Feature Enhancement Network: A Refined Scene Text Detector

Sheng Zhang, Yuliang Liu, Lianwen Jin, Canjie Luo

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

zsscut90@gmail.com, liu.yuliang@mail.scut.edu.cn, {lianwen.jin, canjie.luo}@gmail.com

Abstract

In this paper, we propose a refined scene text detector with a *novel* Feature Enhancement Network (FEN) for Region Proposal and Text Detection Refinement. Retrospectively, both region proposal with *only* 3×3 sliding-window feature and text detection refinement with *single scale* high level feature are insufficient, especially for smaller scene text. Therefore, we design a new FEN network with *task-specific*, *low* and *high* level semantic features fusion to improve the performance of text detection. Besides, since *unitary* position-sensitive RoI pooling in general object detection is unreasonable for variable text regions, an *adaptively weighted* position-sensitive RoI pooling layer is devised for further enhancing the detecting accuracy. To tackle the *sample-imbalance* problem during the refinement stage, we also propose an effective *positives mining* strategy for efficiently training our network. Experiments on IC-DAR 2011 and 2013 robust text detection benchmarks demonstrate that our method can achieve state-of-the-art results, outperforming all reported methods in terms of F-measure.

Introduction

Text detection in natural scene is an important component ((Bissacco et al. 2013), (Yin et al. 2014), (Ye and Doermann 2015), (He et al. 2016b), (He et al. 2017)) for various intelligent applications based on computer vision. For instance, blind navigation, multilingual translation, automotive assistance and image-based geolocation, etc. Different from conventional OCR technique, scene text detection are often challenged by perspective distortions, variation of text size, color or uncontrollable illumination intensity, etc.

In recent years, various methods have been successfully applied for detecting scene text. However, they usually comprise several processing steps, e.g. character or word proposal generation ((Neumann and Matas 2012), (Huang, Qiao, and Tang 2014), (Jaderberg et al. 2016)), proposals filtering and clustering. They often entail much effort in designing heuristic rules and tuning parameters to make each module conduct well, which conversely reduces the speed of text detection.

Currently, Deep Convolutional Neural Networks (DCNN) have advanced general object detection substantially, which is also widely used in the scene text detection field. The main differences between general object detection and scene text detection have been discussed by some previous published papers such as ((Tian et al. 2015), (Sun et al. 2015)), which includes that the scene text are often smaller, thinner and with rich diversity on the aspect-ratios and so on, which makes the detection of scene text a very challenging problem different from general object detection. Besides, for a better performance, almost all object detection algorithms which adopt the DCNN framework take the strategy: i.e. training the proposed network by fine-tuning a model pre-trained on ImageNet dataset (Russakovsky et al. 2015) which is used for image recognition, a different task which requires the extracted features are **position-insensitive**. Contradictorily, the task for object detection is **position-sensitive**. Therefore, (Dai et al. 2016) proposes the *position-sensitive* RoI pooling layer to solve the problem. However, they *only* use the 3×3 sliding-window feature for region proposal and *single scale* high level feature for the refinement of object detection, which is insufficient for general text detection task, specially for smaller text regions; Meanwhile, their *unitary* position-sensitive RoI pooling in general object detection is unreasonable for much variable text regions. Motivated by their work, we propose a refined scene text detection framework via a *novel* Feature Enhancement Network (FEN) which can directly generate word bounding boxes and be end-to-end trainable.

Our key contributions in this paper are as follows:

- We propose a novel FEN network which promotes the recall rate and accuracy of text detection.

- We present an *adaptively weighted* position-sensitive Region of Interest (RoI) pooling module to further raise the accuarcy of the Text Detection Refinement stage.

- We propose a *positives mining* strategy to solve the *sample imbalence* problem during the Text Detection Refinement stage.

