2017 14th IAPR International Conference on Document Analysis and Recognition

Text Detection by Faster R-CNN with Multiple Region Proposal Networks

Yoshito Nagaoka, Tomo Miyazaki, Yoshihiro Sugaya, Shinichiro Omachi
Department of Communications Engineering, Graduate School of Engineering, Tohoku University
Sendai, Japan
Email: { nagayosi, tomo, sugaya, machi }@iic.ecei.tohoku.ac.jp

***Abstract*—We propose an end-to-end consistently trainable text detection method based on the Faster R-CNN. The original Faster R-CNN is an end-to-end CNN for fast and accurate object detection. By considering the characteristics of texts, a novel architecture that make use of its ability on object detection is proposed. Although the original Faster R-CNN generates region of interests (RoIs) by a region proposal network (RPN) using the feature map of the last convolutional layer, the proposed method generates RoIs by multiple RPNs using the feature maps of multiple convolutional layers. This method uses multiresolution feature maps to detect texts of various sizes simultaneously. To aggregate the RoIs, we introduce RoI-merge layer, and this layer enables to select valid RoIs from multiple RPNs effectively. In addition, a training strategy is proposed for realizing end-to-end training and making each RPN be specialized in text region size. Experimental results using ICDAR2013/2015 RRC test dataset show that the proposed Multi-RPN method improved detection scores and kept almost the same detection speed as compared to the original Faster R-CNN and recent methods.**

Keywords—Text detection, Faster R-CNN, Region Proposal Network

I. Introduction

Text in image can be adapted to many applications like translation and mobile visual search system. In order to make use of these text information, researches on text detection and recognition have been conducted for decades [1]. The accuracy of individual character recognition and word recognition has drastically improved by the use of convolutional neural network (CNN) [2]. However, because texts in images have various color, font and size, it is difficult to detect all kinds of text.

Recent trend in this research field is to realize an end-to-end text recognition system by integrating text detection and text recognition techniques that had been studied separately. Wang et al. developed a two-stage pipeline method [3]. Characters in a scene image is detected by a sliding-window method with Random Ferns [4] and non-maximum suppression. Then the detected characters are recognized by a leading OCR engine. Wang et al. used CNNs for both text detection and text recognition [5]. The sliding window method is used to detect a set of candidate characters, and beam search is used to obtain the result. Milyaev et al. evaluated the performance of multiple binarization method for end-to-end text understanding [6]. They showed a pipeline consisting of an appropriate binarization method and off-the-shelf OCR module achieved a good result. Opitz et al. [7] detected texts with a sliding window classifier and Maximally Stable Extremal Regions [8]. Text recognition is performed by a deep CNN. Jaderberg et al. [9] proposed a method for using Edge-Boxes [10] for text detection. A deep CNN is used for recognition. Neumann and Matas presented a real-time text localization and recognition method [11]. Character candidates are detected as Extremal Regions and grouped into text lines. Then an OCR trained on synthetic fonts labels character regions and most probable sequence is selected. Shi et al. proposed an end-to-end trainable neural network for sequence image recognition by combining CNN and the recurrent neural network [12]. The effectiveness is evaluated with cropped scene text images. Ren et al. designed a Chinese scene text information extraction system [13]. Image patches are extracted by a sliding-window method and recognized by text structure features.

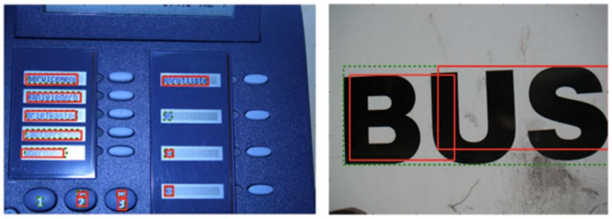
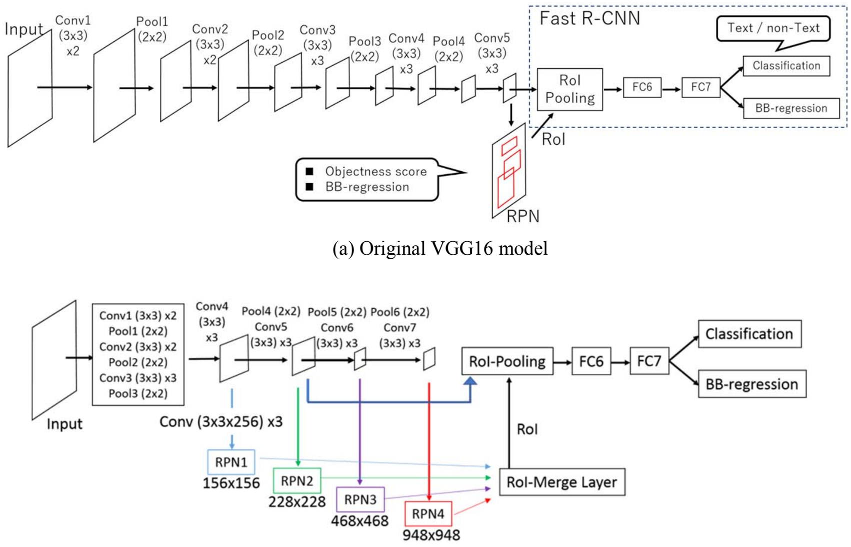


