IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 10, 2019, accepted March 28, 2019, date of current version April 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908933

FTPN: Scene Text Detection With Feature Pyramid Based Text Proposal Network

FAGUI LIU, CHENG CHEN, DIAN GU, AND JINGZHONG ZHENG
School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

Corresponding author: Cheng Chen (c.chen94@foxmail.com)

This work was supported in part by the Engineering and Technology Research Center of Guangdong Province for Logistics Supply Chain and Internet of Things under Project GDDST[2016]176, in part by the Key Laboratory of Cloud Computing for Super-Integration Cloud Computing in Guangdong Province under Project 610245048129, and in part by the Engineering and Technology Research Center of Guangdong Province for Big Data Intelligent Processing under Project GDDST[2013]1513-1-11.

ABSTRACT Scene text detection is to detect the position of a text in the natural scene, the quality of which will directly affect the subsequent text recognition. It plays an important role in fields such as image retrieval and autopilot. How to perform multi-scale and multi-oriented text detection in the scene still remains as a problem. This paper proposes an effective scene text detection method that combines the convolutional neural network (CNN) and recurrent neural network (RNN). In order to better adapt to texts in different scales, feature pyramid networks (FPN) have been applied in the CNN part to extract multi-scale features of the image. We then utilize bidirectional long–short-term memory (Bi-LSTM) to encode these features to make full use of the text sequence characteristics with the outputs as a series of text proposals. The generated proposals are finally linked into a text line through a well-designed text connector, which can be flexibly adapted to any oriented texts. The proposed method is evaluated on three public datasets: ICDAR2013, ICDAR2015, and USTB-SV1K. For ICDAR2013 and USTB-1K, we have reached 92.5% and 62.6% *F*-measure, respectively. Our method has reached 72.8% *F*-measure on the more challenging ICDAR2015 which demonstrates the effectiveness of our method.

INDEX TERMS Scene text detection, multi-orientation, convolutional neural network, recurrent neural network, residual network.

I. INTRODUCTION

Scene text detection is a crucial premise of text recognition. It has been applied in many fields such as image retrieval, machine translation, and autopilot [1]–[3]. In the last decade, many methods [4]–[12] have been proposed to detect and identify text in natural scenes, which have achieved promising results in some fields. However, there continue to be many difficulties in the detection of text. The first on the list is the diversity and variability of natural scene text. Comparing to the text in the document, the text in the natural scene may be multi-scale and multilingual with various shape, orientation, and color. These varied appearances have brought a lot of challenges to the text detection. Scene text appearance background (including signal signs, fences, bricks or grass) second the list. It may possess similar features to text; however, it will definitely disturb the text distinguishing. The next in line is the incomplete text structure caused by the occlusion of external objects, which may lead to potential detection errors. Moreover, the quality of the image cannot be guaranteed due to uncontrollable collection methods. For example, distortion and out of focus may be caused by different shooting angles or distances; noise and shadow may be formed due to illumination from different direction during taking photos.

Traditionally, text detection methods like texture-based methods [5]–[7] that treat the text as a special texture or region-based methods [8]–[12] that extract candidate components are sensitive to rotation and scale changes. The recent development of deep learning-based text detection methods [13]–[19] have embarked some positive trend towards the researches on this. They trained a deep neural network to extract features rather than designing feature extractors manually, which greatly improved the precision of text detection.

To overcome the difficulties mentioned above, we propose a text proposal network based on feature pyramid which

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

combines convolutional neural network (CNN) [20] and recurrent neural network (RNN) for text localization. Different levels' features of convolutional layers are combined into a feature pyramid, which contains features in different resolution. We then use bidirectional Long Short-Term Memory (Bi-LSTM) to encode these features in order to make full use of the text contextual characteristics; meanwhile, avoid mistakes during detecting a single character individually. A well-designed text connector is finally used to connect all the text proposals into text lines in any orientation.

Our contributions are summarized as follows:

1) We propose an end-to-end trainable framework for text detection in the scene. Our framework makes a combination of FPN and Bi-LSTM, which makes full use of multi-scale features and significantly improve the recall rate.
2) A well-designed text connector is adapted to connect the text proposals into text lines in any orientation.
3) We conduct experiments on several public datasets, including ICDAR2013, ICDAR2015, and USTB-SV1K, to prove the superiority of our method over previous ones.

The rest of the paper is organized as follows. In Section II, we review some of the prior proposed methods in this field; Section III introduces the scene text detection method we propose; and in Section IV, we show our experimental details and detection results on multiple public datasets. Section V is the final conclusion and our prospective future work.

