2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Robust Scene Text Detection with Deep Feature Pyramid Network and CNN based NMS Model

Sabyasachi Mohanty, Tanima Dutta and Hari Prabhat Gupta
Dept. of Computer Science and Engineering, IIT (BHU) Varanasi, India

***Abstract*—Scene text detection has attracted great interest from the computer vision and pattern recognition communities since text information plays an important role in image indexing and scene understanding. Deep neural networks have become popular for the task of scene text detection, especially for their ability to learn strong text features. However, existing deep learning based state-of-the-art scene text detection methods detect texts only from a single feature map which is unable to capture semantic information at all scales. In this paper, we propose a novel deep learning based model that leverages the pyramid structure of feature maps for accurate scene text detection. We also design a deep convolutional neural network model for non-maximum suppression. In addition, we develop a novel loss function and training method for end-to-end training. The experimental results validate that our end-to-end system is simple, fast, and achieves high accuracy on standard datasets, namely, ICDAR 2015 and MSRA-TD500. We also create a dataset for scene text detection.**

I. Introduction

Scene text detection is one of the classical problems in the field of computer vision. Scene text detection refers to the task of precisely localizing all instances of texts in a scene image with a bounding box or quadrangle. Localization of texts can be done at the character, word, or text line level depending on the application. Text detection in scene images has numerous applications, such as indoor navigation, autonomous driving, and robot vision. Scene text detection is very challenging because texts can have a variety of color, size, aspect ratio, font, script, and orientation. Moreover, factors such as uneven lighting, noise, and occlusion make the task very difficult.

A revolutionary change has been seen in the field of computer vision with the development of deep learning based models in last few years. Convolutional Neural Network (CNN) models have significantly boosted the performance of the task of generic object classification and detection [1]–[6]. CNN models have also been widely adopted for the task of scene text detection [7]–[14]. Very deep CNNs capture strong features that provide powerful semantic information about texts. The deeper layers have the strongest features. However, the resolution of feature maps in deep layers is less which leads to large receptive fields of these maps on the input image. As a result, they are not able to detect small objects well. On the other hand, shallow layers have less semantic information to detect complex text regions. To overcome these limitations, there is a recent trend of combining feature maps of deep and shallow layers to perform accurate detection [5], [8], [9]. However, the final detections are performed on a single map [5], [8], [9], [14] which is not able to capture information at all scales. To address this problem, we propose a new feature pyramid based model where feature maps capturing different amounts of semantic information form a pyramid and predictions are made separately at each pyramid level (scale).

In generic object detection, anchor boxes, introduced by [3], are often used as references for detecting objects. However, anchor boxes perform poorly in the detection of text regions [5], [11]. This is because texts have varying scale, aspect ratio, and orientation which makes it difficult to design reference anchor boxes for them. Therefore, the concept of anchor boxes is not very useful. In our model, every pixel in each final feature map detects the words that fall inside the part of the input image on which the pixel has its receptive field. Deep learning based scene text detection models usually use greedy Non-Maximum Suppression (NMS) [8], [11] as a post-processing step to suppress redundant word detections. Instead, we design a CNN model for performing NMS. Moreover, we develop a new loss function that makes it possible to jointly train the two CNN models that are performing different tasks, *i.e.*, word quadrangle detection and NMS.

Scene text detection methods often follow character level approaches [12], [13], [15], [16]. In character level approach, text characters are detected and then merged into words or text lines using heuristic based techniques. However, character level detection has its limitation as it requires several post-processing stages, such as removing false characters, word formation from the characters, and optional segmentation. Error is accumulated over each stage which reduces the detection accuracy [17]. Instead, our text detection model detects quadrangles directly at the word level.

The major contributions of this paper are summarized as follows:

- *Feature Pyramid Text (FP-Text)*: A novel deep CNN model based on a pyramid of feature maps is proposed for scene text detection.
- *NMS-Text*: We also develop a new CNN model to perform NMS of the text word quadrangles detected by FP-Text.
- *FP-NMS-Text*: We develop a novel loss function that facilitates end-to-end training of both FP-Text and NMS-Text as a single joint system (network), denoted by FP-NMS-Text.
- *Dataset Creation*: We create a new scene text detection dataset containing texts of various size, color, orientation, aspect ratio, and under different conditions of illumination. It contains scene texts in challenging scenarios.

