding boxes of horizontal text instances are not highly overlapped. However, it has not achieved promising enough results on horizontal text detection benchmarks as presented in [41, 25]. We find that this unsatisfactory performance can be attributed to two specific difficulties of text detection, i.e., small text size and text-like backgrounds. In this paper, we propose to employ skip pooling [52] and online hard example mining (OHEM) [53] to effectively address these two issues. The architecture of our Faster R-CNN based horizontal text detection model is shown in Fig. 6. Details are described in Sec. 3.1.1 and Sec. 3.1.2.

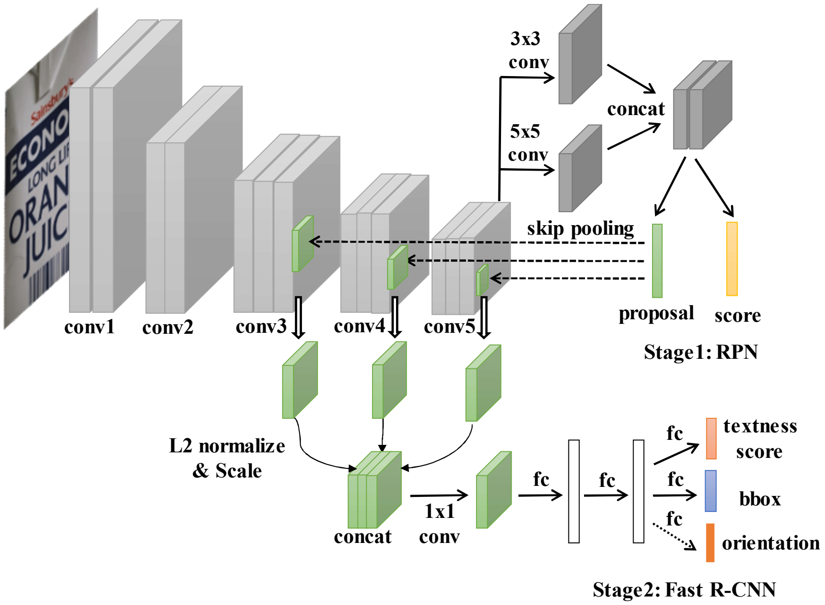


Fig. 6. Architecture of our Faster R-CNN based horizontal text detection baseline model and 1st stage multi-oriented text detection model. The orientation estimation output is only used in the 1st stage multi-oriented text detection model.

3.1.1. ***Skip pooling***

Different from the generic objects, text instances in the natural scene image are of very small size, especially in "incidental scene text" scenarios [3]. Moreover, in practice, for the sake of lower computation and memory usage, we have to resize large images to smaller sizes, which will make the small text instances even smaller. For example, in ICDAR-2013 testing set, when an input image is rescaled such that its shorter side is 500 pixels, about 23% and 86% of text instances' heights are smaller than 16 and 64 pixels, respectively. If the size of a text proposal is as small as 16×16 pixels, the size of its output in the last shared convolutional layer ("Conv5_3") is only one pixel, which does not contain enough information for text/non-text classification and bounding box localization in Fast R-CNN, although RPN can extract good enough region proposals. It is for this reason that Faster R-CNN detector not only misses some text instances of small size, but also struggles with bounding box localization, as shown in Fig. 5 (a).

To overcome this problem, we borrow the idea of skip pooling [52] and combine the features from the "Conv3_3" and "Conv4_3" layers, which have higher resolution and contain more detailed information for small text instances, with more abstract features from the "Conv5_3" layer to enhance the feature representation ability for small text. We find that incorporating skip pooling into Faster R-CNN framework offers a good solution to address small text detection.

The architecture of our text detection network is illustrated in Fig. 6. The architecture design follows Faster R-CNN [36] and ION [52]. Given an input image, VGG16 features of the full image are calculated firstly. Then we slide a small network, which is a mixture of 3×3 and 5×5 convolution, over the feature maps from "Conv5_3" to generate rectangular text proposals. After that, for each proposal, three feature descriptors with a fixed spatial extent of 7×7 are extracted from the "Conv3_3", "Conv4_3" and "Conv5_3" layers using ROI pooling [54], then are L2-normalized and re-scaled back by a learnable per-channel scale parameter (initialized to 2 measured on our training set). These normalized descriptors are then channel-wise concatenated and dimension reduced with a 1×1 convolution layer to obtain a fixed-length feature descriptor of size $512 \times 7 \times 7$, which is fed into two fully-connected layers (fc_6 and fc_7 layers of VGG16 model [51]) for text/non-text classification and bounding box regression.

3.1.2. ***OHEM***

As shown in Fig. 5(b), some backgrounds like signs, fences and bricks have very similar texture as characters or text-lines. As analyzed in [53], these difficult background regions cannot be suppressed effectively by the original mini-batch sampling strategy of the Fast R-CNN detector, which causes lots of false alarms. In order to address this problem, [53] proposed the OHEM algorithm for Fast R-CNN, which automatically selects hard examples that the current model performs worst and feeds them to the model again in order to make model training more effective and efficient. In this paper, we incorporate OHEM algorithm into the end-to-end training process of Faster R-CNN to improve its robustness to the hard text-like backgrounds. Note that there are two modifications in our implementation. Firstly, rather than computing losses for total proposals generated by RPN, we random select N_b background and N_p positive proposals to compute losses for computation and memory saving. Secondly, we still keep the ratio of the positive hard proposals and background hard proposals to 1:3

for each mini-batch to overcome the data imbalance problem existing in the original OHEM algorithm. The implementation detail is described in Sec. 4.2 on end-to-end training strategy.

3.1.3. ***Two-stage approach for multi-oriented text detection***

Even with above improvements, Faster R-CNN still cannot deal with multi-oriented text detection effectively as illustrated in Fig. 3. In order to extend our approach to multi-oriented text detection, we propose a simple yet effective two-stage approach to tactfully converting the difficult oriented text detection problem to a relatively easier horizontal text detection problem. Details of our approach are depicted in Fig. 4.

In the first stage, similar to horizontal text detection, we also use Faster R-CNN to extract the axis-aligned rectangular bounding boxes of candidate text-lines firstly. The difference is that, other than the bounding box, the orientation angle of each text-line is also estimated at the same time (Fig. 4 (a-b)). The architecture of the model used in this stage, which is called 1st stage multi-oriented text detection model, is similar to horizontal text detection model except that a new orientation estimation output is added to the Fast R-CNN module. As analyzed in Sec. 1, some oriented text instances could be missed due to the applied NMS algorithm (See Fig. 3). To overcome this problem, we cluster the nearby extracted oriented text-lines with similar estimated orientation angles into a text-block. Each text-block's orientation angle θ_{blcok} is estimated by calculating the average orientation angle of the text-lines in this text-block (Fig. 4 (c)). The minimum bounding rectangle enclosing all the text-lines in each text-block is taken as the bounding box of that text-block (red rectangle in Fig. 4(c)). After that, we enlarge the bounding box of each text-block by increasing its width and height by the enlargement factors E_w and E_h respectively to include more contextual cues around it (blue rectangle in Fig. 4(c)). The enlarged region is then de-rotated according to its orientation angle θ_{blcok} (Fig. 4 (c-d)) so that the text-lines in this text-block are in the horizontal direction.