Fig. 1. Detection by Faster R-CNN (Green dashed line is ground-truth, red line is detect)

However, all of these methods combine almost independent methods for text detection from scene images and recognition of the detected text. In order to utilize the training ability of the CNN, it is desirable that a system is not only end-to-end but also consistently trainable from input scene image to output text. As a method for object detection, Faster R-CNN [14] had been proposed. The Faster R-CNN is an end-to-end CNN that enables consistent training for region proposal and classification. It is applied in various detection tasks and achieves good results, and regarded as one of the state of the art methods in terms of detection accuracy and speed.

In this paper, we propose an end-to-end consistently trainable CNN model for text detection inspired by the achievement of the Faster R-CNN. A big difference between general objects and texts are diversity, size and aspect ratio. In addition, since the original Faster R-CNN generates region of interests (RoIs) by one RPN using the constant resolution feature map of the last convolutional layer, it cannot detect texts of various sizes simultaneously (see Fig 1). To address these issues,

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.343



(a) Original VGG16 model

(b) Multi-RPN model

Fig. 2. Overall architectures of the original VGG16 model and Multi-RPN model. The number below RPN indicates the size of the receptive field.

we propose a novel method which generates RoIs by multiple RPNs, and this is called Multi-RPN model in this paper. This method enables multiresolution feature extraction and supplements the use of various sizes of windows for region proposal. To serve the purpose of aggregating and selecting the RoIs efficiently, we introduce RoI-merge layer. In addition, a training strategy for Multi-RPN model is proposed for realizing end-to-end training. We conducted experimental validation to show the effect of the proposed method compared to the original Faster R-CNN and some recent text detection methods.

II. Related Work

The proposed method is designed based on the Faster R-CNN. In this section, we address some related work to the Faster R-CNN.

An innovative method for object detection using CNN is the Region based CNN (R-CNN) [15]. Given an input image, multiple regions are detected as RoIs by a region proposal method such as the selective search [16]. The feature for each RoI is calculated by CNN and classified. The drawback of this method is the huge computational cost because classification is performed for every RoI. An improved method of the R-CNN is the Fast R-CNN [17]. It calculates feature map of the input image and get the feature of each RoI by pooling. For these methods, region proposal is a procedure separated from the CNN training, which requires huge computational cost.

The Faster R-CNN [14] can generate RoIs by itself with a region proposal network (RPN). The architecture of the Faster R-CNN is displayed in Fig. 2 (a). As a deep CNN, VGG16 model [18] is used. The RPN uses the feature map of the last convolutional layer of the CNN and outputs RoIs. When detecting RoIs, anchors are used as molds of RoIs. Anchors are rectangles with various scale and aspect ratio that enables to generate various type of RoIs in the feature map. The size of each RoI is adjusted by a process called RoI pooling. For each RoI, objectness score and bounding box are calculated and fed into two sibling layers for classification and bounding box regression (BB-regression). Combining the outputs of these layers, detected objects are put out with class labels. Since the network after RoI pooling is the same as the Fast R-CNN, we refer this part as Fast R-CNN in this paper.

