Improved Localization Accuracy by LocNet for Faster R-CNN Based Text Detection

Zhuoyao Zhong[1,2*], Lei Sun[2], Qiang Huo[2]
[1]School of EIE., South China University of Technology, Guangzhou, China
[2]Microsoft Research Asia, Beijing, China
{v-zhuzho, lsun, qianghuo}@microsoft.com

***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 propose to use a LocNet to improve the localization accuracy of a Faster R-CNN based text detector. Given a proposal generated by 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. Experiments demonstrate that the proposed approach boosts the localization accuracy for Faster R-CNN based text detection significantly. Consequently, our new text detector has achieved superior performance on ICDAR-2011, ICDAR-2013 and MULTILIGUL text detection benchmark tasks.**

Keywords— Accurate text detection; Faster R*‑*CNN; LocNet

I. 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 and computer vision communities recently [1, 2, 3]. However, text detection is still an extremely challenging problem due to following reasons. First, scene text itself is very diverse in terms of languages, fonts, scales, orientations and colors. Second, scene image backgrounds are generally very complex or even have similar textures as text (e.g., signs, fences, bars and bricks). Third, there exist many interference factors like non-uniform illumination, blurring, low contrast and occlusion. Furthermore, the requirement on accurate bounding box prediction poses additional challenge to this domain-specific task.

Existing text detection methods can be roughly divided into two mainstream categories: bottom-up methods [4, 5, 6, 7, 8] and top-down methods [9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. Bottom-up methods extract candidate text connected components (CCs) (e.g., based on MSER [19] or SWT [5]), filter out non-text CCs and group text CCs into text-lines. One of the most popular bottom-up methods are MSER based methods, which achieved state-of-the-art performance in both ICDAR-2011 [1] and ICDAR-2013 robust reading competitions [2]. 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 [16].

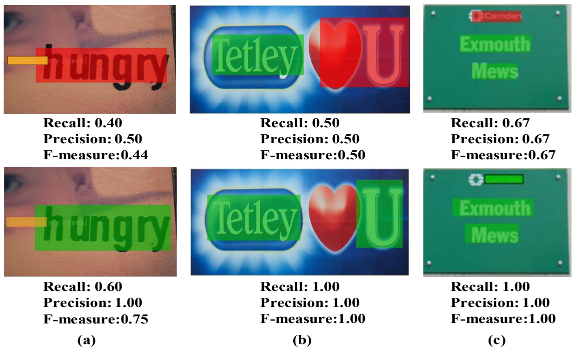

Fig. 1. Detection results of Faster R-CNN with bounding box regression module (1st row) and with LocNet based localization module (2nd row) on ICDAR-2013 dataset. Green and orange regions are correctly detected text regions, while red ones are wrongly detected text regions. Visulization results are captured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

Nowadays, with the astonishing development of deep learning algorithms, convolutional neural network (CNN) based top-down methods become more and more promising. [12, 13] borrow the idea of semantic segmentation and apply a fully convolutional neural network (FCN) [20] 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 [12], so complex post-processing steps are needed to extract accurate bounding boxes of text-lines. Compared with FCN based methods, another group of methods [14, 15, 16, 17, 18, 21] , which adopt CNN based object detection frameworks like R-CNN [22], YOLO [23], RPN [24], SSD [25], Faster R-CNN [24], are more straightforward and detect text instances from images directly. All these methods rely on a crucial bounding box regression module, which uses a regression function to directly predict the object bounding box coordinates. However, this module is

[*]This work was done when Zhuoyao Zhong was an intern in Speech Group, Microsoft Research Asia, Beijing, China.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.155

