DeepText: A Unified Framework for Text Proposal Generation and Text Detection in Natural Images

Zhuoyao Zhong
z.zhuoyao@mail.scut.sdu.cn
Lianwen Jin
lianwen.jin@gmail.com
Shuye Zhang
shuye.cheung@gmail.com
Ziyong Feng
feng.ziyong@mail.scut.edu.cn

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

Abstract

In this paper, we develop a novel unified framework called DeepText for text region proposal generation and text detection in natural images via a fully convolutional neural network (CNN). First, we propose the inception region proposal network (Inception-RPN) and design a set of text characteristic prior bounding boxes to achieve high word recall with only hundred level candidate proposals. Next, we present a powerful text detection network that embeds ambiguous text category (ATC) information and multi-level region-of-interest pooling (MLRP) for text and non-text classification and accurate localization. Finally, we apply an iterative bounding box voting scheme to pursue high recall in a complementary manner and introduce a filtering algorithm to retain the most suitable bounding box, while removing redundant inner and outer boxes for each text instance. Our approach achieves an F-measure of **0.83** and **0.85** on the ICDAR 2011 and 2013 robust text detection benchmarks, outperforming previous state-of-the-art results.

1 Introduction

Text detection is a procedure that determines whether text is present in natural images and, if it is, where each text instance is located. Text in images provides rich and precise high-level semantic information, which is important for numerous potential applications such as scene understanding, image and video retrieval, and content-based recommendation systems. Consequently, text detection in natural scenes has attracted considerable attention in the computer vision and image understanding community [5, 10, 11, 12, 14, 15, 18, 23, 25, 27, 29, 31]. However, text detection in the wild is still a challenging and unsolved problem because of the following factors. First, a text image background is very complex and some region components such as signs, bricks, and grass are difficult to distinguish from text. Second, scene text can be diverse and usually exits in various colors, fonts, orientations, languages, and scales in natural images. Furthermore, there are highly confounding factors,

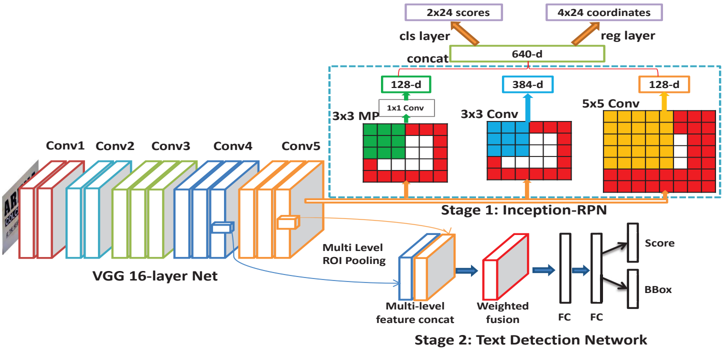

Figure 1: Pipeline architecture of DeepText. Our approach takes a natural image as input, generates hundreds of word region proposals via Inception-RPN (Stage 1), and then scores and refines each word proposal using the text detection network (Stage 2).

such as non-uniform illumination, strong exposure, low contrast, blurring, low resolution, and occlusion, which pose hard challenges for the text detection task.

In the last few decades, sliding window-based and connected component-based methods have become mainstream approaches to the text detection problem. Sliding window-based methods [11, 27] use different ratios and scales of sliding windows to search for the presence of possible text positions in pyramid images, incurring a high computational cost. Connected component based methods, represented by maximally stable extremal regions (MSERs) [10, 18, 23, 29] and the stroke width transform (SWT) [5], extract character candidates and group them into word or text lines. In particular, previous approaches applying MSERs as the basic representation have achieved promising performance in the ICDAR 2011 and 2013 robust text detection competitions [14, 15]. However, MSERs focuses on low-level pixel operations and mainly accesses local character component information, which leads to poor performance in some challenging situations, such as multiple connected characters, segmented stroke characters, and non-uniform illumination, as mentioned in [31]. Further, this bottom-up approach gives rise to sequential error accumulation in the total text detection pipeline, as stated in [25].

Rather than extract character candidates, Jaderberg *et al.* [12] applied complementary region proposal methods called edge boxes (EB) [33] and aggregate channel feature (ACF) [4] to perform word detection and acquired a high word recall with tens of thousands of word region proposals. They then employed HOG features and a random forest classifier to remove non-text region proposals and hence improve precision. Bounding box regression was also used for more accurate localization. Finally, using a large pre-trained convolutional neural network (CNN) to recognize the detected word-cropped images, they achieved superior text spotting and text-based image retrieval performance on several standard benchmarks..