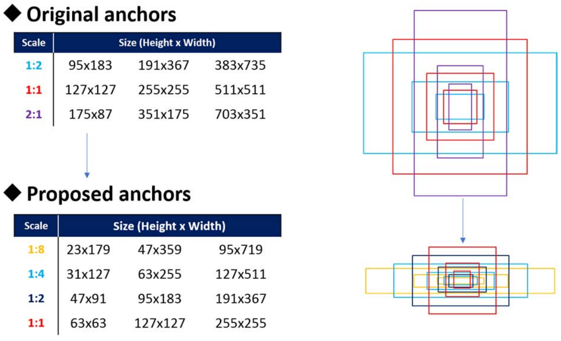
III. Proposed Method

The proposed architecture includes multiple RPNs and RoI-merge layer in addition to the original Faster R-CNN as shown in Fig. 2 (b).

A. Multi-RPN

The original Faster R-CNN (Fig. 2 (a)) has only one RPN that uses a feature map of the last convolutional layer (conv5-3 layer in the VGG16 model). We call this the original RPN. The receptive field size in input image by the original RPN is 228×228, which may be sufficient to get features of typical objects. However, texts in images have various sizes and scales, and this receptive field is sometimes too large or too small to detect texts. If the receptive field is too large for a small text, something surrounding the text may be regarded as noise. If it is too small for a large text, RPN cannot generate RoIs because of the insufficient feature.

Therefore, we introduce multiple RPNs for using various feature maps. We call this architecture Multi-RPN. The overall architecture of Multi-RPN is shown in Fig. 2 (b). The receptive field of RPN after conv4-3 layer is 156×156, which is smaller than the original RPN. On the other hand, the one after conv6-3 is 468×468 and after conv7-3 is 948×948, that are larger than the original RPN. Input image size is between 600 and 900,



◆ Original anchors

Scale	Size (Height x Width)		
1:2	95x183	191x367	383x735
1:1	127x127	255x255	511x511
2:1	175x87	351x175	703x351

◆ Proposed anchors

Scale	Size (Height x Width)		
1:8	23x179	47x359	95x719
1:4	31x127	63x255	127x511
1:2	47x91	95x183	191x367
1:1	63x63	127x127	255x255

Fig. 3. Anchor

so RPN after conv7-3 can utilize overall information of the input image. To utilize these receptive field, we set special anchors to each RPN and train RPNs with special strategy. Those are described in subsection C and D in detail. This strategy enables multiresolution feature extraction, which is suitable for various sizes of anchors. Multi-RPN generates various RoIs and is more effective than the approach using only one RPN.

B. RoI-Merge Layer

Since each RPN outputs separated RoI arrays, we need the layer combining these RoI arrays into one array in complementation. For this purpose, RoI-merge layer is introduced. This layer receives separated RoI arrays and outputs one array. To avoid duplicate RoIs and RoIs with low text likelihood scores (textness), we implement non-maximum suppression (NMS) in condition that intersection over union (IoU) overlap is 0.7. After NMS, we select 100 RoIs with higher textness, so we use 100 RoIs in following process as a result. From above reason, this layer needs only hyper parameters to control the number of RoIs, but does not have any parameters which need to be trained.

C. Anchors for Text Detection

Anchor is a rectangle representing a region for object detection. The original Faster R-CNN consists of nine anchors shown at the top of Fig. 3. While the purpose of the original Faster R-CNN is object detection, our purpose in this work is detecting text. Typical text area is horizontally long and different from that of the typical object. Therefore, we set horizontally long anchors (see bottom of Fig. 3) suitable for text detection.

To detect various sizes of texts, we set small anchors (columns 1 and 2 of the table of the proposed anchors in Fig. 3) in RPN1, all anchors in RPN2 and large anchors (column 2 and 3 of the table of the proposed anchors) in RPN3 and RPN4. RPN1 is specialized in detecting small text, while RPN3 and RPN4 are specialized in large text.

D. Training Strategy

When training the original VGG16 model, we use a training strategy that applies RPN loss function and Fast R-CNN loss function in each iteration.

On the other hand, Multi-RPN model structure is largely different from the original model in the point of the total number of RPNs. In this architecture, we apply RPN loss function for every RPN and Fast R-CNN loss function in one iteration.

Moreover, because RPNs have anchors of various sizes, we assign training data corresponding to each RPN. To train RPN1, we used small annotations of rectangles, and we used large ones to train RPN3 and RPN4. In the experiment, a small annotation composed of rectangles of which larger side is smaller than 150. A large annotation composed of rectangles of which larger side is larger than 250.

These thresholds were determined by the size of the receptive field of each RPN.