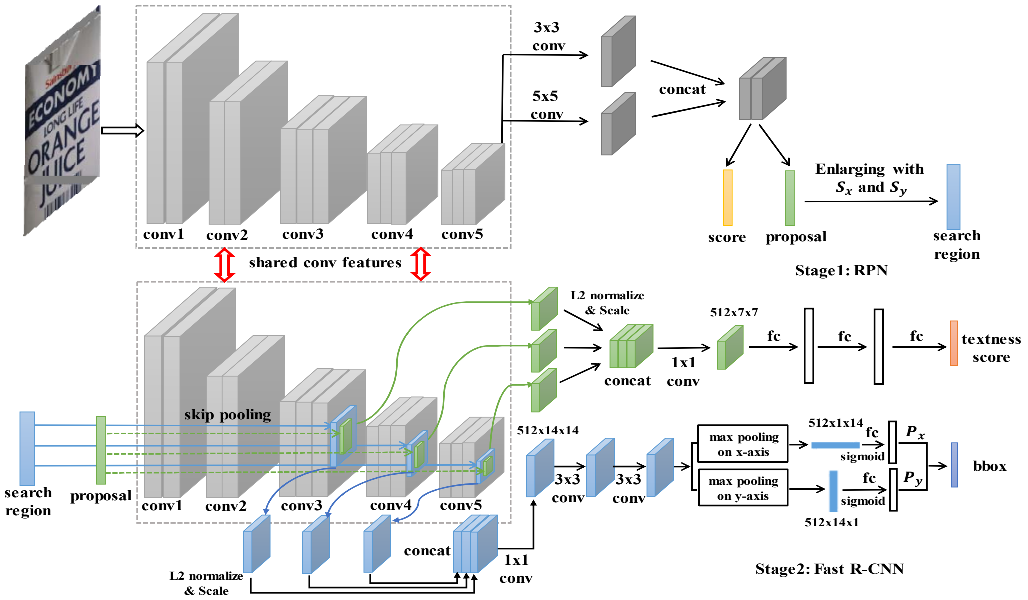

Fig. 2. Architecture of our proposed text detection network.

considered as sub-optimal for bounding box prediction and may affect the localization accuracy of the text detector, because directly regressing the coordinates of the target bounding box is a difficult learning task that cannot yield accurate enough bounding boxes [26]. 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 generic object detection method and has also achieved promising results on text detection tasks [18, 21]. 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 the 1st row of Fig. 1 (a)) partially covers the concerned text instance (without satisfying the default area recall threshold $t_r = 0.8$ [27]), while excessive text detection is that the detected bounding box (red region in the 1st row in Fig. 1 (b-c)) is too loose (without satisfying the default area precision threshold $t_p = 0.4$ [27]). The unsatisfactory text localization accuracy not only degrades the performance of text detection task, but also affects the performance 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 [26] into the Faster R-CNN framework to improve its localization accuracy. Specifically, given a proposal generated by the RPN [24], 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.

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). Owing to this improvement, our new text detector has achieved superior performance on ICDAR-2011 [1], ICDAR-2013 [2] and MULTILIGUL [28] text detection benchmark tasks. Moreover, although our text detection model is not trained with multilingual text images, it generalizes surprisingly well to other languages, which reflects the strong generalization ability of our proposed approach.

II. Text Detection Approach

As depicted in Fig. 2, our text detection network consists of two core sub-networks: (1) Region proposal network (RPN); (2) Fast R-CNN detector, which is composed of two modules, i.e., text/non-text classification module and LocNet based localization module [26]. Given an input image, our approach uses RPN to generate a manageable number of rectangular region proposals for text instances, which are then enlarged to create corresponding search regions. Then, for each region proposal, a text/non-text classification module is used to predict its textness score. If it is classified as a text proposal, its corresponding search region will be taken as the input of a LocNet based localization module to refine the bounding box of the concerned text instance. In this paper, we advocate text-line-level detection, which means that all the ground truth bounding boxes of text instances for training are annotated in a text-line manner. The standard VGG16 network [29] is used as a base network architecture in our experiments, which is shared by both RPN and Fast R-CNN detector. The details of our approach are described as follows.

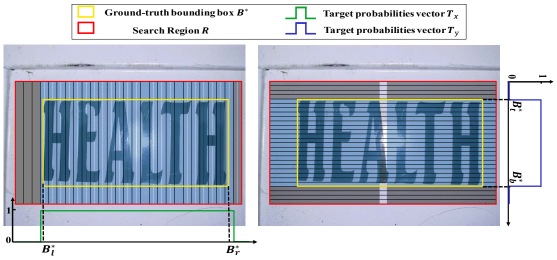
Ground-truth bounding box B^*
Search Region R
Target probabilities vector T_x
Target probabilities vector T_y
HEALTH
HEALTH
0
1
B_l^*
B_r^*
0
1
B_t^*
B_b^*

Fig. 3. 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.

A. RPN

Given an input image, the shared convolutional features of VGG16 model [29] 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". As the variabilities in sizes and aspect ratios of text-lines are higher than general objects, we modify the original 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.