Actually, the region proposal generation step in the generic object detection pipeline has attracted much interest. In recent studies, object detection models based on region proposal algorithms to hypothesize class-specific or class-agnostic object locations have achieved state-of-the-art detection performance [6, 7, 8, 9]. However, standard region proposal algorithms such as selective search (SS) [3], MCG [1], EB [33], generate an extremely large number of region proposals. This leads to high recall, but burdens the follow-up classification and regression models and is also relatively time-consuming. In order to address these issues, Ren *et al.* [21] proposed region proposal networks (RPNs), which computed region proposals with a deep fully CNN. They generated fewer region proposals, but achieved a promising recall rate under different overlap thresholds. Moreover, RPN and Fast R-CNN can be combined into a joint network and trained to share convolutional features. Owing to the above innovation, this approach achieved better object detection accuracy in less time than Fast R-CNN with SS [7] on PASCAL VOC 2007 and 2012.

In this paper, inspired by [21], our motivation is to design a unified framework for text characteristic region proposal generation and text detection in natural images. In order to avoid the sequential error accumulation of bottom-up character candidate extraction strategies, we focus on word proposal generation. In contrast to previous region proposal methods that generate thousands of word region proposals, we are motivated to reduce this number to hundreds while maintaining a high word recall. To accomplish this, we propose the novel inception RPN (Inception-RPN) and design a set of text characteristic prior bounding boxes to hunt high-quality word region proposals. Subsequently, we present a powerful text detection network by incorporating extra ambiguous text category (ATC) information and multi-level region of interest (ROI) pooling into the optimization process. Finally, by means of some heuristic processing, including an iterative bounding box voting scheme and filtering algorithm to remove redundant boxes for each text instance, we achieve our high-performance text detection system, called DeepText. An overview of DeepText is shown in Fig. 1. Our contributions can be summarized by the following points.

(1) We propose inception-RPN, which applies multi-scale sliding windows over the top of convolutional feature maps and associates a set of text characteristic prior bounding boxes with each sliding position to generate word region proposals. The multi-scale sliding-window feature can retain local information as well as contextual information at the corresponding position, which helps to filter out non-text prior bounding boxes. Our Inception-RPN enables achieving a high recall with only hundreds of word region proposals.

(2) We introduce the additional ATC information and multi-level ROI pooling (MLRP) into the text detection network, which helps it to learn more discriminative information for distinguishing text from complex backgrounds.

(3) In order to make better use of intermediate models in the overall training process, we develop an iterative bounding box voting scheme, which obtains high word recall in a complementary manner. Besides, based on empirical observation, multiple inner boxes or outer boxes may simultaneously exist for one text instance. To tackle this problem, we use a filtering algorithm to keep the most suitable bounding box and remove the remainders.

(4) Our approach achieves an F-measure of 0.83 and 0.85 on the ICDAR 2011 and 2013 robust text detection benchmarks, respectively, outperforming the previous state-of-the-art results.

The remainder of this paper is set out as follows. The proposed methodology is described in detail in Section 2. Section 3 presents our experimental results and analysis. Finally, the conclusion is given in Section 4.

2 Methodology

2.1 Text region proposal generation

Our inception-RPN method resembles the notion of RPN proposed in [21], which takes a natural scene image and set of ground-truth bounding boxes that mark text regions as input and generates a manageable number of candidate word region proposals. To search for word region proposals, we slide an inception network over the top of convolutional feature maps (Conv5_3) in the VGG16 model [22] and associate a set of text characteristic prior bounding boxes with each sliding position. The details are as follows.

Text characteristic prior bounding box design. Our prior bounding boxes are similar to the anchor boxes defined in RPN. Taking text characteristics into consideration, for most word or text line instances, width is usually greater than height; in other words, their aspect ratios are usually less than one. Furthermore, most text regions are small in natural images. Therefore, we empirically design four scales (32, 48, 64, and 80) and six aspect ratios (0.2, 0.5, 0.8, 1.0, 1.2, and 1.5), for a total of $k = 24$ prior bounding boxes at each sliding position, which is suitable for text properties as well as incident situations. In the learning stage, we assign a positive label to a prior box that has an intersection over union (IoU) overlap greater than 0.5 with a ground-truth bounding box, while assigning a background label to a prior box with an IoU overlap less than 0.3 with any ground-truths.