978-1-5386-3787-6/18/$31.00 ©2018 IEEE

- *Performance*: We have seen that our system performs better in terms of f-measure on most of the publicly available scene text datasets as well as on our scene text dataset. Moreover, we found that our end-to-end system is very fast and can be performed in real time. The results are either better or comparable in terms of speed with state-of-the-art scene text detection methods.

The rest of the paper is organized as follows: The next section describes the proposed methodology for scene text detection. Section III illustrates the experimental results and we conclude the paper in Section IV.

II. Proposed Methodology

In this section, we propose an efficient system for scene text detection. Fig. 1 depicts the block diagram of our joint system. We first describe the architecture of FP-Text along with the ideas leading to this architecture. Then, we give details of our NMS-Text model and elaborate upon the end-to-end training method used for joint training of FP-Text and NMS-Text.

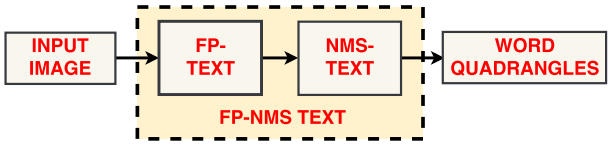

Fig. 1. Block diagram of the proposed system.

A. Architecture of FP-Text

The layers closer to the input (early layers) of a CNN produce feature maps with high resolution but limited semantic information. On the other hand, deep layers have very strong semantic information in their feature maps but the maps have low resolution. This makes it very difficult to detect small text instances from the deep feature maps. FP-Text is designed to address these issues. The architecture introduces four components, which are shown in Fig. 2.

1) DownSampling Branch: The downsampling branch of FP-Text performs feed-forward operation on the input image. We call each set of consecutive convolutional layers producing feature maps of the same resolution as a *stage*. The downsampling branch has six stages ($conv_1$ to $conv_6$). The stages $conv_1$ and $conv_6$ have a single layer while the other stages have two layers. All the layers of a given stage produce feature maps of the same resolution and depth. Each stage is followed by a maxpool operation [5], [8] which reduces the feature map resolution by half. Also, the depth of the feature maps doubles in each successive stage. Thus, the resolutions of the feature maps from stage $conv_1$ to $conv_6$ are $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ times that of the input image. The strength of semantic information captured is the least in the feature map of $conv_1$ stage while it is the highest in the feature map of $conv_6$ stage.

2) UpSampling Branch: The feature maps in the deep layers of the downsampling branch have low resolution due to which their receptive field on the input image is very large. This prevents them from capturing small text regions, such as single character words. The upsampling branch addresses this issue by increasing the resolution of the deep feature maps. This is performed by an unpool operation [8] which is the reverse of maxpooling. This branch also has six stages ($unconv_1$ to $unconv_6$) corresponding to the six stages of the downsampling branch. Each stage of the upsampling branch has layers of the same resolution and depth as those of the corresponding stage of the downsampling branch. Each layer in both the branches is followed by Rectified Linear Unit (ReLU) non-linearity [5], [8].

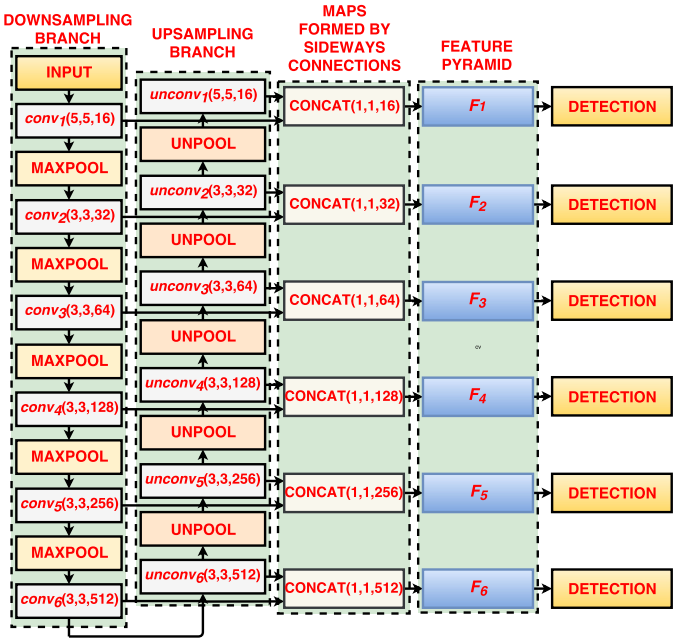

Fig. 2. Architecture of FP-Text. Here, (x, y, z) in a block represents the resolution (x,y) and depth (z) of the feature map produced in that block.