E. Network Architecture

Our network is based on the VGG16 model, so hyper parameters of all layers from conv1 to conv5 are the same as the ones of the VGG16 model. In the Multi-RPN model, we introduced three additional RPNs, so our proposed model has totally four RPNs. In RPN1, we add three convolutional layers (kernel size: 3×3, kernel number: 256) to construct deep architecture. Strides of the last layer is two to save computing cost following RoIs computation, and strides of other are one. In RPN3 and RPN4, three convolutional layers (kernel size: 3×3, kernel number: 512) and max-pooling layer (kernel size: 2×2, stride: 2) are added two times after conv5-3. The input of RoI-Pooling layer was conv5-3 as with the original Faster R-CNN.

F. Post processing

The proposed method produces several detection regions. To improve detection accuracy, we implement NMS (threshold was 0.3) and discard detection regions which textness were less than 0.8. This process suppresses false detection regions.

IV. Experiments

A. Datasets and Evaluation Metric

We used totally 7105 natural scene images for training. These consisted of largely five datasets (229 images in ICDAR 2013/2015 Robust Reading Competition (RRC) challenge2 [19] training dataset, 100 images in Street View Text (SVT) [20] training set, 351 images in KAIST Scene Text Database [21], 1,000 images in ICDAR 2013/2015 RRC challenge4 [19] training dataset and 5,425 images in ICDAR2017 RRC MLT [23] training dataset). Regarding SVT and KAIST, we annotated word level ground-truth labels to only Latin script by ourselves. In other images, we used annotations published by official.

For evaluation, we used ICDAR 2013/2015 RRC challenge2 test dataset (233 images). We compared our method to other methods using DetEval [22] as the evaluation metric.

B. Experimental Environmet

Network weights were initialized with the Faster R-CNN imagenet pre-trained model. RPN layers other than the original one were initialized as with the original RPN and added convolutional layers were initialized with the same as VGG16.

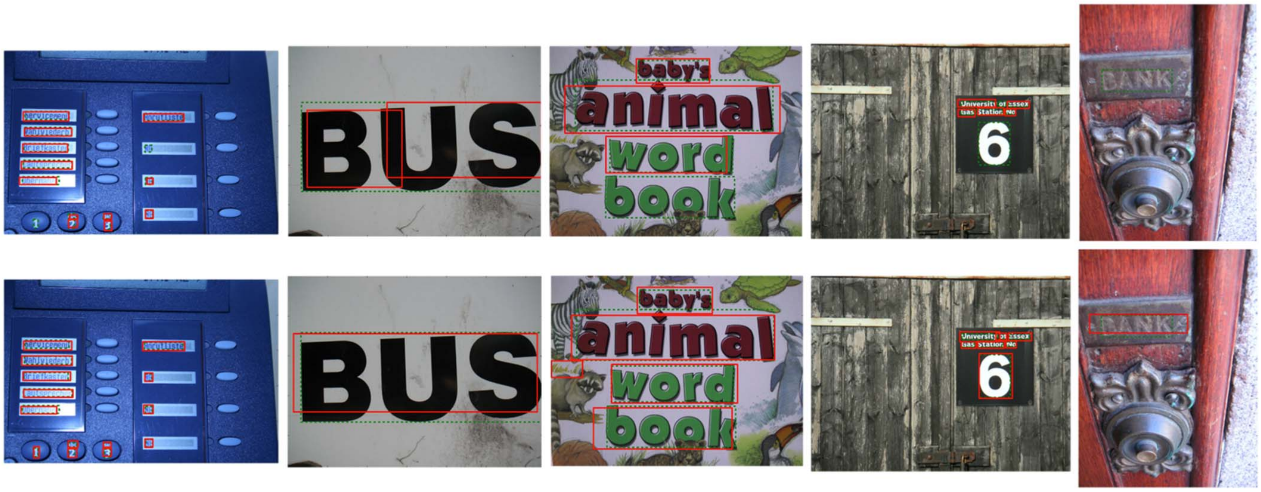


Fig. 4. Some detection results of the original anchor model (upper row) and the proposed Multi-RPN model (lower row). Green dased line shows the ground-truth and red line shows the result of detection.