B. Fast R-CNN Detector

Text/non-text classification module: For each proposal, rather than just performing ROI pooling [30] from the high-level "Conv5_3" layer [24], we borrow the idea of skip pooling [31] and combine features from different layers, i.e., "Conv3_3", "Conv4_3" and "Conv5_3", to improve the representation ability of features for small text proposals. The features from the intermediate "Conv3_3" and "Conv4_3" layers have higher resolution and contain more detailed information, which are complementary to more abstract features from the "Conv5_3" layer. Concretely, as depicted in the middle half of Fig. 2, we apply ROI pooling over "Conv3_3", "Conv4_3" and "Conv5_3" layers and extract three feature descriptors with a fixed spatial grid of 7 × 7 for each proposal. Each fixed-size feature descriptor is then L2-normalized and re-scaled back with a learnable per-channel scaling parameter, which is initialized to 2 according to our training set. These descriptors are then concatenated on the channel axis and dimension reduced with a 1 × 1 convolutional layer to obtain a fixed-length feature descriptor of size 512 × 7 × 7, which is fed into two fully-connected layers (fc_6 and fc_7 layers of VGG16 model [29]) for text/non-text classification.

LocNet based localization module: Given an input proposal B, we increase the width and height of B by the enlargement factors[1] 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[2], and output a conditional probability to each column or row. Here, we use the In-Out probabilities [26] 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. 3, 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:

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

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

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

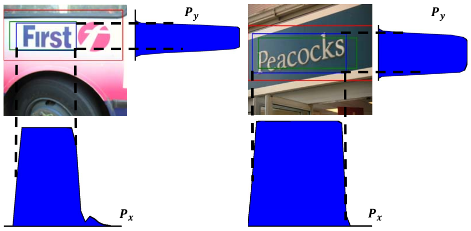

Fig. 4. Illustration of bounding box prediction process with a LocNet localization module. For each natural scene image, green, red and blue rectangles represent input proposal B, search region R and final predicted bounding box, respectively. We visualize In-Out p_x and p_y probability vectors at the 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)

The architecture of the LocNet based localization module is depicted at the bottom part of Fig. 2. The architecture design follows [26, 32]. 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. Similar to text/non-text classification module, after L2-normalization, re-scaling, channel concatenation and dimension reduction operations, we obtain a fixed-length feature descriptor of size 512 × 14 × 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, 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 $\tilde{B}$:

$$\underset{\tilde{B}_l,\tilde{B}_t,\tilde{B}_r,\tilde{B}_b}{\operatorname{argmax}} \frac{\prod_{i\in\{\tilde{B}_l,\dots\tilde{B}_r\}} p_x(i) \prod_{i\notin\{\tilde{B}_l,\dots\tilde{B}_r\}} (1 - p_x(i))}{\prod_{i\in\{\tilde{B}_t,\dots\tilde{B}_b\}} p_y(i) \prod_{i\notin\{\tilde{B}_t,\dots\tilde{B}_b\}} (1 - p_y(i))}. \qquad (3)$$

The visual illustration of bounding box prediction process with the proposed LocNet localization module is depicted in Fig. 4.

III. Training

A. Loss Fucntions

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

$$L_R(c, c^*, t, t^*) = \lambda_{cls} L_{cls}(c, c^*) + \lambda_{loc} c^* L_{loc}^R(t, t^*), \qquad (4)$$

where c and c^* are predicted and ground-truth labels

[1] We use $S_w = 2.4$, $S_h = 1.8$.

[2] We set $M = 28$ in our experiments.

respectively, $L_{cls}(c, c^*)$ is a softmax loss for 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 predicted and ground-truth bounding boxes. We use the parameterizations of t^* stated in [22]:

$$t_x{}^* = \frac{(G_x - A_x)}{A_w},\ t_y{}^* = \frac{(G_y - A_y)}{A_h},$$
$$t_w{}^* = log\left(\frac{G_w}{A_w}\right), t_h{}^* = log\left(\frac{G_h}{A_h}\right), \qquad (5)$$

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. $L^R_{loc}(t, t^*)$ is a smooth L_1 loss [30] for regression task. λ_{cls} and λ_{loc} are two loss-balancing parameters, and we set $\lambda_{cls} = 1, \lambda_{loc} = 3$.