In the second stage, according to the requirements of different tasks, we use the Faster R-CNN based horizontal text detection model to extract text-lines or words for each de-rotated text-block region (Fig. 4 (e)). Finally, the coordinates of detected bounding boxes are mapped back to the original image (Fig. 4 (f)). A small

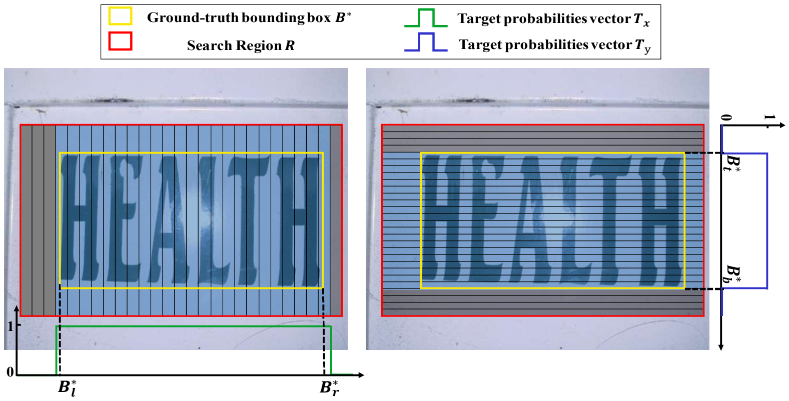

Ground-truth bounding box B^*
Search Region R
Target probabilities vector T_x
Target probabilities vector T_y
HEALTH
HEALTH
1
0
B_l^*
B_r^*
0
1
B_t^*
B_b^*

Fig. 7. Illustration of target probability vectors $T = \{T_x, T_y\}$. The search region R is divided into M equal columns and M equal rows separately. A row or column element is considered to be inside the ground-truth bounding box $B^* = \{B_l^*, B_t^*, B_r^*, B_b^*\}$ if at least part of the region corresponding to this row or column is inside this box, which is assigned value 1, otherwise, assigned 0.

portion of detected results could be overlapped with each other, so the Skewed NMS algorithm for polygons [25] is used to suppress redundant oriented bounding boxes further to get the final results.

3.2. Faster R-CNN with LocNet based localization module for text detection

3.2.1. Horizontal text detection

In order to improve the text localization accuracy of above Faster R-CNN based text detector, we propose to replace the bounding box regression module of Fast R-CNN with a LocNet based localization module. Specially, given an input proposal B, we increase the width and height of B by the enlargement factors S_w and S_h separately to create a search region R. Then we divide R into M equal vertical regions (columns) and M equal horizontal regions (rows) respectively, and output a conditional probability to each column or row. Here, we use the In-Out probabilities [39] for the conditional probabilities, and define two conditional probability vectors $p_x = \{p_x(i)\}_{i=1}^{M}$ and $p_y = \{p_y(i)\}_{i=1}^{M}$ to represent the conditional probabilities of each column and row of R to be inside the bounding box of a text-line respectively. As illustrated in Fig. 7, let B^* be the ground-truth bounding box and (B_l^*, B_t^*) and (B_r^*, B_b^*) be its top-left and bottom-right coordinates, then the target conditional probability vectors $T = \{T_x, T_y\}$ can be denoted as follows:

$$\forall i \in \{1, \dots, M\},\ T_x(i) = \begin{cases} 1, & if\ B_l^* \le i \le B_r^* \\ 0, & otherwise \end{cases}, \tag{1}$$

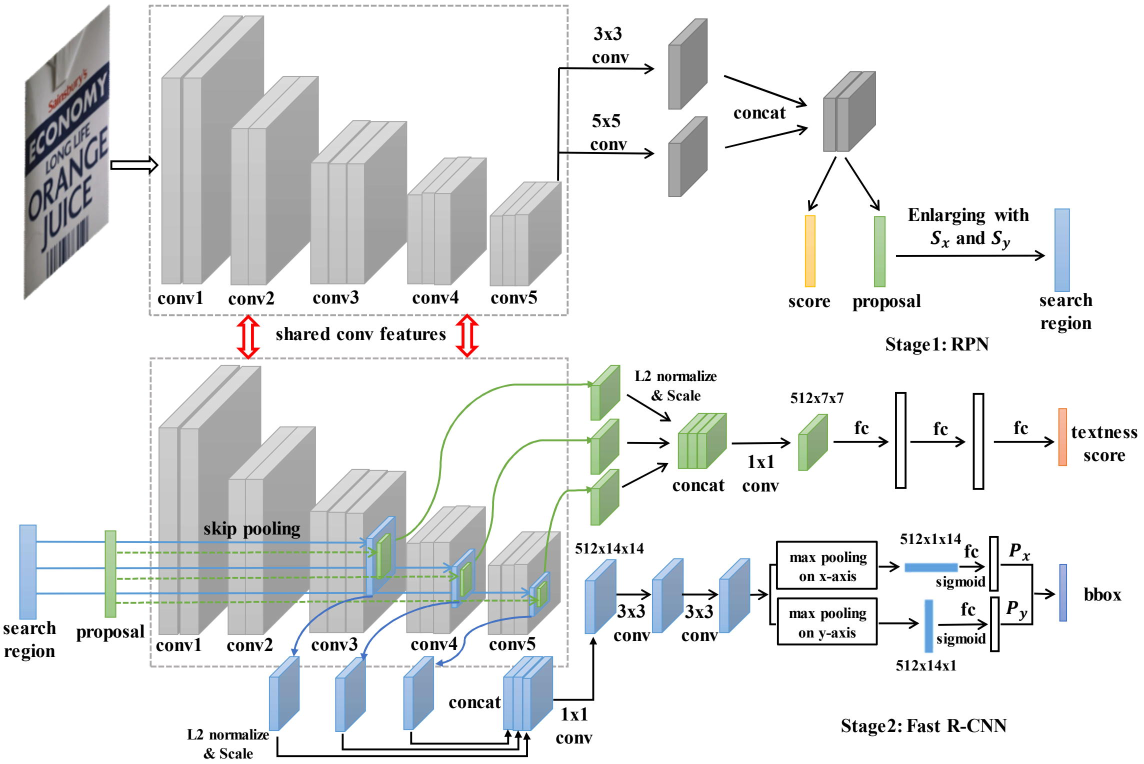


Fig. 8. Architecture of our proposed Faster R-CNN based text detector with LocNet based localization module

$$\forall i \in \{1, \dots, M\},\ T_y(i) = \begin{cases} 1, & if\ B_t^* \le i \le B_b^* \\ 0, & otherwise \end{cases}. \tag{2}$$

Ideally, the output probability vectors $p = \{p_x, p_y\}$ are expected to be equal to T.

The overall network architecture is shown in Fig. 8, where the RPN and text/non-text classification module of Fast R-CNN are the same as above Faster R-CNN baseline text detection model. To achieve LocNet based localization module, for each search region R, we extract three feature descriptors with a fixed spatial extent of 14×14 from the "Conv3_3", "Conv4_3" and "Conv5_3" layers by using ROI pooling. After L2-normalization, re-scaling, channel concatenation and dimension reduction operations, we obtain a fixed-length feature descriptor of size $512 \times 14 \times 14$, which is followed by two stacked 3×3 convolutional layers. Then, the network is split into the X and Y branches via max-pooling on the x-axis and y-axis respectively. Finally, the pooled feature of each branch is fed into the output layer with M nodes to yield the conditional probabilities after sigmoid normalization. Concretely, the X branch is used to output p_x, while Y branch is to p_y.