- Our approach shows state-of-the-art performance on IC-DAR 2011 and 2013 robust text detection benchmarks.
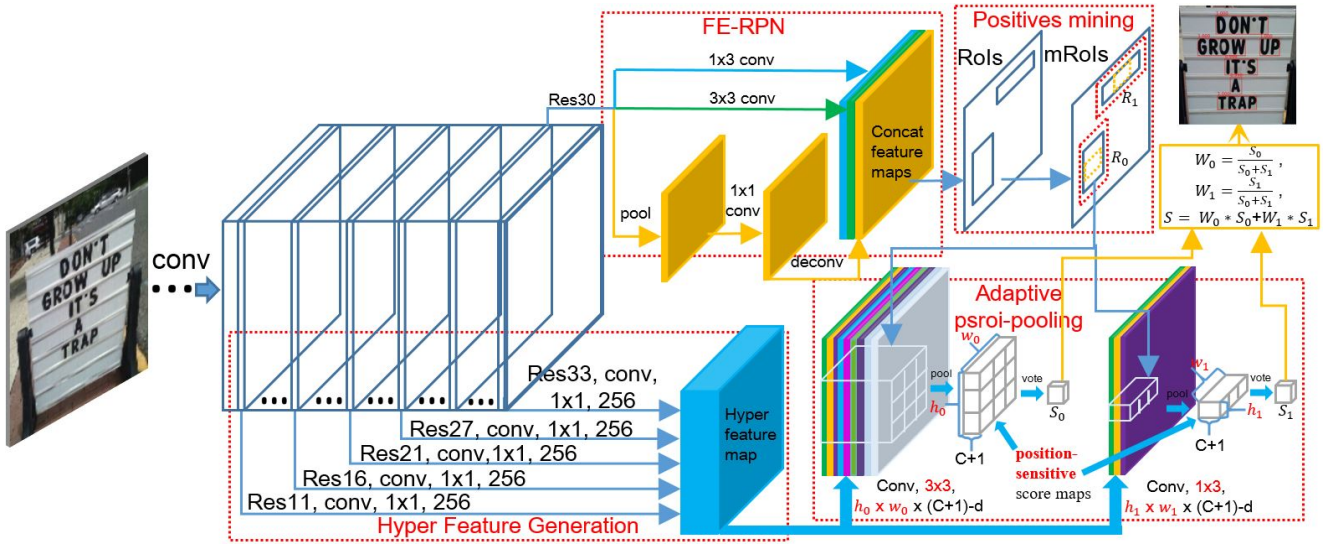
Figure 1: The overall architecture of our FEN. It consists of three innovative components. 1, Feature Enhancement network stem with Feature Enhancement RPN (FE-RPN) and Hyper Feature Generation; 2, Positives mining; 3, Adaptively weighted position-sensitive RoI pooling.

Related Works

In general, text spotting in wild scene can be detailedly categorized by two sub-tasks: text detection and text recognition. The former focuses on accurate text localization in terms of word or text-line bounding boxes in scene images; The latter translates the localized text regions into machine-interpretable character sequences. In this paper, we only concentrate on the text detection task.

Currently, we roughly group mainstream methods for text detection into three categories:

(1) **Character-based:** this kind of method mainly concentrates on the characteristics of individual characters and the relationships between characters (Zhu, Yao, and Bai 2016), *e.g.* connected component based methods ((Zamberletti, Noce, and Gallo 2014), (Shi et al. 2013)). Some of existed approaches often take advantage of Stroke Width Transform (SWT) (Huang et al. 2013) or Maximally Stable Extremal Region (MSER) (Nistér and Stewénius 2008) algorithms to first extract character candidates and then use successive steps to filter non-text patches for exactly connecting the candidates. Although such methods are accurate, they are partly constrained to preserve various true character patches in practice (Cho, Sung, and Jun 2016).

(2) **Word-based:** Text words are considered as the general object to be detected ((Zhong et al. 2016), (Gómez and Karatzas 2017)). (Zhong et al. 2016) proposes a Faster-RCNN based method (Ren et al. 2015). Word proposals are generated with Region Proposal Network and then followed by an embeded text detection refinement stage. However, their performance is not perfect. (Gupta, Vedaldi, and Zisserman 2016) adopts the YOLO framework (Redmon et al. 2016) which directly regresses the text localizations on many regular grids and runs very fast. Nevertheless, it does not perform well on a group of small text regions.

(3) **Text-line-based:** Text lines are first detected by ignoring the noise of blank space between words and then partitioned into individual words. For instance, (Zhang et al. 2015) leverages the symmetric characteristics of text regions to detect text lines and further puts forward to detect text lines with fully convolutional networks (Long, Shelhamer, and Darrell 2015) in year 2016 (Zhang et al. 2016).

Our method is inspired by R-FCN (Dai et al. 2016) and on the basis of text word. Different from original R-FCN network, we have improved the network by *task-specific*, *low* and *high* level semantic features fusion, which observably promote the performance of text detection. Besides, we propose a *positives mining* strategy and an *adaptively weighted* position-sensitive Region of Interest pooling layer which can both remarkably improve the accuary of text detection.