Inception-RPN. We design Inception-RPN, inspired by the idea of the inception module in GoogLeNet [24], which used flexible convolutional or pooling kernel filter sizes with a layer-by-layer structure to achieve local feature extraction. This method has proved to be robust for large-scale image classification. Our designed inception network consists of a 3×3 convolution, 5×5 convolution and 3×3 max pooling layers, which is fully connected to the corresponding spatial receptive fields of the input Conv5_3 feature maps. That is, we apply 3×3 convolution, 5×5 convolution and 3×3 max pooling to extract local featire representation over Conv5_3 feature maps at each sliding position simultaneously. In addition, 1×1 convolution is employed on the top of 3×3 max pooling layer for dimension reduction. We then concatenate each part feature along the channel axis and a 640-d concatenated feature vector is fed into two sibling output layers: a classification layer that predicts textness score of the region and a regression layer that refines the text region location for each kind of prior bounding box at this sliding position. An illustration of Inception-RPN is shown in the top part of Fig. 1. Inception-RPN has the following advantages: (1) the multi-scale sliding-window feature can retain local information as well as contextual information thanks to its center restricted alignment at each sliding position, which helps to classify text and non-text prior bounding boxes, (2) the coexistence of convolution and pooling is effective for more abstract representative feature extraction, as addressed in [24], and (3) experiments shows that Inception-RPN substantially improves word recall at different IoU thresholds with the same number of word region proposals.

Note that for a Conv5_3 feature map of size $m \times n$, Inception-RPN generates $m \times n \times 24$ prior bounding boxes as candidate word region proposals, some of which are redundant and highly overlap with others. Therefore, after each prior bounding box is scored and refined, we apply non-maximum suppression (NMS) [17] with an IoU overlap threshold of 0.7 to retain the highest textness score bounding box and rapidly suppress the lower scoring boxes in the neighborhood. We next select the top-2000 candidate word region proposals for the text detection network.

2.2 Text Detection

ATC incorporation. As in many previous works (e.g., [21]), a positive label is assigned to a proposal that has an IoU overlap greater than 0.5 with a ground truth bounding box, while a background label is assigned to a proposal that has an IoU overlap in the range $[0.1, 0.5)$ with any ground-truths in the detection network. However, this method of proposal

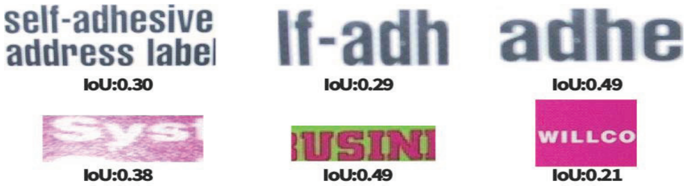

Figure 2: Example word region proposals with an IoU overlap within the interval $[0.2, 0.5)$.

partitioning is unreasonable for text because a proposal with an IoU overlap in the interval $[0.2, 0.5)$ may probably contain partial or extensive text information, as shown in Fig. 2. We note that promiscuous label information may confuse the learning of the text and non-text classification network. To tackle this issue, we refine this proposal label partition strategy to make it suitable for text classification. Hence, we assign a positive text label to a proposal that has an IoU overlap greater than 0.5 with a ground truth, while assigning an additional "ambiguous text" label to a proposal that has an IoU overlap with a ground truth bounding box in the range $[0.2, 0.5)$. In addition, a background label is assigned to any proposal that has an IoU overlap of less than 0.2 with any ground-truths. We assume that more reasonable supervised information incorporation helps the classifier to learn more discriminative feature to distinguish text from complex and diverse backgrounds and filter out non-text region proposals.

MLRP. The ROI pooling procedure performs adaptive max pooling and outputs a max-pooled feature with the original C channels and spatial extents $H \times W$ for each bounding box.Previous state-of-the-art object detection models such as SPP-Net [9], fast-RCNN [7], faster-RCNN [21], all simply apply ROI pooling over the last convolutional layer (Conv5_3) in the VGG16 model. However, to better utilize the multi-level convolutional features and enrich the discriminant information of each bounding box, we perform MLRP over the Conv4_3 as well as Conv5_3 convolutional feature maps of the VGG16 network and obtain two $512 \times H \times W$ pooled feature (both H and W are set to 7 in practice). We apply channel concatenation on each pooled feature and encode concatenated feature with $512 \times 1 \times 1$ convolutional layer. The 1×1 convolutional layer is: (1) combines the multi-level pooled features and learns the fusion weights in the training process and (2) reduces the dimensions to match VGG16's first fully-connected layer. The multi-level weighted fusion feature is then accessed to the follow-up bounding box classification and regression model. An illustration of MLRP is depicted in the bottom half of Fig. 1.