3) Sideways Connections: The feature maps in the upsampling branch have strong context information and good resolution, especially the maps at the top of the branch. However, these maps have undergone large amounts of sampling (both downsampling and upsampling), due to which their activations are poorly localized. The sideways connections merge (concatenate) the feature maps in the corresponding layers of the corresponding stages in the downsampling and upsampling branches. This helps the feature maps in the downsampling branch to provide good localization to the activations in the feature maps of the upsampling branch. This is because the feature maps in the downsampling branch have been sampled fewer times. The depth of these concatenated maps is then reduced by half using 1×1 filters for memory efficiency. The concatenated maps after depth reduction are the final feature maps, referred by $F_{1-6} = \{F_1, F_2, F_3, F_4, F_5, F_6\}$ and form a feature pyramid.

4) Feature Pyramid based Detection: The maps in F_{1-6} are formed after reduction due to which some semantic information is lost. As a result, no specific map has captured complete semantic information about the input image. To tackle this issue, we propose to use feature pyramid based

detection. In this proposal, the detection module is applied to each final feature map separately and the word detections are finally merged using NMS-Text. It is to be noted that we have not used anchor box based predictions as done in [4]. Instead, we predict a word quadrangle per pixel of the final feature map [1]. The detection module is shown in Fig. 3(a). It has three convolutional layers and the final convolutional layer produces the output map. There are six output maps $\{O_1, O_2, O_3, O_4, O_5, O_6\}$ corresponding to F_{1-6} maps, respectively. Each pixel in each output map has a depth of 9, which represents the text confidence score as well as the 8 vertices of the detected word quadrangle.

B. Architecture of NMS-Text

Each output map of FP-Text consists of several word quadrangles as its final detections. Traditionally, greedy NMS (G-NMS) [8], [11] is used as a post-processing step to merge the quadrangles that have high overlap. However, we aim to create an end-to-end trainable scene text detector. So, we propose NMS-Text, a CNN based method, for merging the quadrangles. The method presented in [18] for generic object detection is modified in NMS-Text to make it more accurate and effective for scene text detection. To the best of our knowledge, no work exists in the literature of scene text detection that uses a deep model for NMS. In addition, the proposed work is also the first one to train, in an end-to-end fashion, both detection (FP-Text) and NMS (NMS-Text) networks for the task of scene text detection. As a result, we require no further post-processing. Training and testing are also very fast.

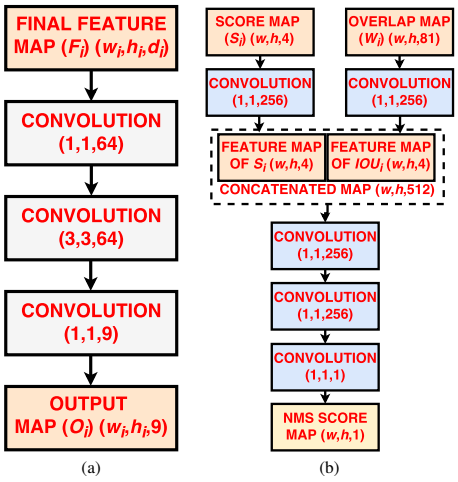

Fig. 3. Columns (a) and (b) illustrate the architectures of detection module and NMS-Text, respectively, where (w,h) is the resolution of S_i and W_i. Here, (x, y, z) in a convolution block represents the resolution (x,y) and depth (z) of the feature map produced by that block.

NMS-Text is applied separately to each output map of FP-Text. For the output map O_i at a particular scale i of the feature pyramid, NMS-Text takes two maps as input, namely, a score map (S_i) and an overlap map (W_i). Both these input maps are obtained from O_i.