Proposed Methodology

The overall architecture of our FEN network is elaborated in Figure 1. To the best of our knowledge, mainstream methods for general object detection often comprise two stages: proposals generation and detection refinement, so is the R-FCN (Dai et al. 2016) framework. In this paper, we integrate three *innovative* components into the R-FCN framework which inherits the well-known ResNet-101 architecture (He et al. 2016a) and removes the last global pooling, classification layers. By the FEN network, we first enhance the text feature for region proposal through the *FE-RPN* and for text detection refinement with Hyper Feature Generation module; Then, we use *Positives mining strategy* on the region proposals to adjust the ratio between positive and negative samples; Finally, during the text detection refinement stage, we apply the *adaptively weighted* position-sensitive RoI pooling on the hyper-features to produce the accurate text detection results.

Feature Enhancement Network Stem

In our FEN network, we use the ResNet-101 (He et al. 2016a) network as our backbone network. Besides, we have integrated following two components into it.

1) **Feature Enhancement RPN (FE-RPN):** Before, researchers in the field of general object detection always generate region proposals with *only* the 3×3 convolutional sliding-window feature on some intermediate layer, which we reckon that it is inadequate for text region proposal. Since widths of most words or text lines are generally much greater than heights; In other words, their aspect-ratios are usually much greater than one. Meanwhile, high level semantic feature has much larger receptive field and contains much contextual information which is conducive to distinguish foreground objects from background objects. As is shown in Figure 1 top-center: FE-RPN, we select *Res30* layer from the network stem as the input layer of FE-RPN and add two branches, one is the text-characteristic and task-specific 1×3 convolution layer, the other holds max pooling, 1×1 convolution, deconvolution layers, which has two superiorities: 1) what we used is the deconvolution layer proposed by (Odena, Dumoulin, and Olah 2016) which can eliminate the influence of checkerboard artifacts of conventional deconvolution operation radically, 2) make the salient feature more salient and extract much contextual information; Then, we concatenate the original 3×3 convolutional sliding-window feature with the outputs of deconvolution layer and 1×3 convolution layer; Finally, a new convolution layer as well as a *ResNet block* are exploited to achieve the goal of feature enhancement for region proposal.

2) **Hyper Feature Generation:** Previous object detection approaches always make full use of *single scale* and *high level* semantic feature to conduct the refinement of object detection, which may lose much information of object details and thus insufficient for accurate objection localization, especially for smaller text regions. However, the low level semantic feature from intermediate layers will make compensate for the high level semantic feature because of its capability of detail retention. In a word, *high level* semantic feature is conducive to object classification while *low level* feature is beneficial for accurate object localization. Accordingly, (Kong et al. 2016) propose the HyperNet to reinforce the feature map for accurate object localization. In their HyperNet, feature maps originated from different intermediate layers have different *spatial size* and are merged together by pooling, convolution, deconvolution operations, which is computational complexity. For the simplicity, we just compress the intermediate feature maps along the channel dimension with *bottleneck* convolution, which is widely acknowledged to refine the salient features effectively. And more importantly, the intermediate feature maps which we reuse for the hyper-feature are *originally* of the *same* spatial size, which has three superiorities: 1) directly provide the deep supervision information for some intermediate residual blocks; 2) parallelize the residual learning rather than the conventional serialization residual learning; 3) improve the efficiency of computation simultaneously (see the left-bottom of Figure 1).

Text Proposals Generation

1) **Text characteristic anchor design:** Anchor mechanism (Ren et al. 2015) is yet the most classical mechanism for accurate object detection, and different tasks should have different anchor design principle. Specific text detection task also has text characteristic anchor design. As aforementioned, aspect ratios of most words or text lines are prone to be larger than one. Besides, text regions are normally smaller than other general objects in natural scene. We empirically select six scales (32, 64, 112, 192, 304 and 416) and five aspect ratios (1, 2, 3, 4 and 6). Nevertheless, some anchors designed before may be useless and unreasonable. For example, $scale = 416, aspect\_ratio = 6$, it is almost impossible that a text word takes over the whole input image. Finally, we manually keep a total of $a = 24$ anchors at each feature point of the feature enhancement map from FE-RPN sub-network.