Multi-task loss for Fast R-CNN: Fast R-CNN also has two sibling output layers: the first is text/non-text classification layer, which is the same as the above-mentioned RPN and the second is In-Out probability prediction layer. Let $L^F_{loc}(p, T)$ denote the loss for In-Out probability prediction and we adopt a binary cross-entropy loss for $L^F_{loc}(p, T)$ following [26]:

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

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

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

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

B. End-to-end Training Strategy

Our text detection network is trained end-to-end with an approximate joint training algorithm [33]. The training procedure is described as follows:

Step 1: Randomly select one training image *I* with its ground truth bounding boxes set $\{G\}$;
Step 2: Calculate text labels and regression targets of anchors according to $\{G\}$; 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 corresponding network parameters and proposal set $\{P\}$;
Step 4: Adopt NMS with the IoU threshold of 0.7 on $\{P\}$ and select the top-2000 ranked proposals to construct $\{D\}$ for Step 5;
Step 5: Calculate text labels and target probabilities T of proposals according to {G}; Randomly sample 96 background (IoU<0.3) and 32 positive (IoU>0.5 or the highest IoU) proposals from $\{D\}$ to compute the Fast R-CNN loss function;
Step 6: Run backward propagation to obtain the gradient for corresponding network parameters**;**
Step 7: Update the network parameters;
Step 8: Repeat Step 1-7 until convergence.

IV. EXPERIMENTS

A. Datasets and Evaluation Protocol

We evaluate our approach on three standard benchmarks, namely ICDAR-2011 [1], ICDAR-2013 [2], MULTILINGUAL [28] datasets. 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 [28] is a multilingual image dataset, which consists of 248 training and 239 testing images captured in natural scenes. We follow the corresponding evaluation protocol for each dataset to make our results comparable to others. Concretely, we follow the standard evaluation protocol proposed by Wolf and Jolion [27] for ICDAR-2011 and MULTILINGUAL datasets, while for ICDAR-2013, we use the online evaluation tool provided by the organizers of ICDAR-2015 "Robust Reading Competition" [3].

B. Experimental Setup

Training data. We have collected a total of 3,217 images for our text detection model training, including 1,707 images from SCUT_FORU [34], 229 training images form ICDAR-2013 [2], 100 training images from SVT [35], 239 and 433 images containing only horizontal text-lines selected from the HUST-TR400 [36] and USTB-SV1K [37] datasets respectively and an indoor SVT-like dataset (509 images) which is not overlapped with any test images in all benchmarks. All the training images were relabeled with accurate text-line bounding boxes.

Implementation details. We use a pre-trained VGG16 model [29] for ImageNet classification to initialize the base network of our text detection model. The weights of new layers for RPN and LocNet regression module 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 [24]. The learning rate is 0.001 for the first 30K iterations and 0.0001 for the next 30K. The momentum is 0.9 and weight decay is 0.0005. Our experiments are conducted on Caffe framework [38]. 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}. In the testing phase, we select the top-300 text-line proposals generated from RPN for Fast R-CNN and apply single-scale testing ($S = 500$) with a single model.

C. Comparison between LocNet Based Localization Module and Bounding Box Rregression Module

We train two Faster R-CNN based models for text detection. The first one applies bounding box regression [24] as the localization module of Fast R-CNN (named BBox reg. model), while the second one adopts the proposed LocNet based localization module (named LocNet model). For fair comparison, these two models are trained with the same hyper-parameters. As stated in [27], 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 [27] 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, respectively. Results are listed in Table 1. It is observed that when applying the default value of $t_r = 0.8$ and $t_p = 0.4$ as suggested in [27], LocNet model outperforms BBox reg. model by 2.01% in recall, 3.48% in precision, 2.76% in F-measure, respectively. Furthermore,

TABLE 1. Comparison between LocNet model and BBox reg. model over various thresholds of the evaluation tool [27] on ICDAR-2011 (R: Recall, P: Precision, F: F-measure).

Model	Thresholds		R (%)	P (%)	F (%)	ΔF (%)
	t_r	t_p				
BBox reg.	0.8	0.4	86.80	84.97	85.87	-
	0.9	0.4	74.18	73.69	73.93	-
	1.0	0.4	30.53	31.14	30.83	-
	0.8	0.5	84.44	82.95	83.69	-
	0.8	0.6	74.26	76.91	75.65	-
LocNet	0.8	0.4	88.81	88.45	**88.63**	**+2.76**
	0.9	0.4	80.01	80.70	**80.38**	**+6.45**
	1.0	0.4	41.97	39.72	**40.81**	**+9.98**
	0.8	0.5	84.61	86.06	**85.33**	**+1.64**
	0.8	0.6	75.44	80.14	**77.72**	**+2.07**

when 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., +6.45% and +9.98%, respectively. Moreover, when t_p becomes stricter, e.g., $t_p = 0.5$ and $t_r = 0.6$, LocNet model can also achieve better results, i.e., outperforming BBox reg. model by +1.64% and +2.07% 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.