Given the predicted In-Out probabilities p_x and p_y , p_x the concerned bounding box location $\tilde{B} = \{\tilde{B}_l, \tilde{B}_t, \tilde{B}_r, \tilde{B}_b\}$ (i.e., the top-left and bottom-right coordinates of $\tilde{B}$) can be inferred by maximizing the likelihood of the In-Out elements of $\tilde{B}$:

$$\underset{\tilde{B}_l,\tilde{B}_t,\tilde{B}_r,\tilde{B}_b}{\operatorname{argmax}} \genfrac{}{}{0pt}{}{\prod_{i\in\{\tilde{B}_l,\ldots\tilde{B}_r\}} p_x(i) \prod_{i\notin\{\tilde{B}_l,\ldots\tilde{B}_r\}}(1 - p_x(i))}{\prod_{i\in\{\tilde{B}_t,\ldots\tilde{B}_b\}} p_y(i) \prod_{i\notin\{\tilde{B}_t,\ldots\tilde{B}_b\}}(1 - p_y(i))}. \tag{3}$$

Experiments demonstrate that the proposed approach improves the localization accuracy of Faster R-CNN based text detection method significantly. Some qualitative comparison examples are presented in the 2nd row of Fig. 1 (a-c).

*3.2.2. **Multi-oriented text detection***

As described in Sec. 3.1.3, the goal of the first stage detection is to fast locate coarse text-line regions for the following text-block generation, so accurate bounding box localization is not the crucial consideration here. Thus, we directly re-use the 1st stage multi-oriented text detection model introduced in Sec. 3.1.3 in this stage. In the 2nd stage, we use the above Faster R-CNN based horizontal text detection model with the proposed LocNet based localization module (Sec. 3.2.1) to accurately detect text-lines or words for each de-rotated text-block region. More qualitative detection results are presented in Fig. 9.

4. Training

4.1. Loss functions

*4.1.1. **Horizontal text detection network***

Loss for RPN: There are two sibling output layers of RPN, i.e., a text/non-text classification layer and a bounding box regression layer. Let c and c^* be the predicted and ground-truth label for the classification task, $t = \{t_x, t_y, t_w, t_h\}$ and $t^* = \{t_x{}^*, t_y{}^*, t_w{}^*, t_h{}^*\}$ represent the four-dimensional parameterized coordinates of

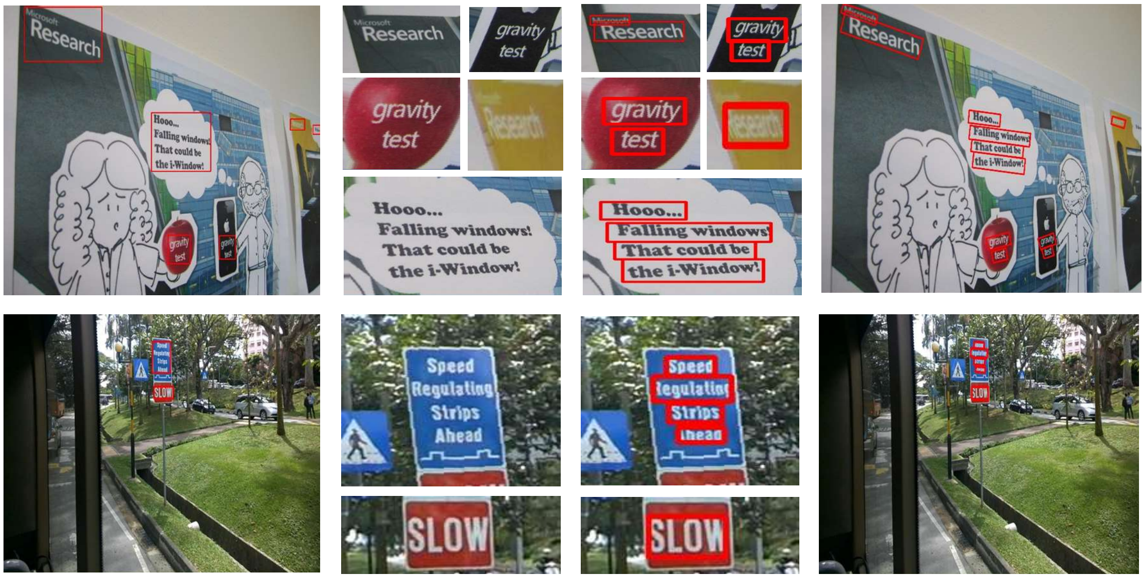


Fig. 9. Qualitative results of our proposed multi-oriented text detection approach. First col: Text-block generation in the first stage. Second col: Text-block de-rotation and enlargement. Third col: Horizontal text detection on de-rotated and enlarged text-block patches. Fourth col: Final detection results.

predicted and ground-truth bounding boxes. We use the parameterizations of t^* stated in [34]:

$$t_x{}^* = \frac{(G_x - A_x)}{A_w},\ t_y{}^* = \frac{(G_y - A_y)}{A_h},$$

$$t_w{}^* = log\left(\frac{G_w}{A_w}\right), t_h{}^* = log\left(\frac{G_h}{A_h}\right), \tag{4}$$

where $A = \{A_x, A_y, A_w, A_h\}$ and $G = \{G_x, G_y, G_w, G_h\}$ denote the center coordinates, width, and height of anchor A and ground-truth box G, respectively. Let $L_{cls}(c, c^*)$ and $L_{reg}(t, t^*)$ denote the classification loss and regression loss. We adopt the standard softmax loss for $L_{cls}(c, c^*)$ and the smooth-L$_1$ loss [54] for $L_{reg}(t, t^*)$. Based on these definitions, the multi-task loss function for RPN is defined as follows:

$$L(k, k^*, t, t^*) = \lambda_{cls} L_{cls}(k, k^*) + \lambda_{reg} c^* L_{reg}(t, t^*), \tag{5}$$

where λ_{cls} and λ_{reg} are two loss-balancing parameters, and we set $\lambda_{cls} = 1, \lambda_{reg} = 3$.

Loss for Fast R-CNN with bounding box regression module: As the above-mentioned RPN, Fast R-CNN with bounding box regression module also apply a text/non-text classification layer and a bounding box regression layer as its sibling output layers. Thus, the multi-task loss function is the same as Equation 5, where we set $\lambda_{cls} = 1, \lambda_{reg} = 1$.

Loss for Fast R-CNN with LocNet based localization module: Fast R-CNN with LocNet based localization module has two sibling output layers: the first one is text/non-text classification layer, which is the same as above RPN and the second one is In-Out probability prediction layer. Let $L_{loc}(p,T)$ denote the loss for In-Out probability prediction and we adopt a binary cross-entropy loss for $L_{loc}(p,T)$ following [39]:

$$L_{loc}(p,T) \ = \sum_{a\in\{x,y\}} \sum_{i=1}^{M} \ T_a(i)\log\big(p_a(i)\big) + \tilde{T}_a(i)\log\big(\tilde{p}_a(\mathrm{i})\big), \tag{6}$$

where $\tilde{T}_a(i) = 1 - \ T_a(i)$ and $\tilde{p}_a(\mathrm{i}) = 1 - p_a(i)$. Then multi-task loss function can be defined as follows:

$$L(c,c^*,p,T) = \lambda_{cls}L_{cls}(c,c^*) \ + \lambda_{loc}c^*L_{loc}(p,T)\ . \tag{7}$$

We set $\lambda_{cls} = 1, \lambda_{loc} = 20$ to bias towards In-Out probability prediction.