Subsequently, we impose the well-designed anchors on each feature point of the feature enhancement map by Equation 1 to attain candidate word proposals.

$$
\begin{aligned}
x &= x_0 + w_0 \times \Delta x, \\
y &= y_0 + h_0 \times \Delta y, \\
w &= w_0 \times exp\left(\Delta w\right), \\
h &= h_0 \times exp\left(\Delta h\right)
\end{aligned}
\tag{1}
$$

Where (x_0, y_0, w_0, h_0) represents the center coordinate, width, height of each anchor, (x, y, w, h) indicates the predicted center coordinate, width, height of each proposal, $(\Delta x, \Delta y, \Delta w, \Delta h)$ is the output of our FE-RPN sub-network and simultaneously denotes the predicted offset for each proposal relate to corresponding anchor. Yet, numerous candidate word proposals are surplus and highly overlap with each other. For this reason, we apply NMS (Neubeck and Van Gool 2006) with an Intersection over Union (IoU) overlap threshold of 0.7 to candidate word proposals to hold the first 200 proposals which are ranked by the scores from FE-RPN sub-network. The held 200 proposals most possibly cover all text regions emerged in the input images and will be fed into the following *Positives mining* layer.

2) **Positives Mining:** Basically, the batchsize of input images in R-FCN (Dai et al. 2016) framework is *only one*, which owns two traits: enhancing the memory of each emerged sample; leading to *samples imbalance*, especially that input images in ICDAR 2011 and 2013 robust text detection datasets frequently contain *only one* text sample and the text region is small, which will cover few anchors for training. Hence, we opt the Positives Mining strategy. As Figure 2 shows, boxes with *shallow blue* color and textness scores are original proposals generated from *FE-RPN* sub-network, others are scaled proposals with scales $\{0.7, 1.3\}$. With these scales, the IoU overlap between the new proposals and original proposals will not exceed previously acknowledged threshold of the RPN NMS (*i.e.* = 0.7). Since we observe that almost all positive proposals rank in top 50 by their scores and we *only* select the first fifty original proposals as references, we call it *Positives*
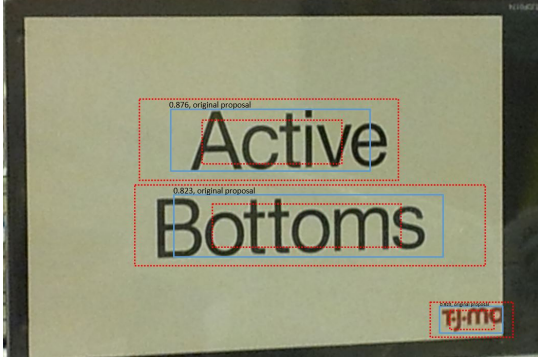
Figure 2: Positives mining. Boxes with *shallow blue* color are original proposals, others are scaled proposals.

mining which can obviously solve the problem of *samples imbalance*.

Text Detection Refinement

Although the text proposal generation stage can recall almost all text regions, it will have a much lower precision for text detection. Thus, in a similar way, we also adopt the text detection refinement stage, which is innovative and different from (Dai et al. 2016) framework.

1) **Adaptively Weighted Position-Sensitive RoI Pooling:** In (Dai et al. 2016) framework, they partition each RoI into $k \times k$ ($k = 7$) bins by a regular grid to encode position information. For an RoI with size $w \times h$, each bin is of size $\frac{w}{k} \times \frac{h}{k}$. Then, they produce k^2 score maps for each category by the last convolutional layer. Inside the (i,j)-th bin $(0 < i, j < k - 1)$, they compute the value of *position-sensitive* RoI pooling layer by pooling *only* over the corresponding position in the (i,j)-th score map:

$$r_c\left(i,j|\Theta\right) = \sum_{(x,y)\in bin(i,j)} \frac{z_{i,j,c}\left(x + x_0, y + y_0 | \Theta\right)}{n} \quad (2)$$

In Equation 2, $r_c\left(i,j|\Theta\right)$ is the pooled value in the (i,j)-th bin for category c, $z_{i,j,c}$ represents a score map from the $k^2(C+1)$ score maps. (x_0, y_0) indicates the top-left corner coordinate of an RoI, n denotes the amount of pixels in the bin, and Θ stands for all network parameters. After the *position-sensitive* RoI pooling procedure, they will get the *coarse* scores of each RoI via globally pooling on the k^2 *position-sensitive* score maps: $r_c\left(\Theta\right) = \sum \frac{r_c(i,j|\Theta)}{k^2}$, which produces a $(C+1)$-dimensional vector. Finally, with the softmax operation across all categories:

$$s_c\left(\Theta\right) = e^{r_c(\Theta)} \Big/ \sum_{c'=0}^{C} e^{r_{c'}(\Theta)} \quad (3)$$

They obtain the *expected* scores which are used for calculating the cross-entropy loss during training and for ranking the RoIs during inference.