1) Score Map: For each pixel in O_i, we consider a neighboring window of size 9×9. After that, G-NMS of all the word quadrangles in this window is performed. The threshold α for this G-NMS is a design choice. The choice of α is a trade-off between precision and recall in G-NMS. To make our model robust, inspired by the work in [18], we include a range of values of α. The score map $\delta_i(\alpha)$ is obtained after applying G-NMS at threshold α in each window of O_i. The final score map S_i is the map formed by the concatenation of score maps obtained at thresholds 0.25, 0.5, 0.75, and 1.0. S_i is given by the following equation where $|$ represents concatenation.

$$S_i = \delta_i(0.25)|\delta_i(0.5)|\delta_i(0.75)|\delta_i(1.0). \quad (1)$$

2) Overlap Map: Again we consider a 9×9 window for each pixel P_j in O_i. For each of the 81 pixels in the window, the overlap with P_j is computed. By overlap of two pixels, we mean the Jaccard overlap (W) of the quadrangles at the 2 pixels. The 81 values obtained for each pixel is encoded and a feature map of depth 81 is produced known as overlap map W_i. This feature map represents the overlap (W) of each pixel with its 81 neighbors (including itself).

The maps S_i and W_i are convolved with 1×1 filters and then the resulting maps are concatenated. This concatenated map is then convolved with three more layers to obtain the final output map of depth one. This output map represents the confidence score of each pixel of having a word quadrangle after NMS. With the information provided by S_i and W_i, the network decides whether a particular detection score corresponds to a correct detection or should be suppressed by a neighboring detection score. Fig. 3(b) illustrates architecture of NMS-Text.

C. End-to-End Training of FP-Text and NMS-Text

P-Text and NMS-Text are two different components of the overall system that perform different tasks. FP-Text predicts the confidence score of containing a text word for each pixel of the feature maps at different levels of the feature pyramid along with 8 coordinates of the word quadrangle. NMS-Text takes all the quadrangles output by FP-Text and performs fast NMS on them and obtains the final set of quadrangles. The joint training of these 2 networks is a challenging task. We design a multi-task loss function L_{mul} to train the two networks (FP-Text and NMS-Text) as a joint system, referred by FP-NMS-Text. L_{mul} is defined as follows:

$$L_{mul} = L_{pred} + \lambda_1 L_{nms}, \quad (2)$$

where L_{pred} and L_{nms} are the losses of FP-Text and NMS-Text, respectively. λ_1 is chosen as 1 to give equal weights to the two losses.

1) Prediction Loss (L_{pred}): The prediction loss is the total loss due to the output predictions of FP-Text. Let a pixel i in the output map at level j predict a text quadrangle Q_{ij} with coordinates encoded in the vector $\hat{z}_{ij}$ with a confidence score c_{ij}. Let the true label of Q_{ij} be c^*_{ij}. A detected quadrangle is said to have a true label of *text* if it has atleast 0.5 Jaccard overlap with any of the Ground Truth (GT) text quadrangles. If Q_{ij} has a true label of text, then $c^*_{ij} = 1$, otherwise $c^*_{ij} = 0$.

If Q_{ij} has text label, then the quadrangle with which it has the maximum overlap is considered as its GT quadrangle with coordinates encoded as the vector $\hat{z}^*_{ij}$. L_{pred} consists of 2 losses, namely classification loss (L_{cls}) and localization loss (L_{cls}), which are defined as follows:

$$L_{pred} = L_{cls} + \lambda_2 L_{loc}, \tag{3}$$

$$L_{cls} = \sum_{j=1}^{level} \sum_{i=1}^{res_j} f_1(c_{ij}, c^*_{ij}), \tag{4}$$

$$L_{loc} = \sum_{j=1}^{level} \sum_{i=1}^{res_j} f_2(\hat{z}_{ij}, \hat{z}^*_{ij}), \tag{5}$$

where $level$ refers to the total number of levels in the feature pyramid of FP-Text and res_j is the resolution of the final feature map at level j. The loss function f_1 is chosen as class-balanced cross-entropy function [5] to facilitate easier training. The smooth L_1 loss [11] is chosen as the loss function f_2 since it is less sensitive to outliers. The value of λ_2 is 1.