*4.1.2. **1st stage multi-oriented text detection network***

The loss function for RPN network is not affected, which is the same as Equation (5). To estimate the orientation angle of each text instance, an orientation estimation layer is added into the output layers of the Fast R-CNN network as shown in Fig. 6. The ground truth orientation angle θ is defined as follows: if $|\theta| \leq \pi/4$, θ is used as the ground truth orientation angle directly, otherwise $\pi/2 - |\theta|$ is used. In this way, θ is limited to the range between $-\pi/4$ to $\pi/4$. All the orientation angles are then uniformly normalized into the range $[-1, 1]$. Let θ and θ^* be the predicted and ground-truth orientation angle of a detected text-line, and let $L_{ori}(\theta,\theta^*)$ denote the loss for orientation estimation. We adopt smooth L_1 loss for $L_{ori}(\theta,\theta^*)$, too. Then the multi-task loss function for the Fast R-CNN network is denoted as follows:

$$L(c,c^*,t,t^*,\theta,\theta^*\ ) = \lambda_{cls}L_{cls}(c,c^*) + \ \lambda_{reg}c^*L_{reg}(t,t^*) + \lambda_{ori}c^*L_{ori}(\theta,\theta^*). \tag{8}$$

We set $\lambda_{cls} = 1, \lambda_{reg} = 1, \lambda_{ori} = 10$ to bias towards better orientation estimation.

4.2. End-to-end training strategy

Our text detection network is trained end-to-end with the approximate joint training algorithm [50]. Details of the training algorithm is depicted in Algorithm 1. Noted that when training model with OHEM, we modify

Algorithm 1 End-to-End Training Strategy

```
Input: A set of training images with ground-truths: $\{(I_1, \{G_1\}), \ldots, (I_N, \{G_N\})\}$; Separate initial network parameters $\mathbf{W^c}, \mathbf{W^p}, \mathbf{W^f}$ for the shared convolutional layers, RPN, and Fast R-CNN; learning rate $\eta(t)$; iteration number $t = 0$.
Step 1: Randomly select one sample $(I_i, \{G_i\})$; produce classification labels and regression targets of anchors according to $\{\mathrm{G_i}\}$;
Step 2: Randomly sample 128 background (IoU<0.1) and 128 positive (IoU>0.5 or the highest IoU) anchors to compute the loss function for RPN;
Step 3: Run backward propagation to obtain the gradient for network parameters $\nabla\mathbf{W}_\mathrm{p}^\mathrm{c}, \nabla\mathbf{W}^\mathrm{p}$ and obtain text proposal set $\{P_i\}$;
Step 4: Adopt NMS with the IoU threshold of 0.7 on $\{P_i\}$ and select the top-2000 ranked proposals to construct $\{D_i\}$ for Step 5;
Step 5: Randomly sample 96 background (IoU<0.3) and 32 positive (IoU>0.5 or the highest IoU) text proposals from $\{\mathrm{D_i}\}$ to compute the loss function for Fast R-CNN;
Step 6: Run backward propagation to obtain the gradient for network parameters $\nabla\mathbf{W}_\mathrm{f}^\mathrm{c}, \nabla\mathbf{W}^\mathrm{f}$;
Step 7: Update the network parameters: $\mathbf{W^c} = \mathbf{W^c} - \eta(t)\left(\nabla\mathbf{W}_\mathrm{p}^\mathrm{c} + \nabla\mathbf{W}_\mathrm{f}^\mathrm{c}\right)$, $\mathbf{W^p} = \mathbf{W^p} - \eta(t) \cdot \nabla\mathbf{W}^\mathrm{p}$,
        $\mathbf{W^f} = \mathbf{W^f} - \eta(t) \cdot \nabla\mathbf{W}^\mathrm{f}$;
Step 8: $t = t + 1$; If the network has converged, end the procedure; otherwise, return to Step 1;
Output: trained parameters $\mathbf{W^c}, \mathbf{W^p}, \mathbf{W^f}$.
```

the step 5 as follows: we randomly select $N_b = 192$ background and $N_p = 64$ positive proposals firstly, then compute the losses for each selected proposal; Next, the positive and background proposals are sorted by losses respectively; Finally, we choose the first half of positive proposals and background proposals to generate a mini-batch for Step 6.

5. Experiments

5.1. Datasets and Evaluation Protocols

We evaluate our approach on several standard benchmark tasks, including ICDAR-2011 [1], ICDAR-2013 [2], MULTILINGUAL [55] datasets for horizontal text detection and more challenging ICDAR-2015 [3] and MSRA-TD500 [45] datasets for multi-oriented text detection. The ICDAR-2011 dataset [1] contains 229 and 255 images for training and testing. The ICDAR-2013 [2] is similar to ICDAR-2011, including 229 training and 233 testing images. The MULTILINGUAL dataset [55] is a multilingual image dataset, which consists of 248 training and 239 testing images captured in natural scenes. The ICDAR-2015 [3] dataset is built for Challenge 4 on Incidental Scene Text of ICDAR-2015 Robust Reading Competition. There are 1000 and 500 images for training and testing and all the images are captured by Google Glass. The ICDAR 2015 dataset is very challenging since there are too many oriented, tiny and low resolution and highly blurred text instances, which is exactly different from focus scene text of ICDAR-2011 and ICDAR-2013. The MSRA-TD500 [45] dataset is a multi-oriented and multilingual dataset, including 300 training images and 200 testing images. Because different evaluation protocols are used for different datasets, we follow the corresponding evaluation protocol for each dataset to make our results comparable to the ones from others. For ICDAR-2011 and MULTILINGUAL datasets, we follow the standard evaluation protocol proposed by Wolf and Jolion [42]. Because the organizers of ICDAR-2015 "Robust Reading Competition" provided online evaluation tools [3] for the ICDAR-2013 and ICDAR-2015 datasets, we use these tools directly. For MSRA-TD500 dataset, we follow the evaluation protocol defined by the authors of [45].

5.2. Experiments on horizontal text detection

5.2.1. ***Experiments setup***

Training data. Our horizontal text detection model was trained on 3,217 training images, including 1,707 images from the SCUT_FORU dataset [56], 229 images from the ICDAR-2013 training set, 100 images from the SVT training set [57], 239 and 433 images containing only horizontal text-lines selected from the HUST-TR400 [58] and USTB-SV1K [46] datasets respectively, and 509 images from an indoor SVT-like dataset which are not overlapped with any test images in all benchmarks. All the training images were relabeled with accurate text-line bounding boxes.

Implementation details. As sizes and aspect ratios of text-lines vary widely, we modify RPN configuration by using 4 scales {32, 64, 96, 128} and 6 aspect ratios {0.2, 0.5, 0.8, 1.0, 1.2, 1.5}, i.e., 24 anchors, at each sliding position. We use a pre-trained VGG16 model [51] for ImageNet classification to initialize the base network of our text detection model. The weights of new layers for RPN and Fast R-CNN are initialized by using random weights with Gaussian distribution of 0 mean and 0.01 standard deviation. During training, we freeze the first two convolutional layers and fine-tune the remaining layers as [36]. All models are trained for

80K iterations with the initial learning rate of 0.001, which is then divided by 10 at 30K and 60K iterations. The momentum is 0.9 and weight decay is 0.0005. Our experiments are conducted on Caffe framework [59]. We apply a multi-scale training strategy. The scale S is defined as the length of the shortest side of an image. In each training iteration, a selected training image is individually rescaled by randomly sampling S from the set $\{300, 400, 500, 600, 700\}$. For LocNet based localization module, we set $S_w = 2.4, S_h = 1.8, M = 28$ in our experiments. In the testing phase, we select the top-300 proposals generated from RPN for Fast-RCNN.

5.2.2. ***Component evaluation***

In this section, we take the baseline horizontal text detection model for example to illustrate the effectiveness of skip pooling and OHEM for text detection tasks. All the models are trained with the same hyper-parameters for fair comparison and tested on the ICDAR-2013 testing set. All the experiments are based on single-scale ($S = 500$) and single model testing.