II. RELATED WORKS

In the last decade, researchers have proposed a number of methods for text detection in natural scenes. Previous text detection methods can be roughly divided into two categories: one is the traditional bottom-up methods [5]–[12] and the other is the deep learning-based top-down approaches [13]–[19]. Traditional bottom-up methods can be listed as following:

Texture-based approaches [5]–[7] treat the text as special texture structures for processing. They [5] either use a discrete cosine transform (DCT) to treat the directivity and periodicity of a partial image block as texture detection for text localization; or use the Fourier spectrum to estimate the fundamental frequency of the text image and the characteristics of the frequency to localize the text areas [6]; or use a Gabor filter to extract four stroke features of Chinese characters which has transformed Chinese text detection problem into texture classification problems and processed with Support Vector Machine (SVM) [7].

Region-based approaches [8]–[10] extract candidate components, and then the non-text part is removed by filter or classifier. After the candidate is extracted, the processing of the candidate reduces the processing area relative to the texture-based method, and the character of the text is more prominent in the small area so that the contrast between the text portion and the non-text portion is stronger, thus the detection efficiency had a large improvement compared to texture-based methods. The most popular methods like stroke width transform (SWT) [8] is to find the stroke width value of each image pixel and extract text candidate regions of different scales and directions from the complex background. Huang *et al.* [10] used the stroke feature transformation (SFT) which is based on SWT and used an SFT filter to separate and connect components. Neumann and Matas [9] use Maximally Stable Extremal Regions (MSER) to screen out the candidate text regions in the image and then used the classifier to filter out the text regions.

The hybrid approaches [11], [12] are a combination of the texture-based approaches and the region-based approaches that take the advantages of both. Liu *et al.* [11] proposed a hybrid method based on the combination of connected components and texture feature analysis of unknown text region contours. Pan *et al.* [12] designed a text region detector for estimating text confidence and scale information in an image pyramid, which helps to segment candidate text components by local binarization. Text and non-text components are then tagged using conditional random fields (CRF).

However, the main drawbacks of texture-based approaches [5]–[7] are the simple feature construction and low detection accuracy. In practice, a lot of rules and parameter constraints are required to improve the detection accuracy and the validity of the feature, which results from low generalization and robustness of the algorithm. As for region-based approaches [8]–[10], although they can extract text candidate regions of different scales and orientations from a complex background, they still need a series of artificially defined rules for text detection. Even in the hybrid methods [11], [12], which combine the advantages of both, poor adaptability to multi-oriented text and scales changes still remains as a problem.

With the continuous development of deep learning, it shows an increasing advantage in the field of computer vision. Currently, the most popular methods are the top-down approaches [13]–[19] based on CNN. After using deep learning methods, the accuracy of text detection is greatly improved, and researchers are freed from complex feature designing. More abstract and higher-level features have been continuously extracted from the higher level through the convolution levels. Then, these methods use the neural network to classify text and background as well as bounding box regression to achieve the goal of text detection.

Connectionist Text Proposal Network (CTPN) [19] is a famous method for horizontal text detection. It detects a text line in a sequence of fine-scale text proposals directly in convolutional feature maps. Liu and Jin [13] proposed Deep Matching Prior Network (DMPNet) which detects text with tighter quadrangle based on CNNs. It uses a quadrilateral sliding window to increase the recall and a shared Monte-Carlo method to compute the polygonal areas.

In recent studies, the method base on state-of-the-art object detection methods, such as Faster R-CNN [22], Single Shot MultiBox Detector (SSD) [23], and Fully Convolutional Networks (FCN) [24], aroused the attention of researchers.

