IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2019, accepted May 25, 2019, date of publication May 30, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919994

A Detection and Verification Model Based on SSD and Encoder-Decoder Network for Scene Text Detection

XUE GAO[1,2], (Member, IEEE), SIYI HAN[1], AND CONG LUO[1]
[1]School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China
[2]SCUT-Zhuhai Institute of Modern Industrial Innovation, Zhuhai 519000, China

Corresponding author: Xue Gao (xuegao@scut.edu.cn)

This work was supported in part by the Natural Science Foundation of Guangdong Province under Grant 2015A030313210, and in part by the Science and Technology Program of Guangzhou under Grant 201604010061 and Grant 201707010141.

ABSTRACT Text detection in natural scene image is challenging due to text variation in size, orientation, color and complex background, contrast, and resolution. In this paper, we focus on the long text detection in complex background. In order to deal with multi-scale text variation and exploit the recognition result to enhance the detection performance, we propose a detection and verification model based on SSD and encoder-decoder network for scene text detection. First, we present a text localization neural network based on SSD, which incorporates a text detection layer into the standard SSD model and can detect horizontal texts, especially long and dense Chinese texts in natural scenes more effectively. Second, a text verification model based on the encoder-decoder network is designed to recognize and verify the initial detection results, in order to eliminate non-text areas that are falsely detected as text areas. A series of experiments have been conducted on our constructed horizontal text detection dataset, which is composed of the horizontal text images in ICDAR 2017 Competition on Reading Chinese Text in the Wild (RCTW 2017) and some scene images taken by cameras. Compared with previous approaches, experimental results show that our method has achieved the highest recall rate of 0.784 and competitive precision rate in text detection, indicating the effectiveness of our proposed method.

INDEX TERMS Scene text detection, SSD, encoder-decoder network, verification model.

I. INTRODUCTION

Text in natural scene image contains rich semantic information and is of great value for image understanding. For example, the texts embedded in packaging can promote the commodity value, and the texts on street signs can improve the precision of automatic navigation. The recognition and understanding of text information in natural scenes is gradually becoming one of the hotspots of research and application in recent years.

As the first step in text reading systems, text detection, which aims at localizing text areas with bounding boxes of words, plays a critical role in text information extraction and understanding. Although there already exist some OCR systems that can detect and recognize texts in formatted documents, text detection in natural scene image is still a challenge due to text variation in size, orientation, color and complex background, contrast, resolution, etc. In this paper, we focus on the long text detection in complex background.

Traditional text detection methods are usually based on sliding window [1]–[3] or connected component extraction [4]–[7]. The text detection methods based on sliding window detect texts by traversing every area of the scene image, which can achieve a high recall rate, however with a high computational complexity. The other text detection methods based on connected component extraction firstly extract candidate texts from the input image with a low computational complexity, but the accurate location of text areas depends on a series of complex post-processing such as filtering and fusion of candidate texts. Driven by the rapid development of deep learning, text detection methods based on deep learning are also becoming popular. While texts in natural scenes can be regarded as a kind of specific objects, it is easy to detect horizontal texts in natural scene images

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

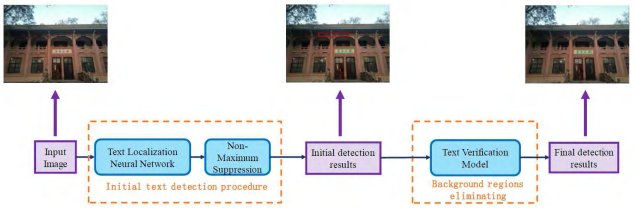


FIGURE 1. Model overview. The whole procedure mainly takes two steps: (1) Initial text detection with text localization neural network. (2) Eliminating background regions with text verification model.

with the help of deep object detection network. However, different from general objects, horizontal scene texts tend to have more scales, and be distributed widely in any region of the scene image, and easier to be disturbed by the background of similar texts. Therefore, directly applying object detection network for text detection is not an effective method.

To this end, we propose in this work a novel method for natural scene horizontal text detection, which is based on Single Shot MultiBox Detector (SSD) [8] and encoder-decoder network. The part of this work was presented at the International Workshop on Deep Learning for Pattern Recognition (DLPR 2018) [9]. An overview of the network architecture is presented in Fig. 1.

The whole network mainly contains two parts: Text Localization Neural Network and Text Verification Model. We present a text localization neural network based on SSD, which is designed for initial text candidate localization with single forward pass as well as a standard Non-Maximum Suppression (NMS), avoiding a series of complex intermediate steps such as text candidate regions filtering and fusion. Also, aiming at the problem that some background areas are misjudged as texts by the text localization neural network, we propose a text verification model based on encoder-decoder network, which further improves the precision of text detection by using recognition results to refine initial detection results and eliminate the non-text areas. The encoder network, which is composed of CNN model and BiGRU network, encodes the input text image features with rich context information. And the decoder network adopts GRU network with attention mechanism to decode the encoded feature vector sequences into words, and better models the concerns of decoder network at current moment.

Specifically, in order to verify the validity of the proposed method in this paper, we construct a text detection dataset composed of images taken by cameras in natural scenes and the horizontal text images from RCTW 2017 [10] dataset, which contains 12k training images and 3k test images. Compared with other approaches, experimental results on our dataset show that our proposed method achieves a highest recall rate of 78.4% and a competitive precision rate 83.0%, which is second only to CTPN [11]. In detail, compared with SSD model, the text localization neural network improves the precision from 75.9% to 80.5%, and the recall rate from 56.8% to 78.4%. Moreover, the text verification model improves the detection precision by 3.1%.

In summary, the main contributions of this paper are thus two-fold:

(1) We propose a Text Localization Neural Network based on SSD, which improves the SSD model according to the characteristics of scene texts, so that it can better adapt to natural scene text detection. The network directly predicts text areas of the input images and obtains candidate text bounding boxes through a standard non-maximum suppression process, eliminating several time-consuming intermediate stages.

(2) A Text Verification Model based on encoder-decoder network is designed. The model recognizes the texts in candidate bounding boxes detected by the text localization neural network, and eliminates non-text areas that are falsely detected as text areas, thereby further improving the precision of natural scene text detection.

The remainder of this paper is organized as follows: In Sec.2, we briefly review the previously related work. In Sec.3, we describe the proposed method in detail, including Text Localization Neural Network and Text Verification Model. Experimental results are presented in Sec.4. Finally, some conclusion remarks and future works are given in Sec.5.

II. RELATED WORK

Text localization and recognition in natural scene images are derived from the analysis and recognition of scanned documents and images in traditional research [12]. With the rapid development of computer vision technology, text detection and recognition tasks are no longer two independent subtasks, but tend to be realized through end-to-end systems [3]. Detecting text in natural scenes has been a hot research topic in the field of computer vision, and plenty of excellent works and effective strategies have been proposed.