2.3 End-to-end learning optimization

Both Inception-RPN and the text detection network have two sibling output layers: a classification layer and a regression layer. The difference between them is as follows: (1) For Inception-RPN, each kind of prior bounding box should be parameterized independently, so we need to predict all of the $k = 24$ prior bounding boxes simultaneously. The classification layer outputs $2k$ scores textness scores that evaluate the probability of text or non-text for each proposal, while the regression layer outputs $4k$ values that encode the offsets of the refined bounding box. (2) For the text detection network, there are three output scores corresponding to the background, ambiguous text, and positive text categories and four bounding box regression offsets for each positive text proposal (only positive text region proposals access the bounding regression model). We minimize a multi-task loss function, as in [8]:

$$L(p,p^*,t,t^*) = L_{cls}(p,p^*) + \lambda L_{reg}(t,t^*), \tag{1}$$

where classification loss L_{cls} is a softmax loss and p and p^* are given as the predicted and true labels, respectively. Regression loss L_{reg} applies smooth-L_1 loss defined in [7]. Besides, $t = \{t_x,t_y,t_w,t_h\}$ and $t^* = \{t_x^*,t_y^*,t_w^*,t_h^*\}$ stand for predicted and ground-truth bounding box regression offset vector respectively, where t^* is encoded as follows:

$$t_x^* = \frac{(G_x - P_x)}{P_w}, \quad t_y^* = \frac{(G_y - P_y)}{P_h}, \quad t_w^* = log(\frac{G_w}{P_w}), \quad t_h^* = log(\frac{G_h}{P_h}). \tag{2}$$

Here, $P = \{P_x,P_y,P_w,P_h\}$ and $G = \{G_x,G_y,G_w,G_h\}$ denote the center coordinates (x-axis and y-axis), width, and height of proposal P and ground-truth box G, respectively. Furthermore, λ is a loss-balancing parameter, and we set $\lambda = 3$ for Inception-RPN to bias it towards better box locations and $\lambda = 1$ for text detection network.

In contrast to the proposed four-step training strategy to combine RPN and Fast-RCNN in [21], we train our inception-RPN and text detection network in an end-to-end manner via back-propagation and stochastic gradient descent (SGD), as given in Algorithm 1. The shared convolutional layers are initialized by a pre-trained VGG16 model for imageNet classification [22]. All the weights of the new layers are initialized with a zero mean and a standard deviation of 0.01 Gaussian distribution. The base learning rate is 0.001 and is divided by 10 for each 40K mini-batch until convergence. We use a momentum of 0.9 and weight decay of 0.0005. All experiments were conducted in Caffe [13].

2.4 Heuristic processing

Iterative bounding box voting. In order to make better use of the intermediate models in the total training process, we propose an iterative bounding box voting scheme, which can be considered as a simplified version of the method mentioned in [6]. We use $D_c^t = \{B_{i,c}^t, S_{i,c}^t\}_{i=1}^{N_{c,t}}$ to denote the set of $N_{c,t}$ detection candidates generated for specific positive text class c in image I on iteration t, where $B_{i,c}^t$ the i-th bounding box and $S_{i,c}^t$ is the corresponding textness score. For $t = 1,...T$, we merge each iteration detection candidate set together and generate $D_c = \bigcup_{t=1}^T D_c^t$. We then adopt NMS [17] on D_c with an IoU overlap threshold of 0.3 to suppress low-scoring window boxes. In this way, we can obtain a high recall of text instances in a complementary manner and improve the performance of the text detection system.

Filtering. Based on empirical observation, we note that even after NMS [17] processing, multiple inner boxes or outer boxes may still exist for one text instance in the detection candidate set, which may severely harm the precision of the text detection system. To address this problem, we present a filtering algorithm that finds the inner and outer bounding boxes of each text instance in terms of coordinate position, preserves the bounding box with the highest textness score, and removes the others. Thus, we can remove redundant detection boxes and substantially improve precision.