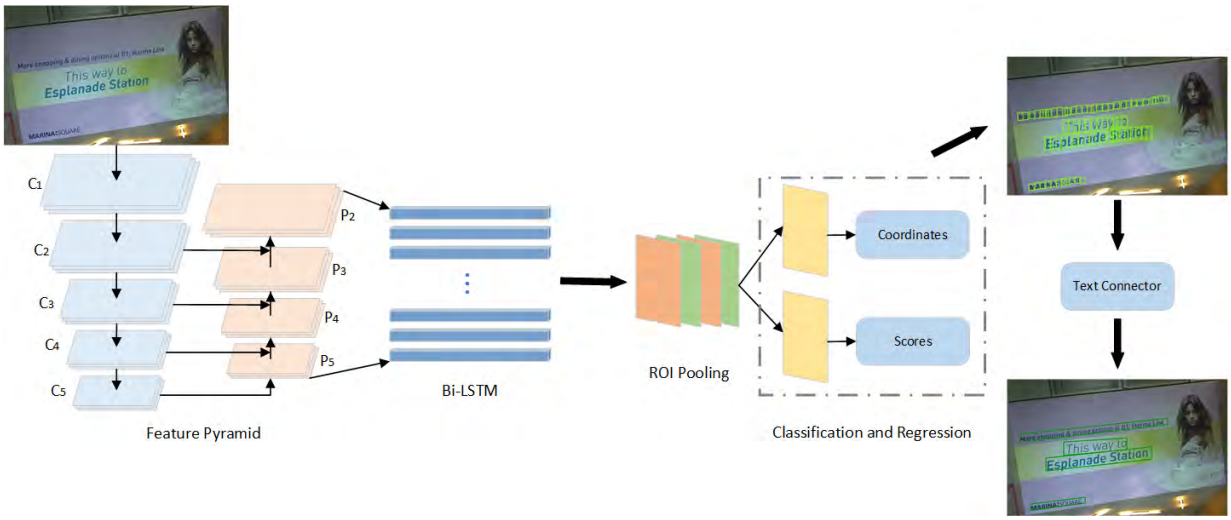

FIGURE 1. The architecture of our work. We first extract features from different layers and consists a feature pyramid. Next, a spatial window slide through the feature maps of the feature pyramid, and the convolutional features of each window are fed into the Bi-LSTM. The Bi-LSTM is connected to a FC layer, which outputs a series of proposals. Then, the RNN part is connected to the ROI Pooling, followed by the output layer, which predicts text/non-text scores and coordinates of the text proposals. Finally, these proposals are connected into text line by a text connector.

Inspired by these techniques, Ma *et al.* [14] proposed a Rotation Region Proposal Networks (RRPN) which incorporated rotation factors into Faster R-CNN. RRPN generates inclined proposals with text orientation angle information. And the Region-of-Interest (RoI) pooling layer of the Faster R-CNN is replaced by a Rotation Region-of-Interest (RRoI) pooling layer which can handle arbitrary-oriented proposals. Liao *et al.* [15] presented another end-to-end trainable fast scene text detector, named TextBoxes, which is based on SSD. There is no post-processing except for a standard non-maximum suppression (NMS) involved in the process, and this is the reason why TextBoxes is so fast. To solve the problem that TextBoxes is not suitable for tilted text detection, they proposed the TextBoxes++ [16] as an improvement, which introduced rotated rectangles or quadrilaterals for detection. Liu *et al.* [30] propose a new framework for arbitrarily oriented scene text detection based on FCN. It detected text center block and word stroke region by two FCN, respectively. A word region surrounding box algorithm make the detections.

There are also many methods that detect and recognize text in an end-to-end trainable model. The Robust text recognizer with Automatic Rectification (RARE) [17] consists of a Spatial Transformer Network (STN) and a Sequence Recognition Network (SRN), which can effectively recognize perspective text and curved text. Besides, Shi *et al.* [18] proposed a Convolutional Recurrent Neural Network (CRNN), which can handle sequences in arbitrary lengths. It integrates feature extraction, sequence modeling, and transcription into a unified framework.

The above methods have made many efforts on multi-scale and multi-oriented text lines, but there are still problems of efficiency or precision. In our work, the use of Feature Pyramid Networks (FPN) [25] can naturally solve the problem of multi-scale text detection without increasing the complexity of calculation. And when solving the problem of multi-oriented text, compared to 5 parameters (rotation detection box) scheme, and 8 parameters (quadrilateral detection box) scheme, there is only 2 required parameters for each anchor in our training process, which greatly reduces the number of parameters. Therefore, our approach is superior.

III. THE PROPOSED METHOD

The architecture of our proposed method is shown in Fig.1. The CNN part is to extract multi-scale features from input images, which consists the feature pyramid. Then we use a spatial window to slide the feature maps of the feature pyramid, and the convolutional features of each window are fed into the RNN part as sequential inputs. The RNN part is applied to encode the contextual information or the text. The internal state of RNN is mapped to the following FC layer and make predictions. Like other two-stage detection methods, we use ROI pooling to make the final predictions and generate a series of text proposals. These text proposals are connected by a text connector which is competent for multi-oriented text.

A. FEATURE EXTRACTION

The previous text detection algorithms [15]–[19] which only use top-level features to make predictions without consideration of the characteristics of other layers do not work well in multi-scale text detection problems. However, using each layer's features for independent prediction indefinitely cause huge computing resources consumption and results in the inevitable use of many superficial and non-robust features.

In our method, FPN serves as the backbone network and consequently solve the above problems. In order to make

full use of its high-resolution and strong semantic features, FPN involves a bottom-up pathway, a top-down pathway, and lateral connections. The bottom-up pathway contains the output of each stage's last residual block of Resnet-101, including conv2, conv3, conv4, and conv5 outputs, denote as $\{C_2C_3C_4C_5\}$. While the top-down pathway upsamples semantically stronger feature maps from higher pyramid levels, to hallucinate higher resolution features. Then, feature maps with the same spatial size from the bottom-up pathway and the top-down pathway are merged by each lateral connection. Finally, we get a set of feature maps, which is marked as $\{P_2P_3P_4P_5\}$.

In the article of FPN, the authors add P_6 into the feature pyramid, where P_6 is simply a stride two subsampling of P_5. And the feature pyramid used for RPN is $\{P_2P_3P_4P_5P_6\}$. However, P_6 is abandoned in our method because of its low resolution for the text detection task. The latter experiment will offer evidence. As a result, we have only adopted the feature $\{P_2P_3P_4P_5\}$.