Previous works on text detection can be roughly divided into two categories, one is the traditional method based on sliding window [1]–[3] or connected component extraction [4]–[7] and the other is the method based on deep learning in recent years. Sliding window based methods adopt the sliding window to look over the whole image and extract the features in the window. Then the pre-trained classifier is used to classify the character/non-character in the window area. Finally, the candidate characters are merged to obtain the final text area. Kim *et al.* [1] took the pixel value of the original image as the input of SVM classifier, trained the text/non-text classifier according to the labeled training data, then performed connected domain texture analysis on the candidate text areas discriminated by SVM classifier, and obtained the final text areas. Pan *et al.* [2] combined LBP [13] features with HOG [14] features, extracted candidate character regions using the cascaded Adaboost [15] classifier, and further fused the character regions to obtain the final text areas. Connected component extraction based methods mainly decide on how to design an effective algorithm for text candidates extraction. Stroke Width Transform (SWT) [4] and Maximally Stable Extremal Regions (MSER) [7]

are two representable algorithms with leading performance in ICDAR 2011 [16] and ICDAR 2013 [17]. Yao *et al.* [6] sought candidate texts with the help of SWT, and designed a multi-oriented text detection algorithm combined with region color and shape features. Sun *et al.* [18], Lei *et al.* [19] proposed color enhancement extremal regions based on MSER for candidate texts generation.

In recent years, with the rapid development of deep learning, text detection algorithms based on deep learning have become the mainstream in the field of text detection. According to the shape of the text bounding boxes to be detected in scene images, the methods of detecting horizontal texts [11], [20], [21] and multi-oriented texts [22], [23] are derived.

Deep learning technologies have advanced performance of text detection in the past years. A technique similar to text detection is general object detection. In the object detection networks based on CNN, on the one hand, the network relies on the region extraction, such as R-CNN [24], Fast R-CNN [25] and Faster R-CNN [26]. On the other hand, SSD [8] and YOLO [27] directly predict the location of objects. Owing to the rapid development of deep object detection networks, horizontal scene text detection can be realized based on those networks, in which the text is regarded as a special kind of objects. Huang *et al.* [20] firstly sought candidate texts via MSER, CNN was then used to classify text/non-text regions. Based on Faster R-CNN [26], DeepText [21] proposed Inception-RPN and made further optimization to adapt text detection. Tian *et al.* [11] designed a network called Connectionist Text Proposal Network (CTPN), which combined CNN and LSTM to detect the text line by predicting a sequence of fine-scale text components. Inspired by YOLO [27], Gupta *et al.* [28] proposed a fully convolutional regression network, which made predictions through a single forward pass.

Multi-oriented text detection task can be regarded as a special pixel-level image segmentation problem. Multi-oriented text detection based on deep learning is usually implemented by FCN [22]. Zhang *et al.* [23] made use of FCN to train and predict the saliency map of text areas, then combined the saliency map and text elements to estimate the line where the text is located, and used another full convolution model classifier to estimate the center of each character, thus eliminated false detection areas. EAST [29] adopted FCN network to output feature layers and generate geometric text boxes in multiple channels of image. By introducing two geometric shapes, RBOX and QUAD, it can detect multi-oriented texts in scene images.

Our work is mainly inspired by recent work [8]. Similar to SSD, we utilize multiple feature layers for text detection. In addition, we introduce Text Detection Layers and three improved strategies for default boxes generation, to better detect long text lines. Also, we present a Text Verification Model based on encoder-decoder network, to eliminate non-text areas by recognizing the initial detection results, which further improves the precision of text detection.

III. METHODOLOGY

In this section, we describe the proposed method in detail. First, a text localization neural network is designed for initial text detection, the key components of that are text detection layers and three improved strategies. Then, false localized text regions are eliminated by the text verification model, which is based on encoder-decoder network.

A. TEXT LOCALIZATION NEURAL NETWORK

1) NETWORK DESIGN

The structure of our text localization neural network is illustrated in Fig. 2. The proposed network is a 28-layer fully convolutional network including two parts, one consists of several convolution layers and pooling layers, corresponding to the first 13-layer base network in Fig. 2. It keeps the conv1_1 to conv5_3 layers in VGG-16 [30] and the last two fully connected layers in VGG-16 that are replaced with conv6 and conv7 respectively. The other part is 15 extra convolution layers, including 9 convolution layers, inserted after conv7 (corresponding to the conv8 to conv11 in Fig. 2), and 6 text detection layers responsible for outputting the predicted default text bounding boxes. The text detection layer is the text localization model designed in this paper, which can detect the core of different scale text areas. As shown in Fig. 2, 6 text detection layers are inserted after 6 different convolution layers (conv4_3, conv7, conv8_2, conv9_2, conv10_2 and conv11_2).

After generating several predicted text bounding boxes via the text detection layer, the text localization neural network goes through a non-maximum suppression (NMS) process, eliminates the redundant bounding boxes, and obtains the final detection results. One important point is that our proposed network does not need complex post-processing, such as text areas filtering and fusion, and the detection process is more compact and efficient.

2) TEXT DETECTION LAYERS

Similar to original SSD model, text detection layers make use of multiple feature maps to predict default text bounding boxes. The working principle of text detection layers is illustrated in Fig. 3. For each position of a feature map cell, a 84 dimensional vector is outputted via the convolution filter. That is to say, among the $14(2 \times 7)$ default boxes responsible for predicting text areas, each default box produces a 6 dimensional output, which contains a 4 dimensional vector for location information (center point coordinates and width and height of the predicted bounding box) and a 2 dimensional vector for text/non-text confidence.

As shown in Fig. 4 (a), each pixel in the convolution feature map is mapped back to original image, corresponding to a cell. The default text bounding box is a rectangular bounding box, generated by taking the center of the cell as center point coordinates. Then, the rectangular bounding box is responsible for predicting a matched ground truth text box. We assume that the size of input image is $w_I \times h_I$, and