Skip pooling from which layers? We train three different models by combining features from different convolutional layers (i.e., "Conv3_3" (C3), "Conv4_3" (C4) and "Conv5_3" (C5)). The results of these three models are presented in the first part of Table 1. It can be seen that the third model which combines features from C3, C4 and C5 layers achieves the best F-measure of 88.35%, outperforming the baseline model which pools features only from the C5 layer by 14.04% in F-measure, which highlights the effectiveness of the skip pooling approach for text detection task. Here the features from the first two convolutional layers ("Conv1_2" and "Conv2_2") are not exploited because we freeze these two layers during the training process.

How skip pooling works? We intend to fully explore how skip pooling boosts the text detection performance. Firstly, we calculate the histogram of text instance (word) heights in the ICDAR-2013 dataset when the image scale S is 500. Then we compare the number of text instances in each range of height which are successfully detected by the first and third models in the first part of Table 1. The results are plotted in Fig. 10. The results show that the model with skip pooling works much better than the model without skip pooling

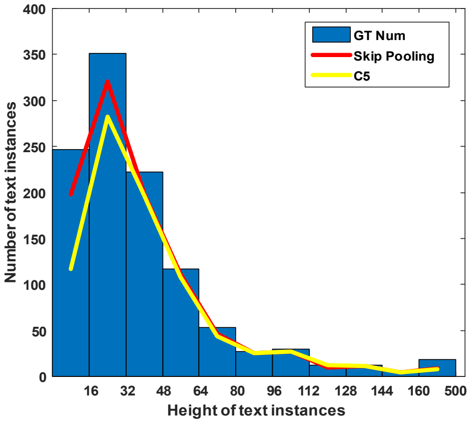


Fig. 10. The histogram of text instance heights in ICDAR-2013 dataset when the image scale is 500. Two curves show the number of text instances in each height range which are detected by the models with and without skip pooling, respectively.

for small text. The performances of these two models are very similar in other height ranges. This experiment demonstrates that the major improvements caused by skip pooling come from small text detection.

OHEM training. As shown in the first part of Table 1, the OHEM training strategy can improve the precision rate from 91.07% to 92.18%, and achieve a better F-measure value of 88.67%.

Table 1. Component evaluation for horizontal text detection on ICDAR-2013 benchmark task.

Localization Module		ROI pooling from			OHEM	Performance (%)		
BBox reg.	LocNet	C3	C4	C5		Recall	Precision	F-measure
√				√		72.89	75.79	74.31
√			√	√		81.92	85.43	83.64
√		√	√	√		85.79	91.07	88.35
√		√	√	√	√	85.41	92.18	88.67
	√			√		79.38	81.23	80.29
	√		√	√		84.86	89.55	87.14
	√	√	√	√		86.70	93.00	89.74
	√	√	√	√	√	**87.53**	**93.93**	**90.61**

*5.2.3. **Comparison between LocNet based localization module and bounding box regression module***

We compare two Faster R-CNN based models on ICDAR-2013, ICDAR-2011 and MULTILINGUAL dataset, which both combine features from C3, C4 and C5 layers and use OHEM training strategy. The only difference between these two models is that the first one adopts the proposed LocNet based localization module (named LocNet model), while the second one applies bounding box regression [36] as the localization module of Fast R-CNN (named BBox reg. model). The results are listed in Table 2. It shows that the LocNet model outperforms the BBox reg. model substantially by improving the F-measure from 88.67% to 90.61%, from 86.60% to 89.41%, from 83.43% to 84.70% on ICDAR-2013, ICDAR-2011 and MULTILINGUAL dataset, respectively.

Table 2. Comparison between LocNet and BBox reg. model on ICDAR-2013, ICDAR-2011 and MULTILINGUAL dataset (%).

Dataset	Model	Recall	Precision	F-measure
ICDAR-2013	LocNet	**87.53**	**93.93**	**90.61**
	BBox reg.	85.41	92.18	88.67
ICDAR-2011	LocNet	**89.82**	**89.00**	**89.41**
	BBox reg.	85.45	87.79	86.60
MULTILINGUAL	LocNet	**84.96**	84.44	**84.70**
	BBox reg.	81.60	**85.36**	83.43

As stated in [42], there are two thresholds on the area recall (t_r) and area precision (t_p), which determine the quality of each match between detected and ground-truth bounding boxes of text instances (we refer readers to review [42] for further details). In order to compare more comprehensively the localization accuracy of these two text detection models, we evaluate detection performance on ICDAR-2011 over the default and stricter t_r and t_p by using the public evaluation tool [42]. Results are listed in Table 3. It is observed that when applying the default value of $t_r = 0.8$ and $t_p = 0.4$, LocNet model outperforms BBox reg. model by 4.37% in recall, 1.21% in precision, 2.81% in F-measure, respectively. Furthermore, where t_r becomes stricter, e.g., $t_r = 0.9$ and $t_r = 1.0$, improvements in F-measure are much more significant, i.e., +7.62% and +4.71%, respectively. Moreover, when t_p becomes stricter, e.g., $t_p = 0.5$ and $t_p = 0.6$, LocNet model can also achieve better results, i.e., outperforming BBox reg. model by +2.59% and +1.93% in F-measure. This demonstrates clearly the effectiveness of the proposed LocNet based localization module for improving the localization accuracy of Faster R-CNN based text detectors.

Furthermore, as shown in the second part of Table 1, LocNet model also outperforms the BBox reg. model substantially when pooling features from C4 and C5 layers or only C5 layer, which also demonstrates that LocNet based localization module can take effect even with insufficient feature representation.

Table 3. Comparison between LocNet and BBox reg. model over various thresholds of the evaluation tool [42] on ICDAR-2011 (%).

Model	Thresholds		Recall	Precision	F -measure	ΔF (%)
	t_r	t_p				
BBox reg.	0.8	0.4	85.45	87.79	86.60	-
	0.9	0.4	72.75	74.86	73.79	-
	1.0	0.4	39.78	40.67	40.22	-
	0.8	0.5	83.51	85.78	84.63	-
	0.8	0.6	75.02	81.18	77.98	
LocNet	0.8	0.4	89.82	89.00	**89.41**	**+2.81**
	0.9	0.4	81.16	81.66	**81.41**	**+7.62**
	1.0	0.4	45.75	44.15	**44.93**	**+4.71**
	0.8	0.5	87.13	87.31	**87.22**	**+2.59**
	0.8	0.6	77.96	81.95	**79.91**	**+1.93**

5.2.4. ***Comparison with prior arts***

We compare the performance of our approach with recently published results on ICDAR-2013 and ICDAR-2011 datasets. As shown in Table 4 and Table 5, our approach achieves the best performance on both datasets. Although ICDAR-2013 is well-tuned by many previous approaches, our approach still achieves the best 87.53%, 93.93% and 90.61% in recall, precision and F-measure, respectively, outperforming the other methods by a notable margin. On ICDAR-2011 dataset, our approach outperforms the closest TextBoxes [24] remarkably by improving the F-measure from 86.00% to 89.41%.

The high performance achieved on both datasets shows the effectiveness and robustness of our approach. As shown in Fig. 11, our detector can detect scene text regions under various challenging conditions such as strong exposure, non-uniform illumination as well as very small scene text with accurate bounding box localization. In terms of run-time, our approach takes about 0.70s on average for each 500×1000 image when using a single M40 GPU.