As for the bounding box regression, it will be done in a similar way except replacing the aforementioned $k^2(C+1)$-

d convolution layer with a sibling, class-agnostic $4k^2$-d convolution layer.

Although their *position-sensitive* RoI pooling layer performs well on general object detection task, it behaves badly on the scent text detection task. Not only because text regions are normally *smaller* than general objects but also owing to the *exaggerated* aspect-ratios of text words. For example, two text samples with spatial sizes 48×48, 48×256 are in an input image, after the *forward-propagate* procedure, the corresponding text regions in the feature map *Res30* will be 3×3, 3×16 respectively, and now it is obviously unreasonable to impose the *position-sensitive* RoI pooling operation with *conventional* size 7×7. On the contrary, the *practical position-sensitive* RoI pooling should be as Figure 3. Accordingly, as Figure 1 shows, we put forward the *Adap-*
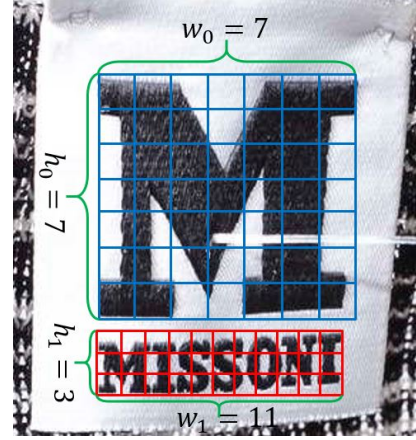

Figure 3: Practical position-sensitive RoI pooling

tively Weighted Position-Sensitive RoI Pooling layer to solve the problem. Firstly, we divide each RoI into $w_l \times h_l$ bins, $(w_l \times h_l) \in \{3 \times 3, 7 \times 7, 3 \times 8, 3 \times 11 | l = 0, 1, 2, 3\}$; Secondly, with the previously generated hyper-feature maps, we produce four kinds of $w_l \times h_l$ score maps with *text-characteristic* filter sizes for each category; then, after applying position-sensitive RoI pooling and global pooling for each RoI to the four kinds of score maps, each RoI will correspond to four scores $(S_l\left(\Theta\right) | l = 0, 1, 2, 3)$, which represent the textness scores with adaptive pooling sizes $\{3 \times 3, 7 \times 7, 3 \times 8, 3 \times 11\}$; Finally, we will leverage the four scores to evaluate the adaptive weight $W_l\left(\Theta\right)$ for each kind of pooling size and the final textness score $S\left(\Theta\right)$ is as follow:

$$W_l\left(\Theta\right) = \frac{S_l\left(\Theta\right)}{\sum\limits_{l'=0}^{4} S_{l'}\left(\Theta\right)}, \quad (l = 0, 1, 2, 3) \quad (4)$$

$$S\left(\Theta\right) = S_l\left(\Theta\right) \times W_l\left(\Theta\right), \quad (l = 0, 1, 2, 3) \quad (5)$$

Clearly, different pooling sizes are suitable for different text regions which own different spatial sizes and aspect-ratios, the most suitable pooling size will get the highest score. Correspondingly, its adaptive weight will be the highest and it will naturally contribute most to the final textness score.

Vice versa. Moreover, with regard to bounding-box regression, we will share the evaluated adaptive weight and do it in the same way, i.e.

$$B(\Theta) = B_l(\Theta) \times W_l(\Theta), \quad (l = 0, 1, 2, 3) \qquad (6)$$

which makes our approach achieve the state-of-the-art results.

Optimization

During the training procedure, we choose the similar multitask loss functions for both text region proposal and text detection refinement stages, i.e.

$$L(s, c, b, g) = \frac{1}{N}(L_{cls}(s, c) + \lambda \times L_{loc}(b, g)) \qquad (7)$$

where N is the amount of anchors or proposals that match ground-truth boxes, and λ ($\lambda = 1$) is a balance factor which weighs the importance between two losses (i.e. L_{cls} : classification error; L_{loc} : localization error). s and c represent the predicted class and ground-truth class respectively. Similarly, b and g separately denote the estimated bounding-box and ground-truth box.

Experiments and Discussion

To prove the effectiveness of our approach, we have tested it on two challenging benchmark datasets, i.e. ICDAR 2011 (Shahab, Shafait, and Dengel 2011) and ICDAR 2013 (Karatzas et al. 2013) robust text detection datasets.