3 Experiments and Analysis

Algorithm 1 End-to-end optimization method for the DeepText training process.

```
Require:
    Set of training images with ground-truths: ${(I_i, \{G_i\})}, ..., (I_N, \{G_N\}))$; learning rate $\eta(t)$ ; samples number $N_* = \{N_b, N_p, N_a, N_n\}$; iteration number $t = 0$.
Ensure:
    Separate network parameters $\mathbf{W^c}, \mathbf{W^p}, \mathbf{W^d}$ for the shared convolutional layer, inception-RPN and text detection network.
1: Randomly select one sample $(I_i, \{G_i\})$ and produce prior bounding boxes classification labels and bounding box regression targets according to $\{G_i\}$;
2: Randomly sample $N_b$ positive and $N_b$ negative prior bounding box from $\{G_i\}$ to compute the loss function in equations (1);
3: Run backward propagation to obtain the gradient with respect to network parameters $\nabla\mathbf{W^c_p}, \nabla\mathbf{W^p}$ and obtain the word proposal set $\{P_i\}$;
4: Adopt NMS with the setting IoU threshold on $\{P_i\}$ and select the top-$k$ ranked proposals to construct $\{D_i\}$ for Step 5;
5: Randomly sample $N_p$ positive text, $N_a$ ambiguous text and $N_n$ background word region proposals from $\{D_i\}$ to compute the loss function in equations (1);
6: Run backward propagation to obtain the gradient with respect to network parameters: $\nabla\mathbf{W^c_d}, \nabla\mathbf{W^d}$;
7: update network parameters: $\mathbf{W^c} = \mathbf{W^c} - \eta(t) \cdot (\nabla\mathbf{W^c_p} + \nabla\mathbf{W^c_d})$, $\mathbf{W^p} = \mathbf{W^p} - \eta(t) \cdot \nabla\mathbf{W^p}$, $\mathbf{W^d} = \mathbf{W^d} - \eta(t) \cdot \nabla\mathbf{W^d}$;
8: $t = t + 1$, if the network has converged,output network parameters $\mathbf{W^c}, \mathbf{W^p}, \mathbf{W^d}$ and end the procedure; otherwise, return the Step 1.
```

3.1 Experiments Data

The ICDAR 2011 dataset includes 229 and 255 images for training and testing, respectively, and there are 229 training and 233 testing images in the ICDAR 2013 dataset. Obviously, the number of training image is constrained to train a reasonable network. In order to increase the diversity and number of training samples, we collect an indoor database that consisted of 1,715 natural images for text detection and recognition from the Flickr website, which is publicly available online[1] and free for research usage. In addition, we manually selected 2,028 images from the COCO-Text benchmark [26]. Ultimately, we collected 4,072 training images in total.

3.2 Evaluation of Inception-RPN

In this section, we compare Inception-RPN with the text characteristic prior bounding boxes (Inception-RPN-TCPB) to state-of-the-art region proposal algorithms, such as SS [3], EB [33] and standard RPN [21]. We compute the word recall rate of word region proposals at different IoU overlap thresholds with ground-truth bounding boxes on the ICDAR 2013 testing set, which includes 1095 word-level annotated text regions. In Fig. 3, we show the results of using N= 100, 300, 500 word region proposals, where the N proposals are the top-N scoring word region proposals ranked in term of these methods. The plots demonstrate that our Inception-RPN-TCPB considerably outperforms standard RPN by 8%-10% and is

[1] https://www.dropbox.com/s/06wfn5ugt5v3djs/SCUT_FORU_DB_Release.rar?dl=0

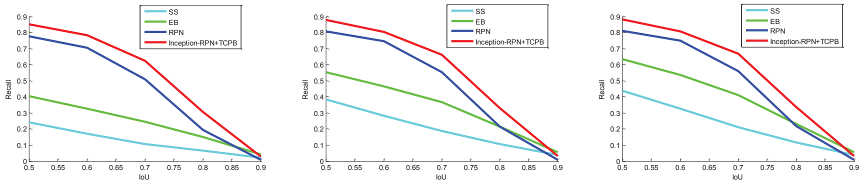

Figure 3: Recall vs. IoU overlap threshold on the ICDAR 2013 testing set. Left: 100 word region proposals. Middle: 300 word region proposals. Right: 500 word region proposals.