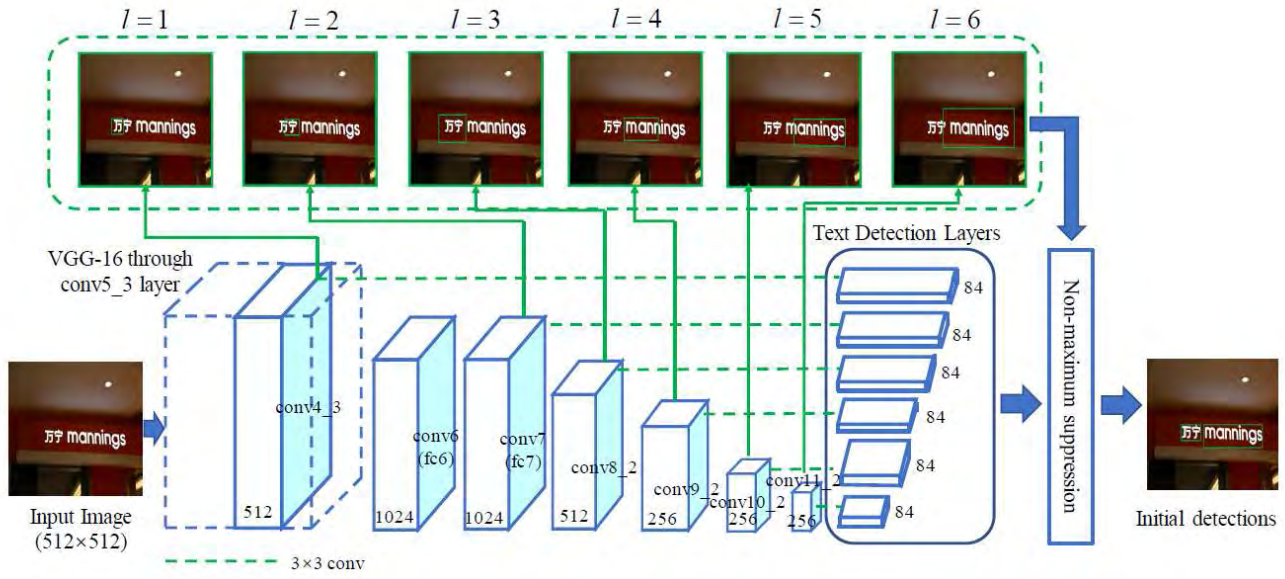


FIGURE 2. Structure of text localization neural network. The network is a 28-layer fully convolutional network with two parts, one is a 13-layer base network inherited from VGG-16, and another part is composed of 15 extra convolution layers. Six extra text detection layers are responsible for prediction of scene text. Green rectangular boxes are examples of size increasing default text bounding boxes among different feature layers.

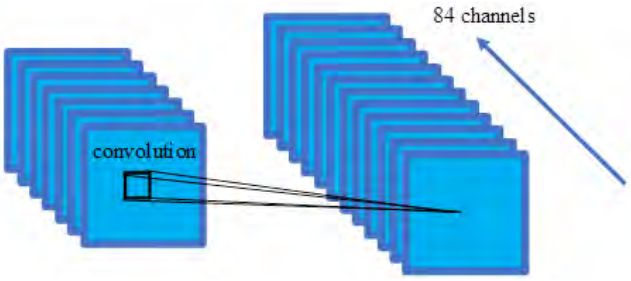


FIGURE 3. Working principle of text detection layers.

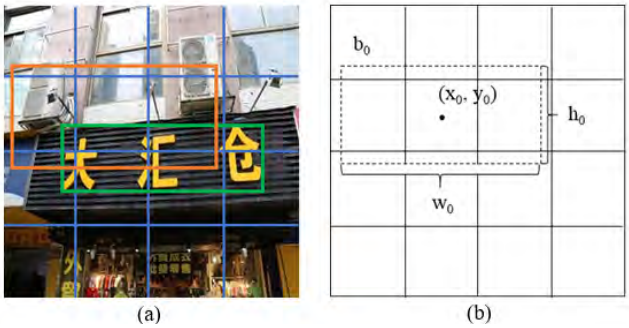


FIGURE 4. An example of default text bounding boxes generated by a 4 × 4 feature map cell. The orange box in (a) is a default text bounding box responsible for predicting the green ground truth text box, corresponding to the dotted box b_0 in (b).

the size of feature map used in prediction is $w_{map} \times h_{map}$. As shown in Fig. 4 (b), the pixel point with coordinate (i, j) in feature map is mapped back to original image, corresponding to a horizontal default text bounding box b_0. Moreover, b_0 is represented by (x_0,y_0,w_0,h_0), where (x_0,y_0) represents the center point coordinates of default box, w_0 and h_0 represent the width and height of default box, respectively.

The input image goes through a forward propagation. The text detection layer predicts $(\Delta x,\Delta y,\Delta w,\Delta h,c_0,c_1)$, and generates a predicted box $b = (x, y, w, h)$. The predicted bounding box also contains text/non-text confidence c_0 and c_1.

3) DEFAULT TEXT BOUNDING BOXES GENERATION

In order to better adapt to the distribution characteristics of Chinese text areas in natural scene, we propose two improved strategies for default boxes generation in text detection layers.

a: LARGER ASPECT RATIOS

While predicting, the text detection layer combines feature map information of different layers. Then, each position of the feature map is mapped back to original image, and we will obtain a fixed-size feature map cell. In this paper, we preset a base default box with the same width and height in the cell center, and the base default box will produce a series of boxes with different width and height, according to the defined aspect ratios. Assuming that the text detection layer uses m feature maps for prediction, for the k-th feature map, the size of base default box is defined as:

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m - 1}(k - 1), \quad k \in [1, m] \tag{1}$$

Assuming that the aspect ratio of default box is a_r, for each position of the k-th feature map cell, the width and height of the corresponding default box in original image are defined as:

$$w_k^a = S_k\sqrt{a_r} \tag{2}$$

$$h_k^a = S_k/\sqrt{a_r} \tag{3}$$

For object detection, SSD sets three different aspect ratios for object default boxes $a_r = (1, 2, 3)$. However, the

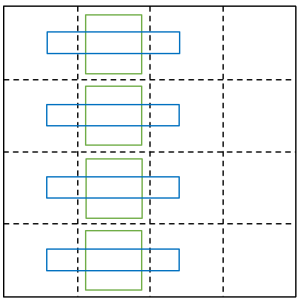


FIGURE 5. Examples of default text bounding boxes in a 4 × 4 feature map cell. Default boxes locate at the center of a feature map cell. The green box is base default box, corresponding $a_r = 1$. The blue box is a generated default box, corresponding $a_r = 5$.

horizontal scene texts, especially Chinese texts, are different from general objects, and tend to have more scales and larger aspect ratios. Therefore, on the basis of original a_r of SSD, we define 4 more aspect ratios (5,7,9,10) for default boxes to better adapt to the horizontal text detection.

For the k-th feature map, when $a_r \neq 1$, according to (2) and (3), every feature map cell will generate two default boxes based on each a_r. Also, for 7 different aspect ratios, every feature map cell will generate 14(2×7) default boxes. When $a_r = 1$, every feature map cell will generate a base default box with the same width and height. At this point, we adopt the same strategy as original SSD model, i.e., adding a square default box with width and height of $\sqrt{S_k S_{k+1}}$. As shown in Fig. 5, default text bounding boxes with larger aspect ratios are illustrated. Note that for simplicity, only aspect ratios 1 and 5 are plotted.

b: DEFAULT BOXES WITH VERTICAL OFFSETS

Different from the fixed locations of general objects in natural scene images, scene texts may be densely distributed in the local area, and the limited default text bounding boxes cannot match all ground truth text boxes. Therefore, we design a method to increase the number of default text bounding boxes. Centering on the feature map cell, we make offsets to default text bounding boxes about half of the height of each feature map cell in vertical direction, in order to generate more default text bounding boxes. Through this improvement, more ground truth text boxes will be exactly matched, and the probability of missed detection is decreased.