Datasets

ICDAR 2011 includes 229 and 255 challenging scene images for training and testing, respectively. ICDAR 2013 contains 229 training images and 233 testing images, which is similar to the ICDAR 2011 dataset. In view of the facts that all above datasets have few training samples and meanwhile almost all previous algorithms adopt the policy by collecting a lot of *extra real scene* or *synthesized* images for training, we also gather about 4000 real scene images for training our network.

Qualitative evaluation

We vividly display the qualitative experimental results on ICDAR datasets by Figure 4, where, in each sub-figure, red bounding-boxes are our detected results and green bounding-boxes are ground-truth boxes. Meanwhile, each sub-figure is captioned by three evaluation indicators, e.g. R: recall; P: precision; H: H-mean (F-measure). Obviously, our approach performs very well on the challenging scene images. Detailedly, sub-figures (a-c) show that our method can recall and accurately localize all text regions with different scales and aspect-ratios; sub-figure (d) proves that our detector can detect much smaller text word finely; some text words in sub-figures (e-g) are disturbed by shadows and high exposure which are hard for previous algorithms, here, they can be robustly detected. However, there also exist some text words or background objects we can not deal with finely. For instance, in the last row of Figure 4, it results in false positive, missed detection. Besides, in sub-figure (g), for words

"HOT MILK", the ground-truth bounding-box may be unreasonable in terms of word-based text detection. In a word, a few erroneous detections are from above three cases.

Quantitative evaluation

1) **The effectiveness of Feature Enhancement Network Stem (FENS):** The original R-FCN framework proposed by (Dai et al. 2016) is suitable for the general object detection and will not fit well for the specific text detection task, which has been evidenced by the first row in Table 1. Clearly, although its precisions are comparable to some methods with our innovative components, it has a low recall rate. As expected, in the last three row of Table 1, experiment results on the two datasets consistently show approximatively 5% gain on the recall rate and 3% gain on the F-measure with our posed Feature Enhancement Network Stem (FENS). However, good and evil are coexisted, our detection speed is almost twice of theirs.

2) **The effectiveness of Positives Mining (PM):** As Figure 5 shows, during the training procedure, the ratio between the numbers of foreground and background samples is less than 1 : 4 by the original R-FCN framework, which *badly* violates the widely acknowledged *minimum* ratio 1 : 3 in all object detection algorithms. For the reason, it has been elaborated in subsection **Positives Mining**. On the contrary, our proposed Positives Mining strategy can obviously improve the situation with approximatively 2 : 5 ratio, which resolves the conventional problem of *sample imbalance*. Besides, we add the half-region proposals (Gidaris and Komodakis 2015) during the test stage, which can improve the precision clearly in the last four row of Table 1. As a whole, this strategy promotes the F-measure by 0.5% roughly.

3) **The effectiveness of adaptively weighted position-sensitive RoI pooling:** In the last two and three rows of Table 1, we have verified the effectiveness of our proposed *adaptively weighted* position-sensitive RoI pooling layer. On the ICDAR 2013 dataset, it avaragely improves the F-measure by 1.9% against *conventional* position-sensitive RoI pooling on both evaluation criterions; On the ICDAR 2011 dataset, it will not enhance the F-measure a lot, just by 1.1% against *conventional* position-sensitive RoI pooling averagely.

4) **Comparisons with other state-of-the-art results:** Our approach and the original R-FCN are both trained and tested with short side 720 except the multi-scale test, which is a method that first tests each input image with different resolutions (e.g. 720×1000, 960×1200, 1200×1600 etc.); then merges all the results together; and last, applys the NMS algorithm on all the results to produce the final results. The multi-scale test strategy is also used for TextBoxes (Liao et al. 2017). Although our short side is 720, we can find from the comparison of single scale and multi-scale test of our FEN that the input resolution doesn't influence the final results too much. All the experiments are carried out on a PC with one Titan X GPU. As can be found in

(a) R: 100%; P: 100%; H: 100%	(b) R: 100%; P: 100%; H: 100%	(c) R: 100%; P: 100%; H: 100%
(d) R: 100%; P: 100%; H: 100%	(e) R: 100%; P: 100%; H: 100%	(f) R: 100%; P: 100%; H: 100%
(g) R: 98.33%; P: 96.92%; H: 97.62%	(h) R: 100%; P: 85.71%; H: 92.31%	(i) R: 66.67%; P: 80%; H: 72.73%

Figure 4: Qualitative evaluation results on ICDAR datasets. Green boxes are ground-truth boxes, red boxes are detected results. R: recall; P: precision; H: H-mean (F-measure)

Table 1: The effectiveness of different components of our method on ICDAR 2011 and 2013 robust text detection datasets. IC13 Eval: ICDAR 2013 evaluation criterion; DetEval: (Wolf and Jolion 2006); R: recall; P: precision; F: F-measure. PM: Positives Mining. FENS: Feature Enhancement Network Stem. MT: multi-scale test.