Model	TP(%)	FP(%)
Baseline model	85.61	11.20
ATC+MLRP	**88.74**	**10.38**

Table 1: Performance evaluation of ATC and MLPB based on TP and FP rate.

superior to SS and EB with a notable improvement when the number of word region proposals drops from 500 to 100. Therefore, our proposed Inception-RPN-TCPB is capable of achieving a high recall of nearly 90% with only hundreds of word region proposals. Moreover, the recall rate of using 300 word region proposals approximates that of using 500 word region proposals, so we simply use the top-300 word region proposals for the text detection network at test time.

3.3 Analysis of text detection network

In this section, we investigate the effect of ATC incorporation and MLRP on the text detection network. First, we use our proposed Inception-RPN-TCPB to generate 300 word region proposals for each image in the ICDAR 2013 testing set. Next, we assign a positive label to word region proposals that have an IoU overlap greater than 0.5 with a ground-truth bounding box, while assigning a negative label to proposals that has an IoU overlap with any ground-truths of less than 0.5. In total, we collected 8,481 positive word region proposals and 61,419 negative word region proposals. We then evaluated the true positive (TP) rate and false positive (FP) rate of the baseline model and model employing ATC and MLRP. The results are shown in Table 1. It can be seen that the model using ATC and MLRP increase the TP rate by 3.13% and decrease the FP rate by 0.82%, which shows that the incorporation of more reasonable supervised and multi-level information is effective for learning more discriminative features to distinguish text from complex and diverse backgrounds.

3.4 Experimental results on full text detection

We evaluate the proposed DeepText detection system on the ICDAR 2011 and 2013 robust text detection benchmarks following the standard evaluation protocol of ICDAR 2011 [28] and 2013 [15]. Our DeepText system achieves **0.83** and **0.85** F-measure on the ICDAR 2011 and 2013 datasets. Comparisons with recent methods on the ICDAR 2011 and 2013 benchmarks are shown in Tables 2 and 3. It is worth to note that though Sun *et al.* [23] achieved superior results on the ICDAR 2011 and 2013 datasets, their method is not comparable because they used millions of additional samples for training, while we only used 4072 training samples. In the tables, we can see that our proposed approach outperforms previous results

Method	Year	Precision	Recall	F-measure
DeepText (ours)	N/A	0.85	**0.81**	**0.83**
TextFlow [25]	ICCV 2015	0.86	0.76	0.81
Zhang *et al.* [32]	CVPR 2015	0.84	0.76	0.80
MSERs-CNN [10]	ECCV 2014	**0.88**	0.71	0.78
Yin *et al.* [29]	TPAMI 2014	0.86	0.68	0.75
Faster-RCNN [21]	NIPS 2015	0.74	0.71	0.72

Table 2: Comparison with state-of-the-art methods on the ICDAR 2011 benchmark.

Method	Year	Precision	Recall	F-measure
DeepText (ours)	N/A	0.87	**0.83**	**0.85**
TextFlow [25]	ICCV 2015	0.85	0.76	0.80
Zhang *et al.* [32]	CVPR 2015	0.88	0.74	0.80
Lu *et al.* [16]	IJDAR 2015	**0.89**	0.70	0.78
Neumann *et al.*[19]	ICDAR 2015	0.82	0.72	0.77
FASText [2]	ICCV 2015	0.84	0.69	0.77
Iwrr2014 [30]	ACCVW 2014	0.86	0.70	0.77
Yin *et al.* [29]	TPAMI 2014	0.88	0.66	0.76
Text Spotter [20]	CVPR 2012	0.88	0.65	0.75
Faster-RCNN [21]	NIPS 2015	0.75	0.71	0.73

Table 3: Comparison with state-of-art methods on the ICDAR 2013 benchmark.

with a substantial improvement, which can be attributed to simultaneously taking high recall and precision into consideration in our system. The High performance achieved on both datasets highlights the robustness and effectiveness of our proposed approach. Further, qualitative detection results under diverse challenging conditions are shown in Fig. 4, which demonstrates that our system is capable of detecting non-uniform illumination, multiple and small regions, as well as low contrast text regions in natural images. In addition, our system takes 1.7 s for each image on average when using a single GPU K40.