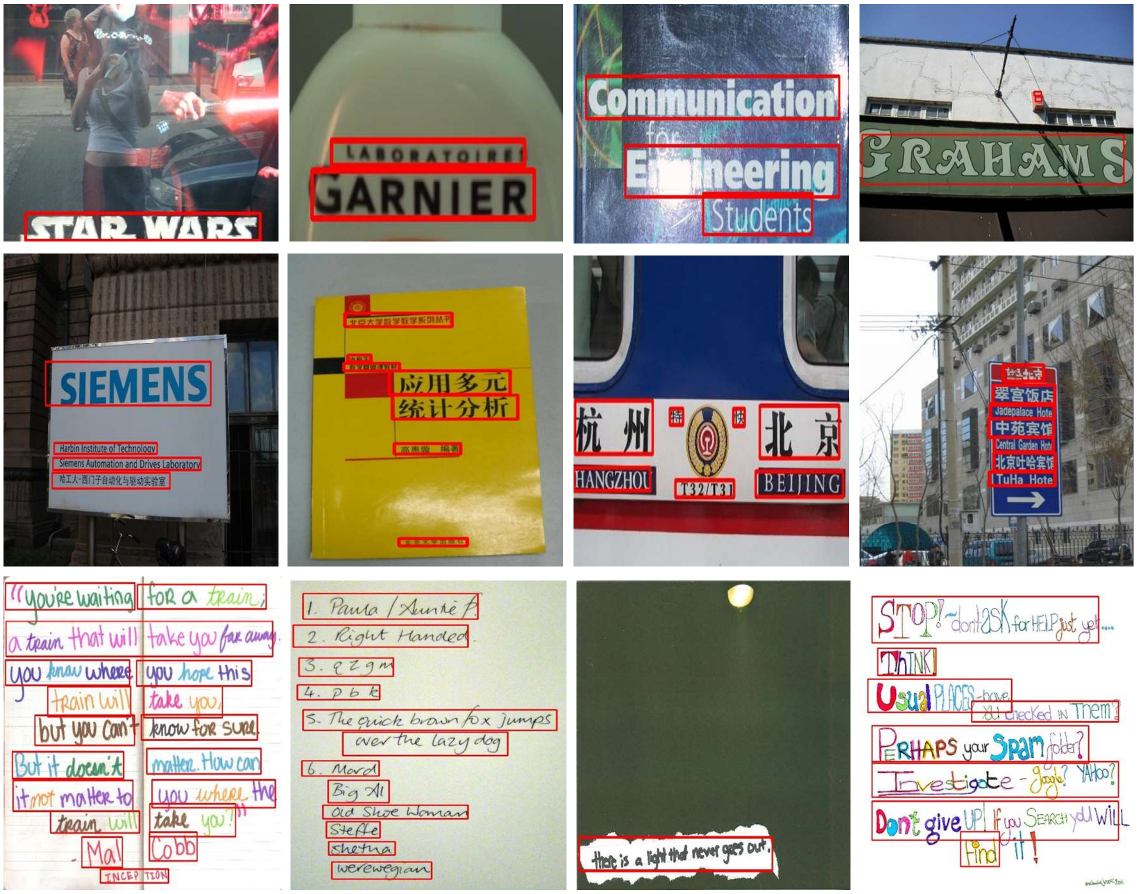

STAR WARS
LABORATOIRE
GARNIER
Communication
for
Engineering
Students
GRAHAMS
SIEMENS
Harbin Institute of Technology
Siemens Automation and Drives Laboratory
哈工大-西门子自动化与驱动实验室
应用多元
统计分析
杭州
北京
HANGZHOU
T32/T31
BEIJING
翠宫饭店
Jadepalace Hote
中苑宾馆
Central Garden Hot
北京计哈宾馆
TuHa Hote
"you're waiting for a train,
a train that will take you far away.
you know where you hope this
train will take you,
but you can't know for sure.
But it doesn't matter. How can
it not matter to you where the
train will take you?"
Mal Cobb
INCEPTION
1. Paula / Auntie P
2. Right Handed
3. ezgm
4. pbk
5. The quick brown fox jumps
over the lazy dog
6. Mard
Big Al
Old Shoe Woman
Steffe
Shetna
werewegian
there is a light that never goes out.
STOP! - don't ask for HELP just yet...
THINK
Usual PLACES have
you checked in Them?
PERHAPS your Spam folder?
Investigate - google? YAHoo?
Don't give UP! If you SEARCH you WILL
Find it!

Fig. 11. Detection results of our horizontal text detection model. First row: ICDAR-2011 and ICDAR-2013. Second row: MULTILINGUAL. Third row: Indoor handwritten text samples. (Best viewed in color).

Table 4. Comparison with prior arts on ICDAR-2013 (%).

Method	Recall	Precision	F-measure
Proposed (LocNet)	**87.53**	**93.93**	**90.61**
CTPN [23]	82.98	92.98	87.69
Zhu et al. [60]	81.02	93.39	86.77
TextBoxes [24]	83.00	89.00	85.89
Zhong et al. [41]	83.00	87.00	85.00
Gupta et al. [22]	75.50	92.00	83.00
TextFlow [61]	75.89	85.15	80.25
1st ICDAR'2013 [2]	69.28	88.80	77.83

Table 5. Comparison with prior arts on ICDAR-2011 (%).

Method	Recall	Precision	F-measure
Proposed (LocNet)	**89.82**	89.00	**89.41**
TextBoxes [24]	82.00	89.00	86.00
CTPN [23]	79.00	89.00	84.00
Zhong et al. [41]	81.00	85.00	83.00
Gupta et al. [22]	74.80	**91.50**	82.30
TextFlow [61]	76.17	86.24	80.89
Zhang et al. [62]	84.00	76.00	80.00
1st ICDAR' 2011 [1]	62.47	82.98	71.28

5.2.5. ***Generalization ability***

It should be noted that the collected 3,217 training images only contain English text. To evaluate the generalization ability of our approach, we test it on MULTILINGUAL dataset as well, which contains both English and Chinese text. The results are quite impressive as shown in Table 6. Our approach achieves the best recall and F-measure, even though some other approaches have used the provided training set. Moreover, we also collect some images containing handwritten text-lines, which are rarely involved in the training set, to test our approach. The detection results in the multilingual and handwritten text images are shown in Fig. 11, which demonstrate that our approach is insensitive to languages or scripts and generalizes well in such scenarios.

Table 6. Comparison with prior arts on MULTILINGUAL (%).

Method	Recall	Precision	F-measure
Proposed (LocNet)	**84.96**	84.44	**84.70**
CPTN [23]	80.00	84.00	82.00
TextFlow [61]	78.40	84.70	81.40
Yin et al. [6]	82.60	68.50	74.60
Pan et al. [55]	64.50	65.90	65.50

5.3. Experiments on Multi-oriented text detection

5.3.1. ***Experimental setup***

Training data. A total of 2,700 real-world oriented text images have been collected from MSRA-TD500 (300 training images) [45], HUST-TR400 (400 images) [58] and USTB-SV1K (1,000 images) [46] and ICDAR-2015 dataset (1,000 training images) [3]. As we need the 1st stage multi-oriented text detection model to estimate different text-line orientations accurately, we prepare a synthetic training set by rotating above 3,217 horizontal text images from $-\pi/2$ to $\pi/2$ at an interval of $\pi/36$. We use this synthetic training data to pre-train the 1st stage multi-oriented text detection model so that this model can be more robust to different text-line orientations. As we want this model to detect text instances in text-line level towards the following text-block generation, the training images are also relabeled with four vertex polygons in the text-line manner. Note that there are many "do not care" ground truth regions in ICDAR-2015 dataset, which are considered as ambiguous regions by us. Hence, we do not use the anchors or proposals in these regions during the training process.