As shown in Fig. 6, the green dotted boxes in (a) and (b) are default text bounding boxes in the 3×3 feature map cell. It is obvious that part of text areas in the image cannot be fully covered, so that the locations of them will not be predicted. The yellow dotted boxes in (a) and (b) are obtained by making offsets to the green dotted boxes about half of the height of each feature map cell in vertical direction, from the center of the feature map cell. These yellow dotted boxes can exactly match the text areas that are not covered by green dotted boxes, and predict the locations of these areas.

4) IMPROVED CONVOLUTION FILTER

Our text localization neural network obtains the output of text detection layers via the convolution filter based on different

FIGURE 6. Vertical offsets to default text bounding boxes in 3 × 3 feature map cell. Through vertical offsets, yellow bounding boxes can be produced, and more text areas will be able to be covered.

FIGURE 7. Comparison examples of convolution kernel sizes 3 × 3 and 1 × 5.

level feature maps. Different from SSD, we draw lessons from the design of the Inception module of Pan et al. in SSTD [31] (Single Shot Text Detector), replacing the original 3×3 convolution kernel with a 1×5 rectangular convolution kernel. As shown in Fig. 7, the green box is the ground truth text box, and the red box is the default text bounding box, which is responsible for predicting the location and confidence of text in the ground truth text box. Since the horizontal text in scene images usually has a larger aspect ratio, the 3×3 regular convolution kernel (shown as the blue box in (a)) does not fully match the features of long texts. Therefore, it is possible to extract non-text features, and bring unnecessary errors to the text/non-text classification task of default boxes. Unlike the setting of 3×3 convolution kernel, the 1×5 convolution kernel (shown as the blue box in (b)) will extract the features in horizontal text areas more effectively, reducing the risk of bringing in the background noise. Therefore, the accuracy of the classification task can be improved.

5) LOSS FUNCTION

In the text localization neural network, the default text bounding box is responsible for predicting the location and confidence of text areas. Therefore, the loss function of the network training is a joint loss function of classification loss and regression loss. In this paper, the loss function for initial text detection uses the form of multi-task loss [8], which is defined as:

$$L(x, c, b, g) = \frac{1}{N}\left(L_{conf}(x, c) + \alpha L_{loc}(x, b, g)\right) \quad (4)$$

where N is the matched numbers of default text bounding boxes to ground truth text boxes, α is the weight term. If $N = 0$, we set the loss $L = 0$. Different from SSD, our text localization neural network only predicts text areas and outputs the confidence of text/non-text. Thus, L_{conf} is 2-softmax loss produced by classification, and we define the loss as:

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij} \log\left(\hat{c}_i^1\right) - \sum_{i \in Neg} \log\left(\hat{c}_i^0\right) \quad (5)$$

Let $x_{ij} = \{1, 0\}$ be an indicator for matching the i-th default text bounding box to the j-th ground truth text box. If it is matched, $x_{ij} = 1$, i.e. when the network is training, the default box can be used as a positive sample of the prediction task; otherwise, $x_{ij} = 0$, the default box is a negative sample. Moreover, c_i^1 is the confidence of that the predicted text box contains text areas, and c_i^0 is the confidence of that the predicted text box contains background. *Pos* is the positive sample set matched with the ground truth text boxes, and *Neg* is the negative sample set.

L_{loc} is the regression loss generated by a default box regressing to corresponding ground truth text box. We use the predicted text box $b = (cx, cy, w, h)$ to regress the matched ground truth text box g, and adopt the smooth-L_1 loss function with corresponding parameters. Where (cx, cy) is the center point coordinate of b, w and h are respectively the width and height of b. We define the regression loss as:

$$L_{loc}(x, b, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij} smooth_{L_1}\left(b_i^m - \hat{g}_j^m\right) \quad (6)$$

$$\begin{aligned} \hat{g}_j^{cx} &= \frac{g_j^{cx} - d_i^{cx}}{d_i^w} \\ \hat{g}_j^{cy} &= \frac{g_j^{cy} - d_i^{cy}}{d_i^h} \\ \hat{g}_j^{w} &= \log\left(\frac{g_j^w}{d_i^w}\right) \\ \hat{g}_j^{h} &= \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned} \quad (7)$$

where the smooth-L_1 loss function is defined as:

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geq 1 \end{cases} \quad (8)$$

B. TEXT VERIFICATION MODEL

1) OVERVIEW

Encoder-decoder network is a method of organizing recurrent neural networks to deal with the prediction problem of sequence-to-sequence. It was first proposed by Dzmitry [32] to solve the problem of machine translation, and was widely used in machine translation systems. Similar to machine translation, which converts the input word sequences that are source language into the output word sequences that are target language, text recognition can also be viewed as 'translating' image data into character sequences. Thus the encoder-decoder network can also be used in text recognition.

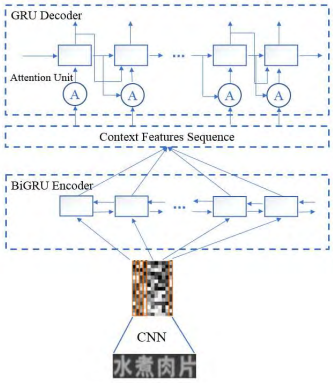


FIGURE 8. Architecture of text verification model.

Natural scene images usually have complex background, as well as rich text and non-text styles, so that some non-text areas are misjudged as text areas by text localization network. To address the above issues, we propose text verification model based on the encoder-decoder network. The model uses the encoder network to encode text areas of the input image, and translates the encoded features into text sequences through the decoder network. Via recognizing the texts in candidate text boxes outputted from the text localization network, the model further eliminates the non-text areas.

Architecture of our text verification model is depicted in Fig. 8. The whole model is based on encoder-decoder network, and makes use of text recognition results to refine the detection. The encoder network is composed of CNN and BiGRU network. It encodes the input image and obtains an encoded feature vector sequence, which combines the context features of the image. The decoder network adopts GRU network to decode the feature vectors. Meanwhile, we introduce attention mechanism in the decoding process, to better model the concerns of decoder network at different times. When training, the text verification model maximizes the probability of generating labeling word sequences in input images. Via correctly recognizing text areas in candidate text boxes generated by text localization network, the model further eliminates the background areas that are misjudged as texts.