2) NMS Loss: NMS loss refers to the loss introduced by NMS-Text. NMS-Text operates in two stages. In the first stage, it outputs a set of final quadrangles at each level of the feature pyramid from the set of quadrangles output by FP-Text at that level. Let the loss for this stage be L_{nms1}. In the second stage, NMS-Text outputs the set of final quadrangles from the set of quadrangles output by NMS-Text in the first stage. We denote the loss for this stage by L_{nms2}. The various NMS related losses are defined as follows:

$$L_{nms} = L_{nms1} + \lambda_3 L_{nms2}, \tag{6}$$

$$L_{nms1} = \sum_{j=1}^{level} \sum_{i=1}^{res_j} f_3(t_{ij}, t^*_{ij}), \tag{7}$$

$$L_{nms2} = \sum_{i=1}^{res} f_3(t_i, t^*_i), \tag{8}$$

where t_{ij} refers to the confidence score of pixel i at level j of the feature pyramid of containing a text quadrangle after first stage of NMS. Similarly, t_i refers to the text confidence score of a pixel i after the second stage of NMS. t^*_{ij} and t^*_i are the corresponding true labels. $level$ and res_j have same meanings as before and res is the resolution of the output map of the second stage of NMS-Text. In addition, we select f_3 as the softmax loss [5] and λ_3 is 1.

Other Training Details: FP-NMS-Text is pre-trained on SynthText [10] dataset and fine-tuned on the real dataset on which it is to be tested. We use standard stochastic gradient descent [5], [8] for learning with a momentum of 0.9. Input images are fed to FP-NMS-Text in mini batches and each mini branch contains 32 images. To make our model robust, we choose a multi-scale training scheme [2]. For the second stage of NMS-Text, we consider only the L_{nms2} loss for training, while the complete L_{nms} loss is used for training the first stage of NMS-Text. For the FP-Text part of our joint system, a sum of both L_{pred} and L_{nms} losses is used for training.

Online Hard Negative Mining: In all the scene text detection datasets [19], [20], negatives (non-texts) constitute most of the training samples. This makes training unbalanced and leads to slow convergence. To overcome this problem, we make use of the online hard negative mining technique [5]. In this technique, only the negative samples that give maximum loss are chosen for training so that the ratio of negative samples to positive ones is maintained at $3:1$. This ratio is maintained for each mini batch during pre-training as well as fine-tuning.

Data Augmentation: Data augmentation refers to the variations introduced in an input image to make the model learn stronger features. We use data augmentation techniques such as random cropping, horizontal flipping, small scale rotations, and small scale translations [1], [3] of the input image.

III. Experiments

The proposed system (FP-NMS-Text) is compared with several state-of-the-art methods of scene text detection on two publicly available datasets, namely, ICDAR 2015 [19] and MSRA-TD500 [20] as well as on the scene text dataset created by us. We also use the SynthText dataset [10] to pre-train our model. The datasets used are described as follows:

SynthText Dataset: The SynthText dataset [10] is a synthetic dataset containing 800,000 training images. It is created by blending natural images with texts of random sizes and fonts. GT is provided at character, word, and text line levels.

ICDAR 2015 Dataset: This dataset is part of the ICDAR 2015 Robust Reading Competition [19]. It contains 1000 images for training and 500 images for testing. Here, the texts in scene images are present as incidental occurrences. As a result, the texts are not focused and have low resolution. GT is present at the word level in the form of quadrangles.

MSRA-TD500 Dataset: MSRA-TD500 [20] dataset contains 300 and 200 images for training and testing, respectively. Here the texts are stable with high resolution. GT is provided at text line level with rotated rectangles as the bounding boxes. This dataset contains texts of both English and Chinese scripts. This dataset has a large number of multi-oriented texts.

Our Scene Text Dataset: We create a new scene text dataset to offer a range of challenging scenarios. Our dataset contains only English script. It has 500 images for training and 500 more for testing. The texts in this dataset offer a wide variety of font, scale, aspect ratio, color, rotation, resolution, and occur in varying lighting conditions. The dataset contains GT at the word level as quadrangles.

A. Implementation Details

Our end-to-end system (FP-NMS-Text) is pre-trained on the SynthText dataset [10]. FP-NMS-Text is implemented using NVIDIA Titan X graphic card along with an Intel E5-2670v3 CPU running at 2.30 GHz. Since the end-to-end system is fully convolutional, it accepts inputs of any size. During testing, a quadrangle with confidence score above the threshold β is chosen as a text quadrangle.