In previous work, the Visual Geometry Group Network (VGG) [20] is a widely used CNN network. VGG16, in particular, is one of the most popular networks that has achieved ideal performance in object detection tasks. In practice, the FC layers of VGG are removed and we use the output of conv for feature extraction. The output of VGG can be defined as:

$$y = \mathcal{F}(x, W) \tag{1}$$

where x and y denote the input and output vectors of layers and W represents the parameters to be learned in this layer. The function F is the non-linear mapping between x and y. After applying VGG16 to FPN for feature extraction, we try to use a deeper network, VGG19 as an improvement. However, VGG19 could not improve the recognition precision as expected. This problem is caused by degradation, which means the performance of the feature extraction doesn't improve with the increase of network depth. As a result, we tried Deep Residual Network (Resnet) [21] in FPN. Different form VGG, as shown in Fig.2, Resnet uses residual mapping to achieve much deeper convolutional layers. The output of Resnet can be defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{2}$$

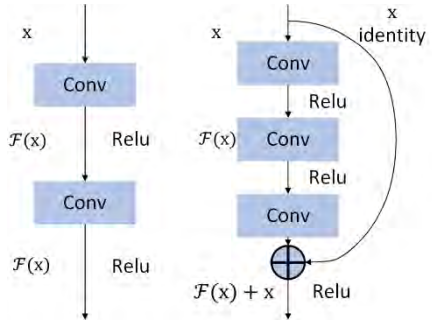

FIGURE 2. The structure of the convolutional layers. The left is the structure in VGG, while the right is the block used in ResNet.

where W_i is the parameters of the *ith* convolutional layers to be learned. The operation $\mathcal{F} + x$ is performed by a shortcut connection and element-wise addition. Comparing with Resnet50 and Resnet152, we used Resnet-101 as the convolutional network, which can achieve a good balance between feature extraction and computing resources.

B. RNN FOR ENCODING

Considering that sequential feature is a significant difference between text and general object detection, which was proved in CRNN experiment, some previous works have been done using RNN to make full use of contextual information so as to reduce false and missing inspections.

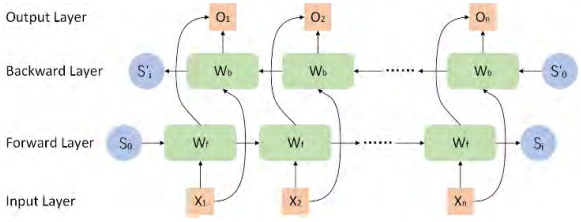

FIGURE 3. The architecture of the Bi-LSTM.

In order to consider both the context before and after the text, we use Bi-LSTM for encoding. As shown in Fig.3, it can encode both the information forward and backward, which fits well with the characteristics of the text line. As shown in Fig.4, we slide each level's feature of feature pyramid from left to right through a $3 \times 3 \times C$ convolution kernel as the sequence input of RNN to cyclically update the internal state:H_t

$$H_t = \varphi(H_{t-1}X_t), t = 1, 2, \ldots, W \tag{3}$$

where $X_t \in R^{3\times3\times C}$is the *tth* sliding window of the feature maps. The number of hidden neurons determines the length of the cell state. The longer the cell state is, the more contextual information can be included, which in theory can obtain higher detection accuracy. But the price comes from a higher amount of calculation. Results show that if the number of hidden neurons in our Bi-LSTM is set to 512 the detection effect on the text will reach an ideal level. In CTPN, all anchors have a horizontal width of 16 pixels, which may cause positional errors, so the author used Side-refinement to eliminate these errors when connecting text proposals. The difference between our method and CTPN is that we use the feature pyramid $P_2P_3P_4P_5$, which has multi-scale features.

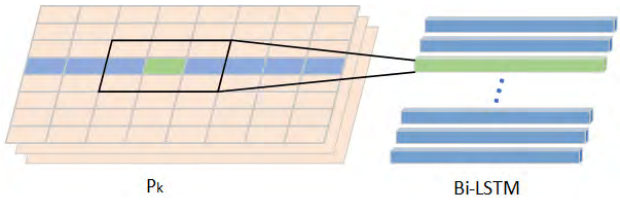

FIGURE 4. The combination of P_k and the Bi-LSTM.

The size of each levels' feature maps is determined by the size of the input image. Taking the input image with 224 × 224 size as an example, the feature pyramid has the feature size 56 × 56, 28 × 28, 14 × 14, and 7 × 7, so the horizontal width of anchors is 4, 8, 16, 32, respectively. Therefore, the position of the direct output has fine position information compare with CTPN. Then we connect the internal state of each level's Bi-LSTM to a FC layer and output a series of region proposals. Finally, we get the proposals in different sizes from different levels.