2) ENCODER NETWORK

Our encoder network, which is composed of CNN Model and BiGRU Network, encodes the input image with context information to obtain the encoded feature vector sequence. In detail, a CNN model is used to extract image features. To better learn the dependencies and relationships between characters in input image, we then apply BiGRU to encode these features.

As shown in Fig.9, the input image has a size of 100×32, via the CNN model extracting features, we obtain a feature

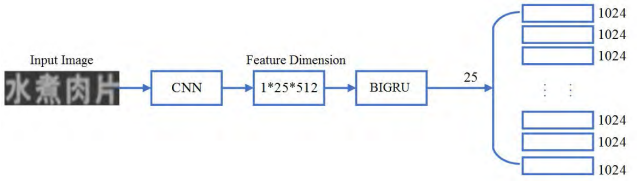


FIGURE 9. Architecture of encoder network.

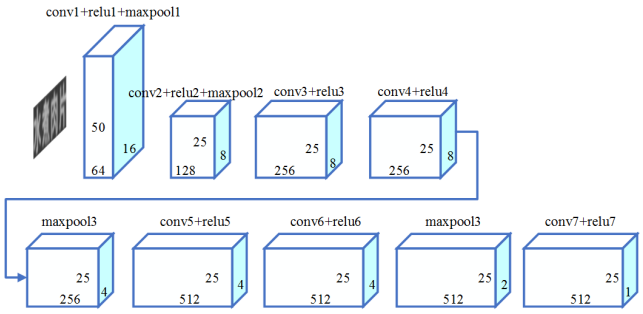


FIGURE 10. Architecture of CNN model.

map output that has a size of 1×25 and 512 channels. Taking the channel as the dimension to stitch features, we further obtain a feature vector sequence of length 25 with 512 dimension of each. Then we use BiGRU to encode the feature vectors. The hidden unit has a size of 512 in BiGRU. For each moment, the two 512 dimensional vectors propagating along positive and negative time sequence respectively are spliced. Therefore we obtain a 1024 dimensional encoded feature vector sequence of length 25.

a: CNN MODEL

Our CNN model adopt the same CNN architecture as in Convolution Recurrent Neural Network (CRNN) [33]. As shown in Fig.10, the CNN model is composed of 18 layers, including 7 convolution layers, 7 activation function layers and 4 max-pooling layers. The last two max-pooling layers adopt a 2×1 rectangular convolution kernel respectively, i.e. the stride_h is 2 and the stride_w is 1. Therefore the length of encoded feature vector sequence will not be compressed and keeps the constant 25. A longer encoded feature vector sequence in a certain range is beneficial for BiGRU network to extract the context features, and further improve the accuracy of following decoding classification.

b: BiGRU NETWORK

As shown in Fig.11, the BiGRU network uses two GRU (Gated Recurrent Unit) [34] units as the basic components. One propagates along positive time sequence and the other propagates along negative time sequence, which is beneficial to capture context information of the input image at the same time. The two GRU units propagating in opposite directions receive an image feature vector sequence of length 25 and generate 512 dimensional output with the same length. Each encoded feature vector is spliced by two feature vectors

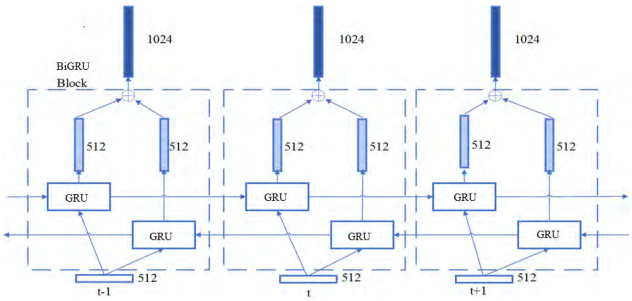


FIGURE 11. Architecture of BiGRU network.

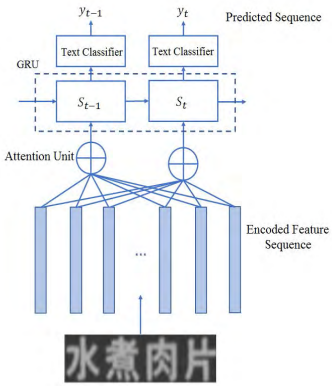


FIGURE 12. Architecture of proposed decoder network.

propagating along positive and negative time sequence respectively, which dimension is 1024 and also carries the context information of input text line images.

3) DECODER NETWORK

a: BASIC STRUCTURE

We adopt GRU network with attention mechanism [35] to decode the feature sequences into words. As shown in Fig.12, the decoder network is composed of Attention Unit, GRU Network and Text Classifier. Attention unit is responsible for assigning different attention weights to encoded features at different times, synthesizing new image context vectors as an input of GRU. GRU network receives the image context vectors and the output of network at previous moment, generating decoded feature vectors. Finally, the text classifier recognizes texts at current moment according to the current decoded feature vectors.

b: WORKING PRINCIPLE

As shown in Fig.13, our text verification model generates a context feature sequence $[h_1, h_2, ..., h_T]$ of text line images via encoding features in CNN and BiGRU network. For the different moment t, the encoded feature h_t of current input corresponds to the specific receptive field of the input text line image, i.e. the specific text area. Therefore, in order to better predict the output text classification at the time t, the decoder network introduces the same attention function as in [32] to better model the current concerns.

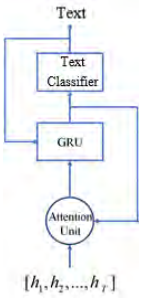


FIGURE 13. Word sequence generation process.

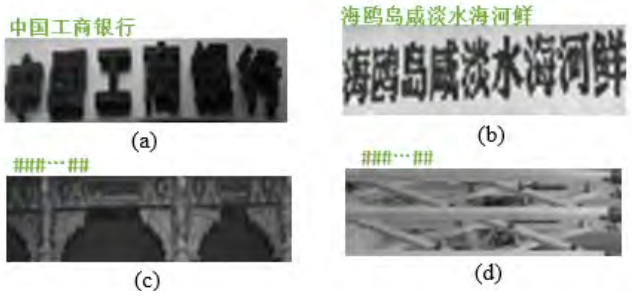


FIGURE 14. Some input image examples of text verification model.