Datasets	ICDAR 2011						ICDAR 2013						Time/s
Evaluation protocol	IC13 Eval			DetEval			IC13 Eval			DetEval			
Methods	R	P	F	R	P	F	R	P	F	R	P	F	
R-FCN (Dai et al. 2016)	0.812	0.873	0.841	0.824	0.881	0.852	0.819	0.907	0.861	0.827	0.910	0.867	**0.14**
R-FCN + our PM	0.810	0.895	0.850	0.821	**0.899**	0.858	0.821	0.918	0.867	0.825	0.923	0.871	0.16
R-FCN + our FENS	0.865	0.874	0.869	0.872	0.886	0.879	0.878	0.907	0.892	0.884	0.918	0.90	0.27
our FEN	0.875	0.892	0.884	0.881	0.894	0.887	0.891	0.936	0.913	0.897	0.939	0.918	0.31
our FEN + MT	**0.889**	**0.896**	**0.892**	**0.895**	0.898	**0.897**	**0.893**	**0.941**	**0.916**	**0.90**	**0.947**	**0.923**	0.90

Table 2: Comparison with state-of-the-art methods on ICDAR 2011 and 2013 robust text detection datasets. IC13 Eval: ICDAR 2013 evaluation criterion; DetEval: (Wolf and Jolion 2006); R: recall; P: precision; F: F-measure. MT: multi-scale test.

| Datasets | ICDAR 2011 | | | | | | ICDAR 2013 | | | | | | Time/s |
Evaluation protocol	IC13 Eval			DetEval			IC13 Eval			DetEval			
Methods	R	P	F	R	P	F	R	P	F	R	P	F	
TextFlow (Tian et al. 2015)	0.76	0.86	0.81	-	-	-	0.76	0.85	0.80	-	-	-	1.4
FCRNall + filts (Gupta, Vedaldi, and Zisserman 2016)	-	-	-	0.75	0.92	0.82	-	-	-	0.76	0.92	0.83	>1.27
FCN (Zhang et al. 2016)	-	-	-	-	-	-	0.78	0.88	0.83	-	-	-	2.1
TextBoxes + MT (Liao et al. 2017)	0.82	0.88	0.85	0.82	0.89	0.86	0.83	0.88	0.85	0.83	0.89	0.86	0.73
DeepText (Zhong et al. 2016)	-	-	-	0.81	0.85	0.83	0.826	0.904	0.863	0.842	0.907	0.873	1.7
CTPN (Tian et al. 2016)	-	-	-	0.79	0.89	0.84	0.83	0.93	0.88	-	-	-	**0.14**
EASTtext (Zhou et al. 2017)	-	-	-	-	-	-	0.827	0.926	0.874	-	-	-	0.75
our FEN	0.875	0.892	0.884	0.881	0.894	0.887	0.891	0.936	0.913	0.897	0.939	0.918	0.31
our FEN + MT	**0.889**	**0.896**	**0.892**	**0.895**	**0.898**	**0.897**	**0.893**	**0.941**	**0.916**	**0.90**	**0.947**	**0.923**	0.90
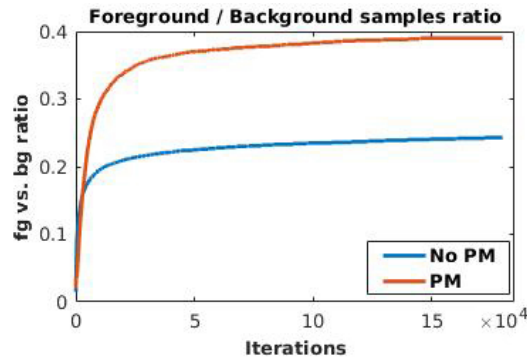
Figure 5: Training with/without Positives Mining (PM)

Table 2, the results of our proposed framework exceed all other state-of-the-art methods by a large margin in terms of no matter recall, precision, F-measure indicators, which strongly proves the effectiveness of our method. In addition, the speed of our approach is also comparable to other state-of-the-art methods in light of single scale test. (We haven't compared with (Sun et al. 2015) which has achieved superior results with many contributions on the ICDAR 2011 and 2013 datasets, however, they have used millions of additional samples for training).