Implementation details. In order to accelerate the convergence speed and improve generalization ability, inspired by the curriculum learning paradigm [63], we use parameters of above well-trained horizontal text detection models in Sec. 5.2 to initialize the models used for multi-oriented text detection. The weights of the new layer for orientation estimation in the 1st stage multi-oriented text detection model are initialized by using random weights with Gaussian distribution of 0 mean and 0.01 standard deviation. Then the 1st stage multi-oriented text detection model is pre-trained on the synthetic training set firstly. In this step, the learning rate is set to 0.0005 for the first 80K iterations and 0.0001 for the next 80K. Subsequently, 2,700 real-world oriented text images are used to fine-tune this model for another 50K iterations with the initial learning rate of 0.0002, which is then divided by 2 at 20K and 40K iterations. To satisfy the requirement of MSRA-TD500 text-line-level and ICDAR-2015 word-level evaluation, we train two different 2nd stage horizontal text detection models for corresponding datasets. For MSRA-TD500, we fine-tune the model on above 2,700 real-world oriented text images, while for ICDAR-2015, the model is fine-tuned on ICDAR-2015 word-level training set. All models in this stage are trained for 50K iterations with the initial learning rate of 0.005, which is then divided by 5 at 20K and 40K iterations. In the testing phase, we set $S = 500$, $E_w = E_h = 1.25$ for MSRA-TD500 dataset, and set $S = 800$, $E_w = 2.4$, $E_h = 1.8$ for ICDAR-2105, respectively.

5.3.2. ***Comparison with prior arts***

The standard ICDAR-2015 and MSRA-TD500 datasets are used to evaluate the performance of our multi-oriented text detection approach and the results are shown in the Table 7 and 8. Note that the evaluation criterion

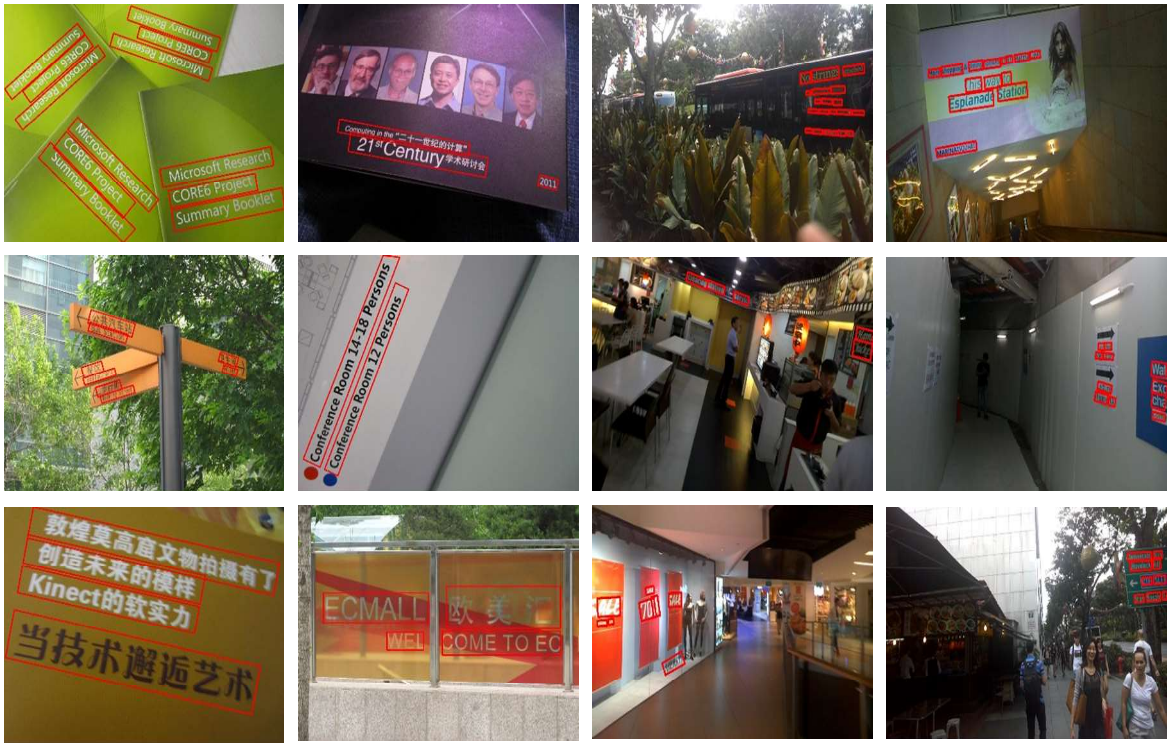


Fig. 12. Detection results of our two-stage multi-oriented text detection approach. First and second col: MSRA-TD500. Third and fourth col: ICDAR-2015. (Best viewed in color).

of these two datasets is based on an IoU threshold of 50%. Although evaluating under such a loose evaluation criterion, our proposed LocNet localization module performed in the 2nd stage still can outperform the bounding box regression module by improving the F-measure from 83.24% to 83.78% and 78.50% to 79.66% on ICDAR-2015 and MSRA-TD500, respectively.

Furthermore, we compare the performance of our approach with prior arts on these two datasets. On the challenging ICDAR-2015 task, our approach outperforms other recently published CNN-based approaches substantially by improving the F-measure from 81.00% to 83.78%. On the MSRA-TD500 dataset, our approach achieves the best F-measure of 79.66% owing to the remarkable improvement in the recall rate of 81.39%. The promising performance on these two challenging datasets reveals the robustness of our proposed two-stage multi-oriented text detection approach. Some detection examples can be seen in Fig. 12, which highlight that our system is strongly capable of detecting arbitrary orientation, complex background and even extremely tiny text regions with accurate bounding box localization.

Table 7. Comparison with prior arts on ICDAR-2015 benchmark task (%).

Method	Recall	Precision	F-measure
Proposed (2nd stage LocNet)	**80.12**	87.81	**83.78**
Proposed (2nd stage BBox reg.)	78.33	**88.81**	83.24
He et al. [29]	80.00	82.00	81.00
Zhou et al. [28]	78.33	83.27	80.27
Ma et al. [25]	73.23	82.17	77.44
Shi et al. [26]	76.80	73.10	75.00
Liu et al. [27]	68.22	73.23	70.64
Yao *et al.* [20]	58.69	72.26	64.77
CTPN [23]	51.56	74.22	60.85
1st ICDAR' 2015 [3]	36.74	77.46	49.84

Table 8. Comparison with prior arts on MSRA-TD500 benchmark task (%).

Method	Recall	Precision	F-measure
Proposed (2nd stage LocNet)	81.39	78.00	**79.66**
Proposed (2nd stage BBox reg)	**81.91**	75.36	78.50
Shi et al. [26]	70.00	86.00	77.00
Zhou et al. [28]	67.43	**87.28**	76.08
Yao et al. [20]	75.31	76.51	75.91
Zhang et al. [18]	67.00	83.00	74.00
Yin et al. [46]	63.00	81.00	71.00
He et al. [19]	65.00	79.00	71.00
Kang et al. [47]	62.00	71.00	66.00
Yin et al. [6]	61.00	71.00	66.00

6. Conclusion and discussion

In this paper, we study how to improve the text localization accuracy of Faster R-CNN based text detectors. We propose to incorporate a LocNet based localization module into the Faster R-CNN based text detectors to improve their localization accuracy. Furthermore, we present a simple yet effective two-stage approach to convert the difficult multi-oriented text detection problem to a relatively easier horizontal text detection problem. Comprehensive evaluations and comparisons are made on five benchmark datasets on which our proposed approach achieves superior performance. Our approach can not only detect robustly text instances under various challenging conditions with accurate bounding box localization, but also generalize well to different languages and scripts.

It is noted that we have not tried to purely push the performance in this paper because the main motivation of this study is to figure out how to improve the text localization accuracy of Faster R-CNN based text detectors. Previous state-of-the-art text detection approaches, which contain the sub-optimal bounding box regression module [39], could also incorporate our proposed LocNet based localization module in their text detectors and thus improve their text detection performance further. Moreover, some widely used techniques such as deeper network topologies (e.g., ResNet [64]), feature pyramid network (FPN) [65], ensemble, etc., can also be used to enhance performance, which could be our future work.