TABLE I. RESULTS

From left, Recall (%), Precision (%), F-measure (%) and detection speed (second/image)

Method	*Recall*	*Precision*	*F-measure*	*Speed*
***Tian et al.* [24]**	***76***	***85***	***80***	**-**
***He et al.* [25]**	***81***	***92***	***86***	***0.9***
***Liao et al.* [26]**	***83***	***89***	***86***	***0.73***
***Zhong et al.* [27]**	***83***	***87***	***85***	***1.7***
***Tian et al.* [28]**	***83***	***93***	***88***	***0.14***
***Shi et al.* [29]**	***87.7***	***83.0***	***85.3***	***0.05***
Anchor	***79.11***	***88.76***	***83.65***	***0.11***
Multi-RPN	***81.10***	***90.24***	***85.42***	***0.12***

When training and testing, we used NVIDIA TITAN X (Pascal). For the original VGG16 model, we iterated 80k, meanwhile for training Multi-RPN models, we iterated 100k.

C. Experimental Results

Experimental result is shown in Table I. To compare our method, we used the original Faster R-CNN with VGG16 model by changing the anchors as described in section III-C (called Anchor) . In addition, the results by the recent text detection methods are shown. Regarding significant figure in Table I, results of recent works ([24] - [29]) are quoted from original papers as they are, but our results values are according to ICDAR2013 official site form. The Anchor achieved F-measure 83.65%, but the proposed Multi-RPN method improved Recall, Precision and F-measure by about 2.0, 1.5 and 1.8 percent points, respectively. Particularly, precision of the proposed method achieved 90%, and F-measure was comparable to the state-of-the-art methods [29]. Moreover, the proposed method was comparable with other methods keeping fast detection speed. Multi-RPN is end-to-end detection system and simple architecture, so our method is favorable compared to other methods.

Some detection results are shown in Fig. 4. The proposed Multi-RPN model could detect text more than the Anchor model, particularly very small size text (column 1) and large size text (column 2) which were not detected by the Anchor model. This is because each RPN specialized in various size detection and training strategy.

Moreover, Multi-RPN could detect other text regardless of the size. This is because Multi-RPN model was trained with many RoIs by special training strategy and was optimized more efficiently than using only one RPN.

D. Discussion

1. RoI of Each RPN

The proposed Multi-RPN method improved the detection accuracy. RoIs generated by the RPN are visualized in Fig. 5. The anchor model could detect text almost the same as the Multi-RPN model, however, RoIs detected by the anchor model were classified to low textness by the RPN. On the other hand, the Multi-RPN generated more valid RoIs with high textness. This is because Multi-RPN collects RoIs from four RPNs, and selects many RoIs with high textness efficiently.

RoIs generated by from RPN1 to RPN4 are shown in Fig. 6. RPN1 generated more small RoIs with high textness. In contrast, RPN3 and RPN4 generated large RoIs with high textness. These results were obtained by the special training strategy. Although RPN1 detected many small RoIs toward to large text (bottom row of Fig. 6) and RPN3 and RPN4 detected RoIs toward small text (top row of Fig. 6), those had low textness and were discarded in the following RoI-marge process. From above, the proposed method is more effective than the original model regarding text size, and our proposed training strategy made effective classifier.

2. Failed Detection

Some results failed by the Multi-RPN are displayed in Fig. 7. Those were occurred by bad situation and diversity of text. In the left figure of Fig. 7, since fence in front of text makes detection difficult, RoIs could not detect “CELCON”. On the other hand, the fonts in the right figure of Fig. 7 are different from typical fonts. RoIs enclosed “MED”, but those were discarded by the following classification process. From above,