Conclusion

In this paper, we have presented the novel Feature Enhancement Network for accurate real scene text detection. It contributes in three aspects: 1) improving the recall rate evidently with *innovative* Feature Enhancement Network Stem which includes both Feature Enhancement RPN and Hyper feature generation for text detection refinement; 2) accurately detecting text regions with *adaptively weighted* position-sensitive RoI pooling layer; 3) solving the *sample imbalance* problem by *positives mining* strategy. Comprehensive evaluations and comparisons on benchmark datasets show that our method can achieve the state-of-the-art results.

In the future, we will extend our work to multi-oriented scene text detection and end-to-end word spotting.

Acknowledgments

References

Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, 785–792.

Cho, H.; Sung, M.; and Jun, B. 2016. Canny text detector: Fast and robust scene text localization algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3566–3573.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, 379–387.

Gidaris, S., and Komodakis, N. 2015. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, 1134–1142.

Gómez, L., and Karatzas, D. 2017. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition* 70:60–74.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, T.; Huang, W.; Qiao, Y.; and Yao, J. 2016b. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing* 25(6):2529–2541.

He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; and Li, X. 2017. Single shot text detector with regional attention. *arXiv preprint arXiv:1709.00138*.

Huang, W.; Lin, Z.; Yang, J.; and Wang, J. 2013. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, 1241–1248.

Huang, W.; Qiao, Y.; and Tang, X. 2014. Robust scene text detection with convolution neural network induced mser trees. In *European Conference on Computer Vision*, 497–511. Springer.

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

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and de las Heras, L. P. 2013. Icdar 2013 robust reading competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 1484–1493. IEEE.

Kong, T.; Yao, A.; Chen, Y.; and Sun, F. 2016. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 845–853.

Liao, M.; Shi, B.; Bai, X.; Wang, X.; and Liu, W. 2017. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 4161–4167.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Neubeck, A., and Van Gool, L. 2006. Efficient non-maximum suppression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, 850–855. IEEE.

Neumann, L., and Matas, J. 2012. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 3538–3545. IEEE.

Nistér, D., and Stewénius, H. 2008. Linear time maximally stable extremal regions. *Computer Vision–ECCV 2008* 183–196.

Odena, A.; Dumoulin, V.; and Olah, C. 2016. Deconvolution and checkerboard artifacts. *Distill*.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Shahab, A.; Shafait, F.; and Dengel, A. 2011. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 1491–1496. IEEE.

Shi, C.; Wang, C.; Xiao, B.; Zhang, Y.; Gao, S.; and Zhang, Z. 2013. Scene text recognition using part-based tree-structured character detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2961–2968.

Sun, L.; Huo, Q.; Jia, W.; and Chen, K. 2015. A robust approach for text detection from natural scene images. *Pattern Recognition* 48(9):2906–2920.

Tian, S.; Pan, Y.; Huang, C.; Lu, S.; Yu, K.; and Lim Tan, C. 2015. Text flow: A unified text detection system in natural scene images. In *Proceedings of the IEEE International Conference on Computer Vision*, 4651–4659.

Tian, Z.; Huang, W.; He, T.; He, P.; and Qiao, Y. 2016. Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision*, 56–72. Springer.

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

Ye, Q., and Doermann, D. 2015. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence* 37(7):1480–1500.

Yin, X.-C.; Yin, X.; Huang, K.; and Hao, H.-W. 2014. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence* 36(5):970–983.

Zamberletti, A.; Noce, L.; and Gallo, I. 2014. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *Asian Conference on Computer Vision*, 91–105. Springer.

Zhang, Z.; Shen, W.; Yao, C.; and Bai, X. 2015. Symmetry-based text line detection in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2558–2567.

Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; and Bai, X. 2016. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4159–4167.

Zhong, Z.; Jin, L.; Zhang, S.; and Feng, Z. 2016. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv preprint arXiv:1605.07314*.

Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; and Liang, J. 2017. East: An efficient and accurate scene text detector. *arXiv preprint arXiv:1704.03155*.

Zhu, Y.; Yao, C.; and Bai, X. 2016. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science* 10(1):19–36.