C. ROI POOLING AND DETECTION

For the reason that massive region proposals will lead to performance problems, we use ROI pooling to extract the characteristics of each proposal. ROI pooling has great benefit to speed up the training and testing process. On the other hand, it can significantly improve not only the text/not-text classification accuracy but also the bounding box regression precision. Following the ROI pooling layer are two FC layers with 1024 neurons and finally connect with the bounding box regression layer and the classification layer, which export text proposals.

D. TEXT CONNECTOR

We get a series of continuous text proposals and we need a text connector to construct our final output. Inspired by CTPN, first, we define a paired neighbor P_j for a proposal P_i as $P_j -> P_i$ if it satisfies the following requirements:

(i) P_j is the nearest to P_i and the distance between them is less than $w_j + w_i$

(ii) P_j and P_i have more than 0.5 vertical overlap

where w_i and w_j is the width of proposal P_i and P_j. And if $P_j -> P_i$ and $P_i -> P_j$, these two proposals are grouped into a pair. The improvement we made is that the text proposals are sequentially connected into a quadrilateral rather than a rectangle. Therefore, the text line can be adapted to any orientation.

So far, we have created an end-to-end text detection model which does well in numerous images that contain textual information or have a lot of background interference.

E. MULTI-TASK LOSS

For the reason that the detection module ends up with text/non-text classification and bounding box regression, its multi-task loss is defined as follows:

$$L\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (4)$$

where $\boldsymbol{i}$ is the index of an anchor in a mini-batch, $\boldsymbol{p_i}$ indicates the predicted probability of the target, $\boldsymbol{p_i^*}$ indicates ground truth (which is 1 or 0 otherwise). $\boldsymbol{t_i}$ and $\boldsymbol{t_i^*}$ indicate the position of the prediction frame and the position of the ground truth box, respectively. The loss of classification is based on Softmax loss, and the regression loss uses smooth L1 loss, which is calculated as follows:

$$L_{cls}(p_i, p_i^*) = -log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (5)$$

$$L_{reg}(t_i, t_i^*) = \sum_{i\in\{y,h\}} smooth_{L_1}(t_i - t_i^*) \quad (6)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (7)$$

We follow the relative predicted coordinates applied in CTPN. Each parameter in the bounding box regression is calculated as follows:

$$t_c = (c_y - c_y^a)/h^a t_h = log(h/h^a) \quad (8)$$

$$t_c^* = (c_y^* - c_y^a)/h^a t_h^* = log(h^*/h^a) \quad (9)$$

where $t = \{\boldsymbol{t_c t_h}\}$ and $\boldsymbol{t^*} = \{\boldsymbol{t_c^* t_h^*}\}$ are relative prediction coordinates and the coordinates of ground truth box, respectively. $\boldsymbol{c_y}$, $\boldsymbol{c_y^a}$, and $\boldsymbol{c_y^*}$ indicate the center coordinates of the predicted, anchor and ground truth box, respectively. While $\boldsymbol{h}$, $\boldsymbol{h^a}$, and $\boldsymbol{h^*}$ are the height of the predicted, anchor and ground truth box, respectively. And the width, as mentioned above, is varied from 4 to 32 pixels, which is based on the size of the feature map.

IV. EXPERIMENTS AND ANALYSIS

A. DETAILS

We implement our approach on Tensorflow [26] which works as our deep learning framework. Since the CNN of the feature extraction part of our proposed method is based on Resnet-101, we use the pre-trained model on ImageNet. The weight of the other new layers in our model is the random weight of a random Gaussian distribution with a mean of 0 and a standard deviation of 0.001. Then we use the training data set to fine-tune our model.

Our proposed text detection model can be used for end-to-end training. To train our model, we use the standard SGD algorithm as our optimizer, setting the learning rate, momentum, weight decay and batch size to 0.0001, 0.9, 0.0005, and 1, respectively. To demonstrate the superiority of our model, we have applied neither online hard example mining (OHEM) for balancing the positive and negative samples, nor random crop for data augmentation. The above training process was trained on an Intel i7 5930K CPU, an Nvidia Titan X GPU (Maxwell) and a 16G RAM host. For training and testing efficiency, we train our model on rescaled training images, whose height no larger than 600 pixels, width no larger than 1200 pixels, and ratio are kept the same. On the ICDAR2015 dataset, the average time for a trained picture is 1.6s while the average time to test a picture is 0.3s. We have performed a total of 150k iterations.

B. LABELS

In order to train the RPN, we need to assign a label to each anchor. The text is the positive anchor and the background

is the negative anchor. The positive and negative labels are defined as follows:

A positive label is defined as:

1) An anchor that has an Intersection over Union (IoU) overlap higher than 0.7 with any ground truth box;
2) An anchor with the highest IoU overlap with a ground truth box.

A Negative label is defined as: An anchor has an IoU overlap less than 0.5 with all ground truth boxes.

C. DATASETS

We evaluate our proposed algorithm on three popular public text detection datasets.