4 Conclusion

In this paper, we presented a novel unified framework called DeepText for text detection in natural images with a powerful fully CNN in an end-to-end learning manner. DeepText consists of an Inception-RPN with a set of text characteristic prior bounding boxes for high quality word proposal generation and a powerful text detection network for proposal classification and accurate localization. After applying an iterative bounding box voting scheme and filtering algorithm to remove redundant boxes for each text instance, we achieve our high-performance text detection system. Experimental results show that our approach achieves state-of-the-art F-measure performance on the ICDAR 2011 and 2013 robust text detection benchmarks, substantially outperforming previous methods. We note that there is still a large room for improvement with respect to recall and precision. In future, we plan to further enhance the recall rate of the candidate word region proposals and accuracy of the proposal classification and location regression.

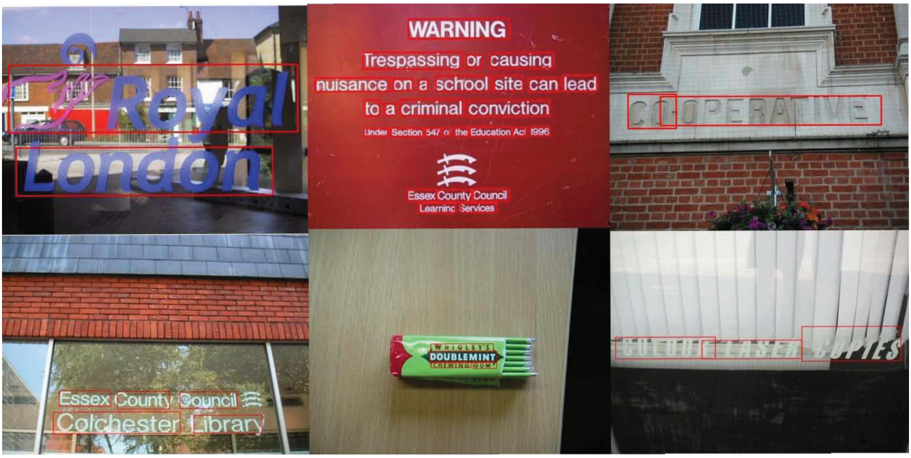

Figure 4: Example detection results of our DeepText system on the ICDAR 2011 and ICDAR 2013 benchmarks.

References

[1] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. CVPR*, 2014.

[2] M. Busta, L. Neumann, and J. Matas. Fastext: Efficient unconstrained scene text detector. In *Proc. ICCV*, 2015.

[3] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *Proc. ICCV*, 2011.

[4] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, 2014.

[5] B. Epshtein, E. Ofek, and Y.Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. CVPR*, 2010.

[6] S. Gidaris and N. Komodakis. Object detection via a multiregion & semantic segmentation-aware cnn model. In *Proc. ICCV*, 2015.

[7] R. Girshick. Fast r-cnn. In *Proc. ICCV*, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. ECCV*, 2014.

[10] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolutional neural networks induced mser trees. In *Proc. ECCV*, 2014.

[11] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proc. ECCV*, 2014.

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1): 1–20, 2016.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arxiv preprint arXiv:1408.5093*, 2014.

[14] D. Karatzas, S. Robles Mestre, J. Mas, F. Nourbakhsh, and P. Pratim Roy. Icdar 2011 robust reading competition. In *Proc. ICDAR*, 2011.

[15] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, , and L. P. de las Heras. Icdar 2013 robust reading competition. In *Proc. ICDAR*, 2013.

[16] S. Lu, T. Chen, S. Tian, J. Lim, and C. Tan. Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition*, 18(2):125–135, 2015.

[17] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *Proc. ICPR*, 2006.

[18] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. ACCV*, 2010.

[19] L. Neumann and J. Matas. Efficient scene text localization and recognition with local character refinement. In *Proc. ICDAR*, 2015.

[20] L. Neumann and K. Matas. Real-time scene text localization and recognition. In *Proc. CVPR*, 2012.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

[23] L. Sun, Q. Huo, and W. jia. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015.

[25] 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 *Proc. ICCV*, 2015.

[26] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arxiv preprint arXiv:1601.07140*, 2016.

[27] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. ICPR*, 2012.

[28] C. Wolf and J. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4):280–296, 2006.

[29] X. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):970–983, 2014.

[30] A. Zamberletti, L. Noce, and I. Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *Proc. Workshop of ACCV*, 2014.

[31] S. Zhang, M. Lin, T. Chen, L. Jin, and L. Lin. Character proposal network for robust text extraction. In *Proc. ICASSP*, 2016.

[32] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proc. CVPR*, 2015.

[33] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. ECCV*, 2014.