Assume that $c_{t'}$ is the context vector of image at time t', and $s_{t'}$ is the hidden layer output of GRU network. $s_{t'}$ is defined as:

$$s_{t'} = g\left(y_{t'-1}, c_{t'}, s_{t'-1}\right) \tag{9}$$

where $y_{t'-1}$ is the precious prediction from the decoder network, $s_{t'-1}$ is the output of GRU network at time $t'-1$, g is the activation function. In addition, $c_{t'}$, the input context vector of decoder network at time t', is the weighted average of the output of encoder network at different times. We define $c_{t'}$ as:

$$c_{t'} = \sum_{t=1}^{T} \alpha_{t't} h_t \tag{10}$$

We define the attention weight $\alpha_{t't}$ at time t' as:

$$\alpha_{t't} = \frac{exp\left(e_{t't}\right)}{\sum_{k=1}^{T} exp\left(e_{t'k}\right)} \tag{11}$$

where $e_{t't}$ is correlated with the output $s_{t'-1}$ of GRU network at time $t'-1$ and the output h_t of encoder network at time t, it is defined as:

$$e_{t't} = v^T tanh\left(W_s s_{t'-1} + W_h h_t\right) \tag{12}$$

where v, W_s, W_h are parameters to be learned.

As shown in Fig.14, (a) and (b) are candidate bounding boxes generated by the text localization network. After image preprocessing, the network obtains gray images of 100×32 size, which are all positive samples of the text verification model. The green texts above image are labeling sequence, which represents the text sequence to be recognized from the image. (c) and (d) are background areas that are misjudged as texts by the text localization network. The background areas will be recognized by the encoder-decoder network that constitutes the text verification model, and then be correctly eliminated as non-texts. Since the background areas do not contain any texts, we uniformly represent them with a special character '#', and the length of the background sequence is 25, which is the same as the length of the encoded feature sequence.

In this paper, the output space takes 3775 common Chinese characters into account, also '#' character is used for background areas, and a special 'EOS'(End Of Sequence) token is used to denote the end of the text sequence to be recognized. That's to say, output space has a size of 3777.

Text recognition will be achieved through the text classifier in Fig.13. For different moments, the GRU network in decoder network first decodes the encoded feature sequence and outputs a 1024 dimensional decoded vector. In order to predict the probability of outputting the specific word at current moment, we adopt the fully connected layer with 1024×3777 dimension as the output of text classifier. The output vector is normalized by the softmax function, predicting the probability of occurrence for every word.

The decoder network makes use of encoded feature vectors to generate text sequences, which is actually a sequence classification process. At test time, the token with the highest probability in previous output is selected as the input token at next step. During the training, assuming that the output sequence is $y_1, y_2, ..., y_T$, the output at time t depends on the previous output of decoder network and the current input context vector c, we define the joint probability of the predicted output sequence as:

$$P\left(y_1, y_2..., y_T\right) = \prod_{t'=1}^{T} P\left(y_{t'} | \left\{y_1, y_2, ..., y_{t'-1}\right\}, c\right) \tag{13}$$

The training target of the text verification model is to minimize the loss function, which is defined as:

$$L = -\log P\left(y_1, y_2..., y_T\right) \tag{14}$$

IV. EXPERIMENTS

In this section, we evaluate the proposed method on our text detection dataset, the evaluation protocols include Precision, Recall and F-Measure.

A. OUR DATASET

The dataset, RCTW 2017 (ICDAR 2017 Competition on Reading Chinese Text in the Wild) [10], is a newly published dataset in the ICDAR 2017 Text Reading Competition, which is mainly prepared for Chinese text detection. In this dataset, some scene texts are multi-oriented. In this paper, we focus on the horizontal long text detection in complex background. In order to test the effectiveness of our proposed method,

FIGURE 15. Some examples of scene texts labeling in our dataset.

we construct our text detection dataset which is composed of the horizontal text images from RCTW 2017 and some scene images taken by ourselves. The training set contains 12k scene images, including 9k images taken using camera by ourselves and 3k images selected from the RCTW 2017. The test set contains 3k scene images, including 2k images taken by ourselves and 1k images selected from the RCTW 2017.

In our dataset, the images are labeled with the minimum rectangular box surrounding text areas. An example of labeling the scene text is shown in Fig.15 (a), and some examples of labeling images are shown in Fig.15 (b). The green rectangular boxes are ground truth boxes surrounding text areas.

B. EFFECTIVENESS OF TEXT LOCALIZATION NEURAL NETWORK

To perform initial text detection in a fast and elegant way, we design text detection layers based on SSD model, which are able to predict text in single forward pass followed by a standard non-maximum suppression. Besides, we add larger aspect ratios for default text bounding boxes, make vertical offsets to default boxes, and adopt a 1×5 convolution kernel instead of the 3×3 regular convolution kernel to extract the features more effectively. In this section, we will verify the effectiveness of the above designs through a series of experiments.

1) IMPLEMENTATION DETAILS

Our text localization network based on SSD is pre-trained on SynthText [28] and fine-tuned on our constructed dataset. For both pre-training and fine-tuning, input images are resized to 512×512. In pre-training, the learning rate is set to 10^{-3} for the first 60k iterations, and decayed to 10^{-4} for the following 60k iterations, with a weight decay of 5×10^{-4}. During fine-tuning, the learning rate is fixed to 10^{-4} for 20k iterations. Our network is optimized by BGD (Batch Gradient Descent) algorithm with a momentum of 0.9, then batch size is set to 32.

The whole network is implemented on Caffe [36] and encoded in Ubuntu 14.04. During training, the learning and optimization of all parameters are accelerated by Titan X GPU.

TABLE 1. Performance comparisons after adding larger aspect ratios.

a_r(aspect ratio) of default boxes	Text Localization Neural Network				
include$\{1, 2, 3\}$box	√	√	√	√	√
include$\{5\}$box		√	√	√	√
include$\{7\}$box			√	√	√
include$\{9\}$box				√	√
include$\{10\}$box					√
Precision	75.9	76.4	77.5	78.4	**79.2**
Recall	56.8	63.6	68.2	74.3	**75.8**
F-Measure	65.0	69.4	72.6	76.3	**77.5**

TABLE 2. Performance comparisons between SSD and Text Localization Neural Network (with larger aspect ratios).

Model	Precision	Recall	F-Measure
SSD	75.9	56.8	65.0
Text Localization Neural Network (larger aspect ratios)	**79.2**	**75.8**	**77.5**

2) LARGER ASPECT RATIOS

In this paper, we define 4 more aspect ratios on the basis of the original a_r in SSD, and $a_r = (1, 2, 3, 5, 7, 9, 10)$. We can calculate the width and height of the default box via (2) and (3). Therefore, as far as possible, we select the prime number as a_r to avoid the redundancy with other preseted a_r, and more text areas with different sizes and scales can be exactly detected. In addition, we retains the two ratios of 9 and 10, in order to improve the robustness of detecting long Chinese text in scene images via the text localization neural network. Table 1 shows the comparison of the detection results, obtained by adding larger aspect ratios to the default text boxes.

As shown in table 1, via adding larger aspect ratios, the precision and recall rate of text localization are all improved gradually. However the improvement rate becomes slower with a larger aspect ratio. Besides, via the experiment, we find that the precision and recall rate of text localization decrease with $a_r > 10$, which is mainly because that for the different short text line areas in dataset, the longer text boxes will falsely detect them as one merged text lines and cause the false localization.