ICDAR2013 [27]: It contains 229 images for training and 233 natural images for testing. The scene text in the images is focused and horizontal, which is the expected input for translator-like applications. The dataset is a subset of ICDAR2011, it follows most of the ICDAR2011's images and modifies some of the annotations.

ICDAR2015 [28]: In contrast to ICDAR2013 dataset which is based on well-captured images, ICDAR2015 dataset focuses on incidental scene text. It contains 1000 images for training and 500 images for testing. Users take those pictures without any specific prior action. Incidental scene text covers another wide range of applications linked to wearable cameras or massive urban captures.

USTB-1K [4]: It contains 500 images for training and 500 images for testing. All of them are captured from Google Street View, therefore these images often have low resolution and exhibited high variability. Besides, this dataset includes a lot of small or blurred texts, which increases the difficulty of detection.

D. EVALUATION PROTOCOL

In order to verify the validity of the model, we use the ICDAR2013 competition criteria, which is based on DetEval [29], including precision, recall, and F-measure. DetEval also judges the quantity and quality of the intersection between the detection frame and the ground truth box. The matching relationship not only considers one-to-one matching case but also the one-to-many and many-to-one matching cases. Precision, recall, and F-measure are defined respectively as follows:

$$Precision = \frac{\sum_i^N \sum_j^{|D^i|} M_D(D_j^i, G^i)}{\sum_i^N |D^i|} \tag{10}$$

$$Recall = \frac{\sum_i^N \sum_j^{|G^i|} M_G(G_j^i, D^i)}{\sum_i^N |G^i|} \tag{11}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

where N is the total pictures in the data set $|\boldsymbol{D^i}|$ and $|\boldsymbol{G^i}|$ are the number of the ***ith*** picture detection boxes and ground truth boxes $\boldsymbol{M_D(D_j^i G^i)}$ and $\boldsymbol{M_G(G_j^i D^i)}$are the matching scores of the detection box $\boldsymbol{D^i}$ and ground truth box $\boldsymbol{G^i}$.If there is one-to-one correspondence between them, $\boldsymbol{M_D(D_j^i G^i)}$ and $\boldsymbol{M_G(G_j^i D^i)}$ are set to 1, for one-to-many correspondence they are set to 0.8, and if there is no match, they are set to 0.

E. RESULTS ON ICDAR2013

We test our method on the ICDAR2013 dataset, which is the benchmark for the well-known ICDAR Robust Reading competition, including complex backgrounds, uneven illumination, and interference characteristics in natural scenes. If the overlap between the two boxes is greater than the threshold, they are considered to be matched. Table 1 shows the comparison of our proposed method and other methods. We achieved 93.2% precision, 91.9% recall, and 92.5% F-measure.

TABLE 1. ICDAR2013 datasets.

Algorithm	Precision	Recall	F-measure	Published
Proposed	**0.932**	**0.919**	**0.925**	-
Liu et al. [30]	0.902	0.863	0.882	2019
Wei et al. [39]	0.873	0.811	0.843	2018
Liao et al. [15]	0.88	0.83	0.85	2017
Tian et al. [31]	0.837	0.840	0.838	2017
Wei et al. [32]	0.835	0.772	0.802	2017
Zhang et al. [33]	0.88	0.78	0.83	2016
He et al. [34]	0.93	0.73	0.82	2016
Zhang et al. [35]	0.88	0.74	0.80	2015
Neumann L et al. [36]	0.818	0.724	0.771	2015
Tian et al. [37]	0.852	0.759	0.803	2015
Yin et al. [38]	0.863	0.683	0.762	2014

Our approach achieves a higher F-measure than other methods. Due to the focus on horizontal text line detection of the dataset, we have adjusted our text line generation strategy to generate only horizontal text lines. The reason why our method achieves better precision and recall than others is that the feature pyramid we use can handle features of different scales well, thus has good adaptability to texts of different scales. As what can be seen from the Fig.5, no matter whether the scale of the text is large or small or there is any

FIGURE 5. Detection examples of the proposed method on the ICDAR 2013 dataset.

FIGURE 6. Sample results of the proposed method for arbitrary orientation text detection on USTB-SV1K dataset.

FIGURE 7. Some successful detection results in challenging cases on ICDAR2015 dataset.

interference from the background, our method can detect the text successfully.

F. RESULTS ON USTB-SV1K

USTB-SV1K dataset is a more specific dataset compared with ICDAR2013. Images are captured by video cameras with multi-orientation and multi-view texts, which are distorted and the backgrounds are more complex. Besides, there are lots of long text lines that have extreme aspect ratio. Despite, our method still shows a good performance on this dataset. Table 2 shows the comparison to previous methods. We have achieved 61.4% precision, 63.8% recall, and 62.6% F-measure. It has shown its great text detection in the multi-directional text detection in complex scene. The experiment also shows that for long text area detection, our method has better adaptability and less missing detection. Our approach is the most effective among all listed algorithms.