D. 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 2 and Table 3, our approach achieves the best performance on both datasets. On ICDAR-2013 dataset, our approach achieves the best 86.70%, 93.00% and 89.74% in recall, precision and F-measure, respectively, outperforming the other methods by a notable margin, though many previous approaches have been well-tuned on ICDAR-2013 task. On ICDAR-2011 dataset, our approach outperforms the closest TextBoxes [17] remarkably by 2.63% improvement on F-measure. It is worth noting that, although the method in [8] also achieves superior results on these two datasets, it is not comparable because that approach needs millions of training samples to achieve good performance, while our approach is only trained with thousands of images. Our detection results on several challenging images in these two datasets are shown in Fig. 5. The results demonstrate the effectiveness and robustness of our proposed text detector, which is able to 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.

TABLE 2. Comparison with prior arts on ICDAR-2013 (%).

Method	Recall	Precision	F-measure
Proposed	**86.70**	**93.00**	**89.74**
CTPN [16]	82.98	92.98	87.69
Zhu et al. [39]	81.02	93.39	86.77
TextBoxes [17]	83.00	89.00	85.89
Zhong et al. [18]	83.00	87.00	85.00
Gupta et al. [15]	75.50	92.00	83.00
TextFlow [40]	75.89	85.15	80.25
1st ICDAR'2013 [2]	69.28	88.80	77.83

TABLE 3. Comparison with prior arts on ICDAR-2011 (%).

Method	Recall	Precision	F-measure
Proposed	**88.81**	88.45	**88.63**
TextBoxes [17]	82.00	89.00	86.00
CTPN [16]	79.00	89.00	84.00
Zhong et al. [18]	81.00	85.00	83.00
Gupta et al. [15]	74.80	**91.50**	82.30
TextFlow [40]	76.17	86.24	80.89
Zhang et al. [41]	84.00	76.00	80.00
1st ICDAR' 2011 [1]	62.47	82.98	71.28

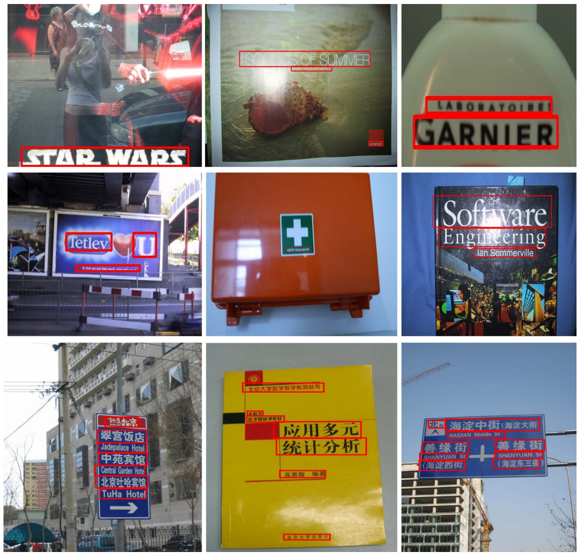

Fig. 5. Example results of our approach. 1st and 2nd row: ICDAR-2011 and

TABLE 4. Comparison with prior arts on MULTILINGUAL (%).

Method	Recall	Precision	F-measure
Proposed	**84.23**	82.45	**83.33**
CPTN [16]	80.00	84.00	82.00
TextFlow [40]	78.40	**84.70**	81.40
Yin et al. [6]	82.60	68.50	74.60
Pan et al. [28]	64.50	65.90	65.50

E. 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 4. Our approach achieves the best recall and F-measure, even though some other approaches have used the provided training set. As shown in Fig. 5, the robust detection results in the multilingual text images demonstrate that our approach generalizes well in such scenario and is insensitive to languages.

V. Conclusion

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. Comprehensive evaluations and comparisons are made on three benchmark datasets on which our proposed approach achieves superior performance. Our approach can not only detect robustly text-lines under various challenging conditions with accurate bounding box localization, but also generalize well to different languages.

VI. References

[1] A. Shahab, F. Shafait, 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 Tran. 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] K. Wang and S. Belongie, "Word spotting in the wild," in *ECCV*, 2010, pp. 591-604.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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*, 2016, pp. 4164-4167.

[18] 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. 1281-1212.

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

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431-3440.

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

[22] 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.

[23] 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.

[24] 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.

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

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

[27] 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.

[28] 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.

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

[30] R. B. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440-1448.

[31] 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.

[32] S. Gidaris and N. Komodakis, "Attend refine repeat: Active box proposal generation via in-out localization," in *arXiv preprint arXiv:1606.04446*, 2016.

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

[34] 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.

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

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

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