Table 2 compares results of SSD and our text localization neural network with larger aspect ratios. The precision is improved from 75.9% to 79.2%, the recall rate is improved from 56.8% to 75.8%, and the F-Measure is improved from 65.0% to 77.5%, indicating that our model can improve the detection performance effectively.

3) DEFAULT BOXES WITH VERTICAL OFFSETS

In case of some dense text areas in natural scene images, we make offsets to default text bounding boxes in the feature map cell, and more ground truth text boxes can be matched. As shown in table 3, via this improvement, the precision and recall rate raises by 0.3% and 2.0%.

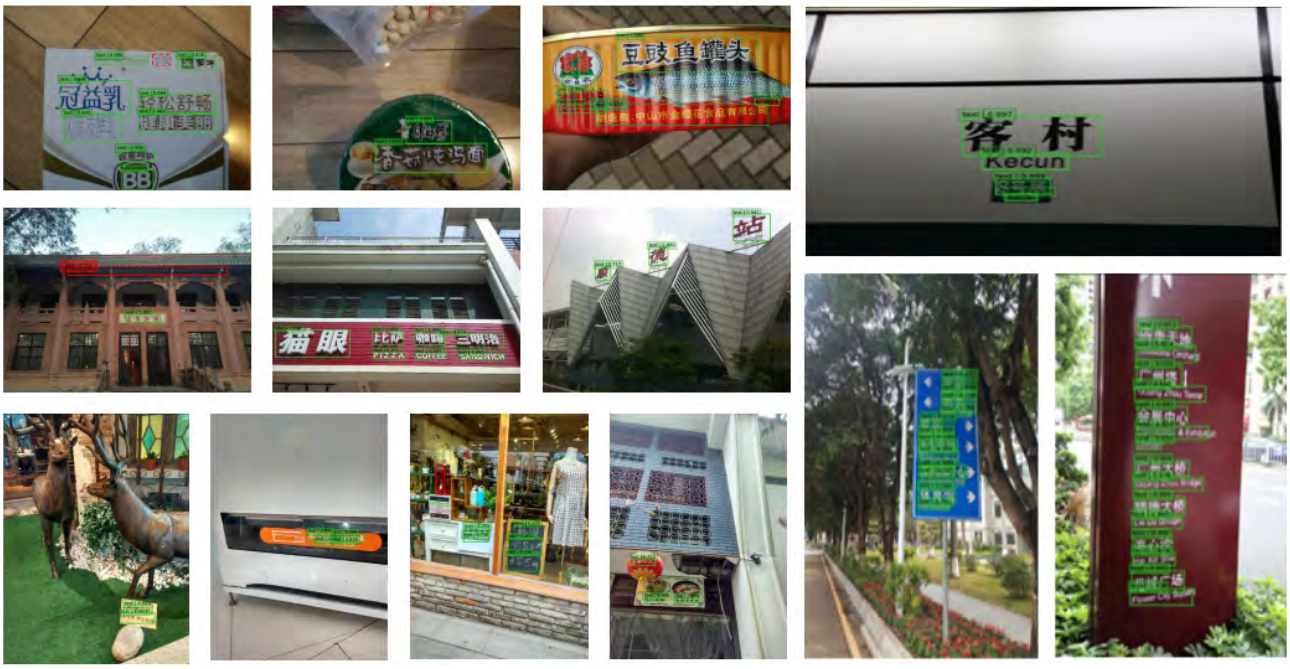


FIGURE 16. Some detection examples of Text Localization Neural Network on our dataset.Green solid boxes are the correctly detected scene texts. Red solid boxes are the falsely detected scene texts. Red dashed boxes are the ignored scene texts.

TABLE 3. Performance comparisons after expanding default boxes.

Model	Precision	Recall	F-Measure
Text Localization Neural Network (larger a_r)	79.2	75.8	77.5
Text Localization Neural Network (larger a_r&more default boxes)	**79.5**	**77.8**	**78.6**

TABLE 4. Performance comparisons after improving the convolution filter.

Model	Precision	Recall	F-Measure
Text Localization Neural Network (larger a_r&more boxes)	79.5	77.8	78.6
Text Localization Neural Network (larger a_r&more boxes&improved convolution filter)	**80.5**	**78.4**	**79.4**

4) IMPROVED CONVOLUTION FILTER

Table 4 shows that adopting a 1×5 convolution kernel instead of the 3×3 regular convolution kernel can more effectively extract the text features of input images, which obtains 80.5% precision and 78.4% recall rate.

5) EXAMPLE RESULTS

Fig. 16 shows some detection results on test dataset. As can be seen, our text localization neural network can detect horizontal scene text especially long and dense text as far as possible, with the low probability of missed detection, which proves the effectiveness of our proposed method. However, for some background areas, the text localization network will also misjudge them as text areas with high confidence, which reduces the accuracy of location to a certain extent.

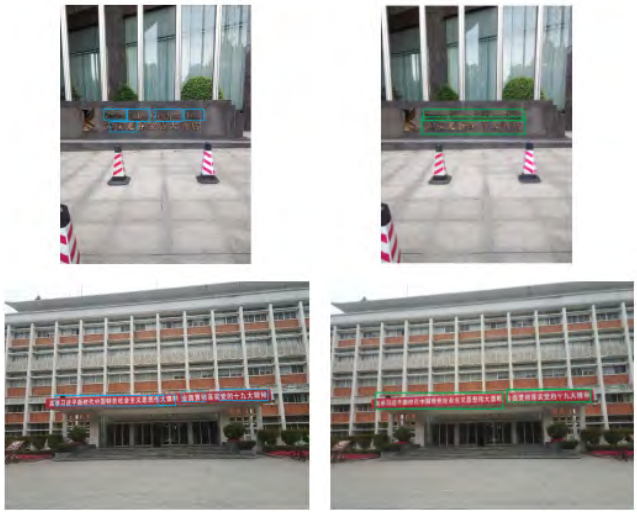


FIGURE 17. Results comparison between our text localization network and SSD. The left green boxes are results of our text localization network, and the right blue boxes are results of SSD.

6) COMPARISON WITH SSD

As shown in Fig. 17, the left green boxes are results of our text localization network, and the right blue boxes are results of the general object detection model SSD. As can be seen from the figure, our text localization network can correctly detect long texts, instead of dividing long text lines into several independent short text lines. The network is more robust to long text detection. Moreover, the text localization network can detect the text areas that have been missed by SSD model.

TABLE 5. Performance comparisons with SSD.