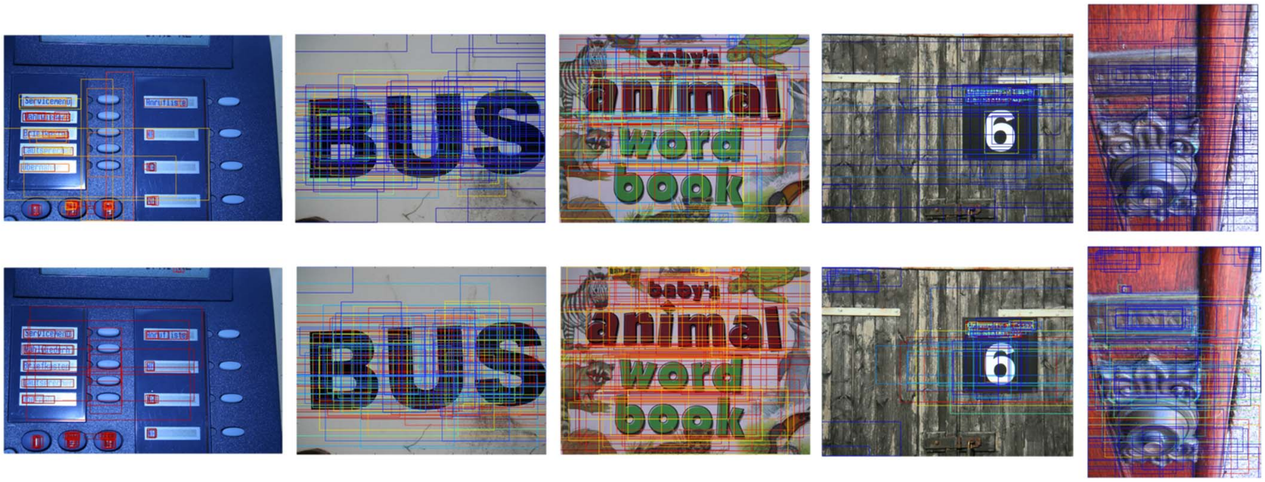


Fig. 5. The RoIs results of original anchor model (upper row) and Multi-RPN model (lower row). Red line indicates high textness and blue line is low textness score.

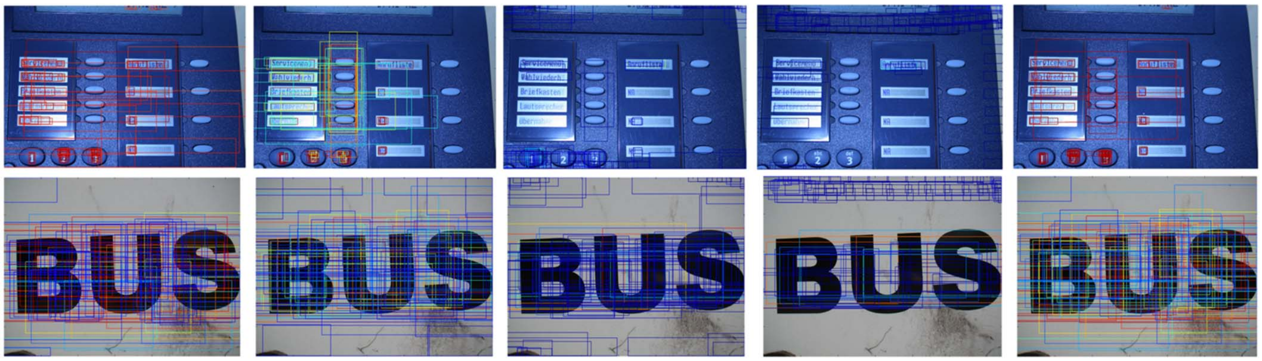


Fig. 6. Visualization of 100 RoIs generated by each RPN (From left to right: RPN1, RPN2, RPN3, RPN4, and RoI-Merge) . Red line indicates high textness and blue line indicates low textness.

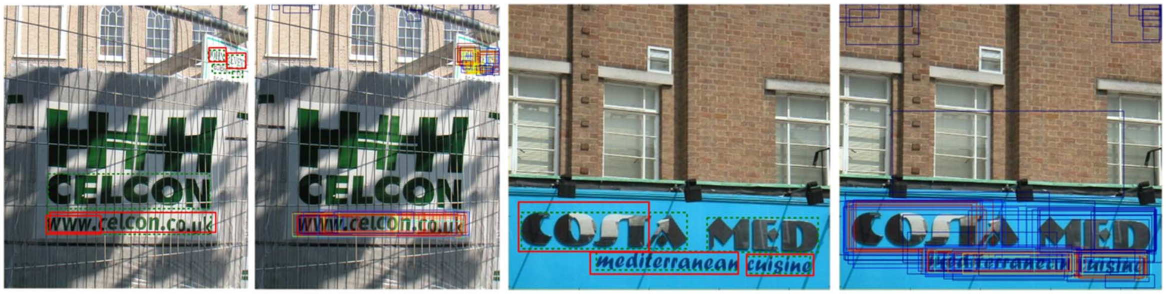


Fig. 7. Results failed by the Multi-RPN (columns 1 and 3 show detection results, and columns 2 and 4 show RoIs).

we need to make the method robust to photography environment and diversity of text.