Our approach still has limitations. First, our approach cannot deal with curved text-lines. Second, although our approach can run fast on GPUs, its running speed on CPUs is much slower. More researches are needed to address this challenging problem.

References

[1] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *ICDAR*, 2011, pp. 1491-1496.

[2] D. Karatzas, et al., "ICDAR 2013 robust reading competition," in *ICDAR*, 2013, pp. 1484-1493.

[3] D. Karatzas, et al., "ICDAR 2015 robust reading competition," in *ICDAR*, 2015, pp. 1156-1160.

[4] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *ACCV*, 2010, pp. 770-783.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963-2970.

[6] X.-C. Yin, X.-W. Yin, K.-Z. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. PAMI,* vol. 36, no. 5, pp. 970-983, 2014.

[7] W.-L. Huang, Y. Qiao, and X.-O. Tang, "Robust scene text detection with convolutional neural networks induced MSER trees," in *ECCV*, 2014, pp. 497-511.

[8] L. Sun, Q. Huo, W. Jia, and K. Chen, "A robust approach for text detection from natural scene images," *Pattern Recognition,* vol. 48, no. 9, pp. 2906-2920, 2015.

[9] M. Busta, L. Neumann, and J. Matas, "FASText: Efficient unconstrained scene text detector," in *ICCV*, 2015, pp. 1206-1214.

[10] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *CVPR*, 2012, pp. 3538-3545.

[11] L. Gomez and D. Karatzas, "A fast hierarchical method for multi-script and arbitrary oriented scene text

extraction," *IJDAR,* vol. 19, no. 4, pp. 335-349, 2016.

[12] A. Zhu, R.-W Gao and S. Uchida, "Could scene context be beneficial for scene text detection?," *Pattern Recognition,* vol. 58, pp. 204-215, 2016.

[13] L. Gómez and D. Karatzas, "TextProposals: A text-specific selective search algorithm for word spotting in the wild," *Pattern Recognition,* vol. 70, pp. 60-74, 2017.

[14] K. Wang and S. Belongie, "Word spotting in the wild," in *ECCV*, 2010, pp. 591-604.

[15] T. Wang, D.-J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *ICPR*, 2012, pp. 3304-3308.

[16] R. Minetto, N. Thone, M. Cord, N. J. Leite, and J. Stolfi, "T-HOG: An effective gradient-based descriptor for single line text regions," *Pattern Recognition,* vol. 46, no. 3, pp. 1078-1090, 2013.

[17] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *ECCV*, 2014, pp. 512-528.

[18] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *CVPR*, 2016, pp. 4159-4167.

[19] T. He, W.-L Huang, Y. Qiao, and J. Yao, "Accurate text localization in natural image with cascaded convolutional text network," arXiv preprint arXiv:1603.09423, 2016.

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

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

[22] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localization in natural images," in *CVPR*, 2016, pp. 2315-2324.

[23] Z. Tian, W.-L. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *ECCV*, 2016, pp. 56-72.

[24] M.-H. Liao, B.-G. Shi, X. Bai, X.-G. Wang, and W.-Y. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *AAAI*, 2017, pp. 4161-4167.

[25] J.-Q. Ma, et al., "Arbitrary-Oriented scene text detection via rotation proposals," arXiv preprint arXiv:1703.01086, 2017.

[26] B.-G. Shi, X. Bai and S. Belongiey, "Detecting oriented text in natural images by linking segments," in *CVPR*, 2017, pp. 2550-2558.

[27] Y.-L. Liu and L.-W. Jin, "Deep matching prior network toward tighter multi-oriented text detection," in *CVPR*, 2017, pp. 1962-1969.

[28] X.-Y. Zhou, et al, "EAST: An efficient and accurate scene text detector," in *CVPR*, 2017, pp. 5551-5560.

[29] W.-H. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," arXiv preprint arXiv:1703.08289, 2017.

[30] K. Jung, K. Kim and A. Jain, "Text information extraction in images and video: a survey," *Pattern recognition,* vol. 37, no. 5, pp. 977-997, 2004.

[31] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, 2002, pp. 384-393.

[32] H.-I. Koo and D.-H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. IP,* vol. 22, no. 6, pp. 2296-2305, 2013.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in

CVPR, 2015, pp. 3431-3440.

[34] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580-587.

[35] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779-788.

[36] S.-Q. Ren, K.-M. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91-99.

[37] W. Liu, et al., "SSD: Single shot multiBox detector," in *ECCV*, 2016, pp. 21-37.

[38] L.-C. Huang, Y. Yang, T.-F. Deng, and Y.-N. Yu, "Densebox: Unifying landmark localization with end to end object detection," arXiv preprint arXiv:1509.04874, 2015.

[39] S. Gidaris and N. Komodakis, "LocNet: Improving localization accuracy for object detection," in *CVPR*, 2016, pp. 789-798.

[40] J. Huang, et al, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017, pp. 7310-7319.

[41] Z.-Y. Zhong, L.-W. Jin, and S.-P. Huang, "DeepText: A new approach for proposal generation and text detection in natural images," in *ICASSP*, 2017, pp. 1208-1212.

[42] C. Wolf and J. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *IJDAR,* vol. 8, no. 4, pp. 280-296, 2006.

[43] S. Gidaris and N. Komodakis, "Attend refine repeat: active box proposal generation via In-Out localization," in *arXiv preprint arXiv:1606.04446*, 2016.

[44] Z.-Y. Zhong, L. Sun and Q. Huo, "Improved localization accuracy by LocNet for Faster R-CNN based text detection," in *ICDAR*, 2017.

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

[46] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. PAMI,* vol. 37, no. 9, p. 1930–1937, 2015.

[47] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *CVPR*, 2014, pp. 4034-4041.

[48] T.-Y. Lin, et al., "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740-755.

[49] S. Gidaris and N. Komodakis, "Object detection via a multi-region & semantic segmentation-aware CNN model," in *ICCV*, 2015, pp. 1134-1142.

[50] R. B. Girshick, "Training R-CNNs of various velocities," ICCV-2015 tutorial slides https://github.com/rbgirshick/py-faster-rcnn, 2015.

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

[52] S. Bell, C. L. Zitnick, K. Bala, and R. B. Girshick, "Inside-Outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016, pp. 2874-2883.

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

[54] R. B. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440-1448.

[55] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. IP,* vol. 20, no. 3, pp. 800-813, 2011.

[56] S.-Y. Zhang, M.-D. Lin, T.-S. Chen, L.-W. Jin, and L. Lin, "Character proposal network for robust text extraction," in *ICASSP*, 2016, pp. 2633-2637.

[57] K. Wang, B. Babenko, and S. Belongie, "Eng-to-end scene text recognition," in *ICCV*, 2011, pp. 1457-1464.

[58] C. Yao, X. Bai, and W.-Y. Liu, "A unified framework for multi-oriented text detection and recognition," *IEEE Trans. IP,* vol. 23, no. 11, pp. 4737-4749, 2014.

[59] Y.-Q. Jia, et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.

[60] S.-Y. Zhu and R. Zanibbi, "A text detection system for natural scenes with convolutional feature learning and cascaded classification," in *CVPR*, 2016, pp. 625-632.

[61] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C.-L. Tan, "Textflow: A unified text detection system in natural scene images," in *ICCV*, 2015, PP. 4651-4659.

[62] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *CVPR*, 2015, pp. 2558-2567.

[63] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41-48.

[64] K.-M. He, X.-Y. Zhang, S.-Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770-778.

[65] T.-Y. Lin, et al., "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117-2125.