G. RESULTS ON ICDAR2015

In order to verify the effectiveness and robustness of our method in multi-oriented text detection tasks, we also test on multi-orientation text datasets using the well-known ICDAR2015 dataset. Unlike the ICDAR2013 dataset, the judging criteria use a quadrilateral box to test the model's ability for detecting multi-orientation text. In addi-

TABLE 2. USTB-SV1K datasets.

Algorithm	Precision	Recall	F-measure	Published
Proposed	0.614	**0.638**	**0.626**	-
Liu et al. [30]	**0.723**	0.503	0.593	2019
Wei et al. [39]	0.541	0.559	0.550	2018
Tian et al. [31]	0.838	0.488	0.512	2017
Yin et al. [4]	0.499	0.454	0.475	2015
Yin et al. [38]	0.450	0.452	0.451	2014
Yao et al. [40]	0.458	0.441	0.449	2012

TABLE 3. ICDAR2015 datasets.

Algorithm	Precision	Recall	F-measure	Published
Proposed	0.682	**0.780**	**0.728**	-
Liu et al. [13]	**0.732**	0.682	0.706	2017
Tian et al. [19]	0.52	0.74	0.61	2016
Zhang et al. [33]	0.43	0.71	0.54	2016
Yao et al. [41]	0.724	0.570	0.638	2015
StradVision2[28]	0.37	0.77	0.50	2015
StradVision1[28]	0.46	0.53	0.50	2015
CASIA USTB[28]	0.40	0.62	0.48	2015

tion, the background of ICDAR2015 is more complicated, and there are more interference factors such as illumination and out of focus. Therefore, this data set poses a great challenge to the text detection model. From Table 3 we can see that we have achieved 68.2% precision, 78% recall, and 72.8% F-measure. Among them, the recall is much higher than other methods.

However, as for the precision, there is still a gap between our approach and the state-of-the-art methods. This is because the text is labeled in word level in this dataset, while we have a text line as an output, which may contain several words; therefore, it has introduced some errors in the final detections.

H. CONTRIBUTIONS OF DIFFERENT LAYERS

In the original FPN, the feature pyramid contains $\{P_2P_3P_4P_5P_6\}$ and uses three ratio detection anchors of {1:2, 1:1, 2:1} in different feature layers. These settings of the FPN have achieved good results in the fields of target detection. However, for text detections, the extraction efficacy of some feature layers may not be obvious, so some of the layers in feature pyramid can be abandoned. As shown in Table 4, there is no significant change in the effect of text detection after P_6's removal, because the size of the P_6 is too small for text detection, we do not need to use information that deep. After the removal of P_5, we find that the effect of text detection has dropped significantly. It's obvious that P_5 plays an important role in the model. Then, we have tried the other extreme by removing P_2. The effect of the model also

TABLE 4. F-MEASURE with different Feature.

Layers	ICDAR2013	ICDAR2015	USTB-SV1K
$\{P_2, P_3, P_4\}$	0.909	0.710	0.612
$\{P_3, P_4, P_5\}$	0.901	0.695	0.603
$\{P_2, P_3, P_4, P_5\}$	**0.925**	**0.728**	**0.626**
$\{P_2, P_3, P_4, P_5, P_6\}$	**0.925**	0.726	**0.626**

TABLE 5. F-MEASURE with different hidden neurons.

Hidden neurons	ICDAR2013	ICDAR2015	USTB-SV1K
128	0.903	0.711	0.590
256	0.915	0.716	0.598
512	**0.925**	**0.728**	0.626
1024	0.922	0.727	**0.629**

shows a significant decline. From the above analysis, we are already aware that P_2 is obtained by P_3 upsampling, which corresponds to a larger feature map. It provides more accurate position information and has stronger semantic information. Therefore, it plays an indispensable role.

I. RESULTS COMPARISON WITH DIFFERENT HIDDEN NEURONS

The number of hidden neurons determines the length of the cell state. In previous practices, the longer the cell state, the more context information can RNN contain, meanwhile, the higher precision can a model achieve at the cost of more computing resource it will need. In order to pick a suitable amount for hidden neurons, we set the hidden neurons to 128, 256, 512, and 1024 for Bi-LSTM respectively, then evaluate it in several test sets. The comparison is shown in Table 5. The detection precision increases with the increase of the hidden neurons. However, when the number of hidden neurons increases from 512 to 1024, the improvement is not significant. As a result, it achieves the best balanced when we set the number of hidden neurons to 512.

V. CONCLUSION AND FUTURE WORKS

In this paper, we introduced a text detection model that combines CNN and RNN. We make full use of the feature pyramid in FPN to extract features on multi-scales and use the sequence of text to generate a series of text proposals using Bi-LSTM. Finally, we connect them through a text connector. It can adapt to multi-oriented and multi-scale scene text detection tasks and achieves ideal results on multiple public datasets. However, there is still room for improvement for the model. Firstly, as the text connector is designed for the text proposal with a special strategy, our method is not suitable for curve text. We plan to further improve the connection method to accommodate this. Moreover, our model is designed to make full use of the sequence characteristics of text; therefore, we mainly use English and digital datasets to train and validate our method. We plan to improve our model and challenge more multi-language datasets, such as Chinese-English mixed detection datasets, in the future.