Model	Precision	Recall	F-Measure
SSD	75.9	56.8	65.0
Text Localization Neural Network	**80.5**	**78.4**	**79.4**

FIGURE 18. Some detection results of our text verification model. The green boxes are text areas detected by the text localization network, the red boxes are non-text areas detected falsely, and the texts in orange boxes are the recognition results of our text verification model.

In summary, table 5 compares results of our text localization neural network and SSD. In terms of precision, recall and F-Measure, the text localization neural network outperforms SSD by a large margin.

C. EFFECTIVENESS OF TEXT VERIFICATION MODEL

To train our text verification model, we take the candidate text box images generated from the text localization network as the positive samples, and the falsely detected background images as the negative samples. The input images are uniformly resized to 100×32, and the training set contains 200k images, the test set contains 20k images.

As shown in Fig. 18, (a) is the detection results of our text localization neural network. To get more background bounding boxes, we set the threshold to 0.95 in the non-maximum suppression (NMS) stage of our text localization neural network. (b) shows the recognition results of our text verification model for the bounding boxes. It can be seen that the text verification model can correctly recognize the text inside text bounding boxes and the background as the pre-defined characters '##...#', therefore it can be able to eliminate non-text detection areas. Figure (c) is the final localization result of the input image, which is detected by our text localization neural network and refined by our text verification model.

Table 6 shows the evaluation results after adopting our text verification model. Since the text verification model can only eliminate the false location of background areas, and cannot recall the missing text areas, it only affects the precision of text location. After eliminating False Positive (FP) detections, our detection precision raises by 2.5%.

D. COMPARISON WITH OTHER METHODS

Table 7 compares the results of the proposed method and other state-of-the-art methods. As we can see, our method achieves the highest recall rate of 78.4%, while only slightly lower in precision than CTPN [11]. Overall, our proposed method has a better performance for natural scene text detection, especially for the horizontal long and dense Chinese texts in natural scenes.

TABLE 6. Performance comparisons after applying our text verification model.

Model	Precision	Recall	F-Measure
Text Localization Neural Network	80.5	78.4	79.4
Text Localization Neural Network+ Text Verification Model	**83.0**	78.4	**80.6**

TABLE 7. Performance comparisons with some of the state-of-the-art methods.

Method	Precision	Recall	F-Measure
CNN+MSERS(Huang et al. [20])	80.7	67.9	73.7
CTPN(Tian et al. [11])	**84.3**	77.8	**80.9**
Multi-oriented FCN(Zhang et al. [23])	80.7	74.6	77.5
FCRN(Gupta et al. [28])	82.9	72.2	77.2
Our Method	83.0	**78.4**	80.6

V. CONCLUSION

We have presented an effective method for natural scene text detection based on SSD and encoder-decoder network. We make improvements on SSD to better handle horizontal text detection, especially long Chinese text in natural scenes. In detail, we add larger aspect ratios for default text bounding boxes, make vertical offsets to them, and adopt a 1×5 convolution kernel instead of the 3×3 regular convolution kernel. Besides, we propose an encoder-decoder network and make use of recognition results to refine the detection results. The encoder network, which is composed of CNN model and BiGRU network, encodes the text image features with rich context information. And the decoder network adopts GRU network with attention mechanism to decode the feature sequences into words. Comprehensive evaluations on our constructed dataset well demonstrate the effectiveness of our proposed method.

REFERENCES

[1] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.

[2] Y.-F. Pan, X. Hou, and C.-L. Liu, "A robust system to detect and localize texts in natural scene images," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, Sep. 2008, pp. 35–42.

[3] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. Eur. Conf. Comput. Vis.* Springer-Verlag, 2010, pp. 591–604.

[4] 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. (CVPR)*, Jun. 2010, pp. 2963–2970.

[5] A. R. Chowdhury, U. Bhattacharya, and S. K. Parui, "Scene text detection using sparse stroke information and MLP," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 294–297.

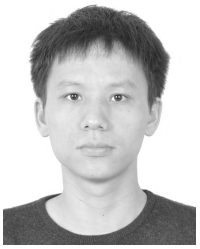
[6] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1083–1090.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2004.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.
[9] C. Luo, X. Gao, "Scene text detection with a SSD and encoder-decoder network based method," in *Video Analytics. Face and Facial Expression Recognition*, vol. 11264. Cham, Switzerland: Springer, 2019, pp. 25–34.
[10] B. Shi, C. Yao, M. Liao, M. Yang, P. Xu, L. Cui, S. Belongie, S. Lu, and X. Bai, "ICDAR2017 competition on reading Chinese text in the wild (RCTW-17)," 2017, *arXiv:1708.09585*. [Online]. Available: https://arxiv.org/abs/1708.09585
[11] 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.*, 2016, pp. 56–72.
[12] A. Vinciarelli, "A survey on off-line cursive word recognition," *Pattern Recognit.*, vol. 35, no. 7, pp. 1433–1446, Jul. 2002.
[13] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proc. 12th Int. Conf. Pattern Recognit.*, Oct. 2002, pp. 582–585.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[15] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 119–139.
[16] A. Shahab, F. Shafait, and A. Dengel, "Robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1491–1496.
[17] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.
[18] L. Sun, Q. Huo, W. Jia, and K. Chen, "Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 2715–2720.
[19] L. Sun, Q. Huo, W. Jia, and K. Chen, "A robust approach for text detection from natural scene images," *Pattern Recognit.*, vol. 48, no. 9, pp. 2906–2920, 2015.
[20] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 497–511.
[21] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A unified framework for text proposal generation and text detection in natural images," 2015, *arXiv:1605.07314*. [Online]. Available: https://arxiv.org/abs/1605.07314
[22] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[23] 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. (CVPR)*, Jun. 2016, pp. 4159–4167.
[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.
[25] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[26] 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.
[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[28] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.
[29] 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. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2642–2651.
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[31] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 6, Oct. 2017, pp. 3066–3074.
[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473
[33] 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.
[34] K. Cho and B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: https://arxiv.org/abs/1409.1259
[35] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: https://arxiv.org/abs/1508.04025
[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678.

XUE GAO received the Ph.D. degree in electronic science and technology from the South China University of Technology, in 2003, where he is currently an Associate Professor with the School of Electronic and Information Engineering. From 2004 to 2005, he was a Postdoctoral Researcher with the École Polytechnique de l'Université deNantes, Nantes, France. Since 2016, he has been a Senior Research Member with the SCUT-Zhuhai Institute of Modern Industrial Innovation. His current research interests include document analysis and recognition, and machine learning. He is a member of the IEEE.

SIYI HAN received the B.E. degree from the School of Electronic and Information Engineering, South China University of Technology, in 2017, where she is currently pursuing the master's degree. Her current research interests include deep learning and natural scene text detection.



CONG LUO received the M.Eng. degree from the School of Electronic and Information Engineering, South China University of Technology, in 2018. His research interests include deep learning and natural scene text detection.