TABLE I
PERFORMANCE ON DIFFERENT SCENE TEXT DATASETS

Dataset	**ICDAR 2015**			**MSRA-TD500**			**Our Scene Text Dataset**		
Method	**Precision**	**Recall**	**f-measure**	**Precision**	**Recall**	**f-measure**	**Precision**	**Recall**	**f-measure**
Proposed	**0.82**	**0.83**	**0.82**	0.80	**0.78**	**0.79**	**0.78**	**0.76**	**0.77**
He *et al.* [11]	0.82	0.80	0.81	0.77	0.70	0.74	0.76	0.73	0.75
DMPNet [7]	0.73	0.68	0.71	0.75	0.66	0.70	0.71	0.66	0.68
EAST [8]	0.81	0.68	0.74	0.83	0.67	0.74	0.77	0.70	0.73
SegLink [5]	0.73	0.77	0.75	**0.86**	0.70	0.77	0.75	0.71	0.73
Zhang *et al.* [9]	0.71	0.43	0.54	0.83	0.67	0.74	0.58	0.67	0.62
Tian *et al.* [21]	0.74	0.52	0.61	0.80	0.68	0.74	0.68	0.64	0.66
Yao *et al.* [22]	0.72	0.59	0.65	0.77	0.75	0.76	0.66	0.61	0.63
Yin *et al.* [23]	0.78	0.72	0.75	0.81	0.63	0.71	0.72	0.68	0.70

TABLE II
TEST TIME SPEED OF DIFFERENT CNN BASED METHODS

Method	**Resolution**	**Device**	**fps**
Proposed	MS	Titan X	13.52
He *et al.* [11]	MS	Titan X	1.11
DMPNet [7]	720p	Titan X	0.95
EAST [8]	720p	Titan X	**16.80**
SegLink [5]	720p	Titan X	8.90
Zhang *et al.* [9]	MS	Titan X	0.48
Tian *et al.* [5]	MS	GPU	7.14
Yao *et al.* [22]	480p	K40m	1.61
Yin *et al.* [23]	720p	Titan X	0.71

B. Text Detection Performance Comparison

We compare the performance of our framework with deep learning based state-of-the-art scene text detection methods [5], [7]–[9], [11], [21]–[23]. The performance of our system on various datasets in terms of precision, recall, and f-measure is described below:

ICDAR 2015 Dataset: On this dataset, we use the value of β as 0.8. Table I shows that FP-NMS-Text achieves highest values in terms of all three comparison metrics when compared with state-of-the-art methods. The high precision can be attributed to the robust NMS performed by NMS-Text while high recall is obtained due to the use of feature maps from different scales of the feature pyramid for detection.

MSRA-TD500 Dataset: This dataset provides large instances of multi-oriented texts as well as Chinese script. We use β as 0.6 on this dataset. We obtain high recall and f-measure on MSRA-TD500 as seen from Table I. The method [5] achieves high precision since it uses local text information through segments and links. However, as illustrated in Fig. 5, the method [5] is unable to detect words with large inter-character distance in the MSRA-TD500 [20] dataset. On the other hand, FP-NMS-Text is able to detect such words because we detect texts directly at the word level and not by combination of local text components (segments and links).

Our Scene Text Dataset: Our dataset has texts in various fonts, colors, and orientations, along with varying lighting conditions. As observed from Table I, our system is very robust in scene text detection. This is the reason we obtain the best performance in terms of precision, recall, and f-measure. We use 0.6 as the value of β.

TABLE III
PERFORMANCE ON ICDAR 2015 DATASET OF DIFFERENT WAYS OF TRAINING FP-TEXT AND NMS-TEXT

Method	**Precision**	**Recall**	**f-measure**
FP-NMS-Text	**0.82**	**0.83**	**0.82**
(FP-Text + NMS-Text)	0.80	0.81	0.80
(FP-Text + G-NMS)	0.79	0.79	0.79

C. Test Time Speed of FP-NMS-Text

Test time speed comparison of FP-NMS-Text with other state-of-the-art scene text detection methods is demonstrated in Table II. The value reported is the average *frames per second* (fps) of running our model on ICDAR 2015 dataset. The other models are also run on ICDAR 2015 dataset with the input resolution as reported in Table II, where MS stands for Multi Scale. The result suggests that our model is fast and has comparable speed with the state-of-the-art CNN models.