REFERENCES

[1] S. Karaoglu, R. Tao, T. Gevers, and A. W. M. Smeulders, "Words matter: Scene text for image classification and retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1063–1076, May 2017. doi: 10.1109/tmm.2016.2638622.

[2] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 19–36, 2015. doi: 10.1007/s11704-015-4488-0.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CS, USA, 2012, pp. 1097–1105.

[4] 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, Sep. 2015. doi: 10.1109/tpami.2014.2388210.

[5] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385–392, Apr. 2000. doi: 10.1109/34.845381.

[6] B.-K. Sin, S.-K. Kim, and B.-J. Cho, "Locating characters in scene images using frequency features," in *Proc. Object Recognit. Supported Interact. Service Robots*, Quebec City, QC, Canada, Aug. 2002, pp. 489–492.

[7] J. Yan, J. Li, and X. Gao, "Chinese text location under complex background using Gabor filter and SVM," *Neurocomputing*, vol. 74, no. 17, pp. 2998–3008, 2011. doi: 10.1016/j.neucom.2011.04.031.

[8] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 2963–2970.

[9] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 770–783.

[10] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1241–1248.

[11] Y. Liu, S. Goto, and T. Ikenaga, "A contour-based robust algorithm for text detection in color images," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 1221–1230, 2006. doi: 10.1093/ietisy/e89-d.3.1221.

[12] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011. doi: 10.1109/tip.2010.2070803.

[13] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3454–3461.

[14] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[15] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. AAAI*, 2017, pp. 4161–4167.

[16] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[17] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4168–4176.

[18] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[19] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 56–72.

[20] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[22] 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, Jun. 2017. doi: 10.1109/tpami.2016.2577031.

[23] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[26] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," *Proc. OSDI*, vol. 16, 2016, pp. 265–283.

[27] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Washington, DC, USA, Aug. 2013, pp. 1484–1493.

[28] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, Tunis, Tunisia, Aug. 2015, pp. 1156–1160.

[29] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *Int. J. Document Anal. Recognit.*, vol. 8, no. 4, pp. 280–296, 2006. doi: 10.1007/s10032-006-0014-0.

[30] Z. Liu, W. Zhou, and H. Li, "Scene text detection with fully convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–23, Jan. 2019. doi: 10.1007/s11042-019-7177-4.

[31] C. Tian, Y. Xia, X. Zhang, and X. Gao, "Natural scene text detection with MC–MR candidate extraction and coarse-to-fine filtering," *Neurocomputing*, vol. 260, pp. 112–122, Oct. 2017. doi: 10.1016/j.neucom.2017.03.078.

[32] Y. Wei, Z. Zhang, W. Shen, D. Zeng, M. Fang, and S. Zhou, "Text detection in scene images based on exhaustive segmentation," *Signal Process., Image Commun.*, vol. 50, pp. 1–8, Feb. 2017. doi: 10.1016/j.image.2016.10.003.

[33] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4159–4167.

[34] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2529–2541, Jun. 2016. doi: 10.1109/tip.2016.2547588.

[35] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2558–2567.

[36] L. Neumann and J. Matas, "Efficient scene text localization and recognition with local character refinement," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, Aug. 2015, pp. 746–750.

[37] 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. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4651–4659.

[38] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014. doi: 10.1109/tpami.2013.182.

[39] Y. Wei, W. Shen, D. Zeng, L. Ye, and Z. Zhang, "Multi-oriented text detection from natural scene images based on a CNN and pruning non-adjacent graph edges," *Signal Process., Image Commun.*, vol. 64, pp. 89–98, May 2018. doi: 10.1016/j.image.2018.02.016.

[40] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.

[41] C. Yao *et al.* (2015). "Incidental scene text understanding: Recent progresses on ICDAR 2015 robust reading competition challenge 4." [Online]. Available: https://arxiv.org/abs/1511.09207

FAGUI LIU received the M.S. degree from Beihang University, in 1991, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2006, where she is currently a Professor with the School of Computer Science and Engineering. Her research interests include service computing, the Internet of Things, cloud computing, and big data.

CHENG CHEN received the B.S. degree from Northwestern Polytechnical University, China, in 2017. He is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include deep learning networks and computer vision.

DIAN GU received the B.S. degree from Jinan University, Zhuhai, China, in 2018. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Her research interests include machine learning and object detection.

JINGZHONG ZHENG received the B.S. degree from North China Electric Power University, China, in 2017. He is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include neural networks, natural language processing, and sentiment analysis.