V. Conclusion

We proposed the Multi-RPN Faster R-CNN model for text detection. With multiple RPNs, multiresolution feature map can be utilized. Also, it was clarified that the proposed training strategy was more effective than the conventional strategy of the Faster R-CNN. Multi-RPN model is end-to-end trainable and simple architecture as with Faster R-CNN. Therefore, detection speed is almost as fast as the original Faster R-CNN and it can detect text consistently. By these, the proposed model improved detection scores dramatically. However, multi-RPN cannot detect all kinds of text and is limited to horizontal text. Recently, detecting multi-oriented text [30] is studied. We will improve the detection algorithm so as to detect more variety of texts and multi-oriented text in natural scene images in the future.

REFERENCES

[1] Q. Ye and D. Doermann, "Text detection and recognition in imagery: a survey," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 37, no. 7, pp. 1480-1500, 2015.

[2] X. Liu, T. Kawanishi, X. Wu, and K. Kashino, "Scene text recognition with high performance CNN classifier and efficient word inference," in Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1322-1326, 2016.

[3] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in Proceedins of the 2011 International Conference on Computer Vision, pp. 1457-1464, 2011.

[4] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pp. 448-461, 2010.

[5] T. Wang, D. J. Wu, and A. Coates, "End-to-end text recognition with convolutional neural networks," in Proceedings of the 21st International Conference on Pattern Recognition, 2012.

[6] S. Milyaev, O. Barinova, T. Novikova, P. Kohli, and V. Lempitsky, "Image binarization for end-to-end text understanding in natural images, in Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 128-132, 2013.

[7] M. Opitz, M. Diem, S. Fiel, F. Kleber, and R. Sablatnig, "End-to-End Text Recognition Using Local Ternary Patterns, MSER and Deep Convolutional Nets," in Proceedings of the 11th IAPR International Workshop on Document Analysis Systems, pp.186-190, 2014.

[8] J. Matas, O Chuma, M. Urbana, and T. Pajdlaa, "Robust wide-baseline stereo from maximally stable extremal regions," Image and Vision Computing, vol. 22, no. 10, pp. 761-767, 2004.

[9] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," International Journal of Computer Vision, vol. 116, no. 1, pp. 1-20, 2015.

[10] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in Proceedins of the *European Conference on Computer Vision (ECCV)*, 2014.

[11] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 38, no. 9, pp. 1872-1885, 2016.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, 2016.

[13] X. Ren, Y. Zhou, Z. Huang, J. Sun, X. Yang, K. Chen, "A novel text structure feature extractor for Chinese scene text detection and recognition," IEEE Access, vol. 5, pp. 3193-3204, 2017.

[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedins of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[16] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," International Journal of Computer Vision, vol. 104, no. 2, pp. 154-171, 2013.

[17] R. Girshick, "Fast R-CNN," in Proceedins of the IEEE International Conference on Computer Vision (ICCV), 2015.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedins of the *International Conference on Learning Representations (ICLR)*, 2015.

[19] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, E. Valveny, "ICDAR 2015 competition on Robust Reading," in Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1156-1160, 2015.

[20] K. Wang and S. Belongie, "Word spotting in the wild," in Proceedins of the *European Conference on Computer Vision* (ECCV), 2010.

[21] http://www.iapr-tc11.org/mediawiki/index.php?title=KAIST_Scene_Text_Database

[22] C. Wolf and J.-M. Jolion, "Object count / area graphs for the evaluation of object detection and segmentation slgorithms," International Journal of Document Analysis and Recognition, vol. 8, no. 4, pp. 280-296, 2006.

[23] http://rrc.cvc.uab.es/?ch=8

[24] S. Tian, Shangxuan, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan, "Text flow: A unified text detection system in natural scene images," *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[25] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep Direct Regression for Multi-Oriented Scene Text Detection," *arXiv:1703.08289* (2017).

[26] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A Fast Text Detector with a Single Deep Neural Network," *AAAI*. 2017.

[27] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "Deeptext: A unified framework for text proposal generation and text detection in natural images," *arXiv:1605.07314* (2016).

[28] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," *European Conference on Computer Vision*. Springer International Publishing, 2016.

[29] B. Shi, X. Bai, and S. Belongie. "Detecting Oriented Text in Natural Images by Linking Segments," *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[30] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," *IEEE Conference on Computer Vision and Pattern Recognition* (2017).