D. Evaluation of Joint Training of FP-Text and NMS-Text

We evaluate the performance of joint training of PF-Text and NMS-Text in comparison to two other models. The first model is a disjoint system that trains PF-Text and NMS-Text separately, and is denoted by (FP-Text + NMS-Text). The other model uses FP-Text for detection but uses G-NMS for post-processing. This model is denoted by (FP-Text + G-NMS). As observed from Table. III, FP-NMS-Text achieves the best precision, recall, and f-measure when tested on ICDAR 2015 [19] dataset.

IV. CONCLUSION

In this paper we propose an end-to-end trainable deep learning based model (joint system) that directly detects words from a scene image. This model consists of two different components. The first component (FP-Text) is a CNN that detects text regions from a pyramid of feature maps. The other component (NMS-Text) performs fast and accurate NMS of detected text regions. The joint system is fast, simple to train, and achieves state-of-the-art performance on ICDAR 2015 and MSRA-TD500 scene text detection datasets.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. of CVPR*, 2016, pp. 779–788.

Fig. 4. First, second, and third rows illustrate the results of text detection on ICDAR 2015 [19], MSRA-TD500 [20], and our scene text datasets, respectively.

(a) (b) (c) (d)

Fig. 5. Columns (b) and (d) show the effectiveness of FP-NMS-Text in detecting words with large inter-character distance as compared to the detection performance of [5] (columns (a) and (c)) on such words.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. of CVPR*, 2017, pp. 6517–6525.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] T. Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. of CVPR*, 2017, pp. 936–944.

[5] B. Shi, X. Bai, and S. Belongie, "Detecting Oriented Text in Natural Images by Linking Segments," in *Proc. of CVPR*, 2017, pp. 3482–3490.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. of MICCAI*, 2015, pp. 234–241.

[7] Y. Liu and L. Jin, "Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection," in *Proc. of CVPR*, 2017, pp. 3454–3461.

[8] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," in *Proc. of CVPR*, 2017, pp. 2642–2651.

[9] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented Text Detection with Fully Convolutional Networks," in *Proc. of CVPR*, 2016, pp. 4159–4167.

[10] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic Data for Text Localisation in Natural Images," in *Proc. of CVPR*, 2016, pp. 2315–2324.

[11] W. He, X. Y. Zhang, F. Yin, and C. L. Liu, "Deep Direct Regression for Multi-oriented Scene Text Detection," in *Proc. of ICCV*, 2017, pp. 745–753.

[12] S. Tian, S. Lu, and C. Li, "WeText: Scene Text Detection under Weak Supervision," in *Proc. of ICCV*, 2017, pp. 1501–1509.

[13] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting Word Annotations for Character Based Text Detection," in *Proc. of ICCV*, 2017, pp. 4950–4959.

[14] D. He, X. Yang, C. Liang, Z. Zhou, A. G. Ororbia, D. Kifer, and C. L. Giles, "Multi-scale FCN with Cascaded Instance Aware Segmentation for Arbitrary Oriented Word Spotting in the Wild," in *Proc. of CVPR*, 2017, pp. 474–483.

[15] S. Mohanty, T. Dutta, and H. P. Gupta, "Text preserving animation generation using smart device," in *Proc. of ICME*, 2017, pp. 1039–1044.

[16] C. Animesh, S. Mohanty, T. Dutta, and H. P. Gupta, "Fast text detection from single hazy image using smart device," in *Proc. of ICME Workshops*, 2017, pp. 423–428.

[17] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan, "Text Flow: A Unified Text Detection System in Natural Scene Images," in *Proc. of ICCV*, 2015, pp. 4651–4659.

[18] J. H. Hosang, R. Benenson, and B. Schiele, "A Convnet for Non-maximum Suppression," in *Pattern Recognition*. Springer International Publishing, 2016, pp. 192–204.

[19] D. Karatzas et al., "ICDAR 2015 competition on Robust Reading," in *Proc. of ICDAR*, 2015, pp. 1156–1160.

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

[21] Z. Tian, W. H., T. He, P. He, and Y. Qiao, "Detecting Text in Natural Image with Connectionist Text Proposal Network," in *Proc. of ECCV*. Springer, 2016, pp. 56–72.

[22] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene Text Detection via Holistic, Multi-Channel Prediction," *CoRR*, vol. abs/1606.09002, 2016.

[23] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-Orientation Scene Text Detection with Adaptive Clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, 2015.