IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 9, 2020, accepted August 26, 2020, date of publication August 31, 2020, date of current version September 14, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3020387*

# Toward Arbitrary-Shaped Text Spotting Based On End-To-End

**GUANGCUN WEI[1,2], (Member, IEEE), WANSHENG RONG[1], YONGQUAN LIANG[1], XINGUANG XIAO[1], AND XIANG LIU[1]**

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]College of Intelligent Equipment, Shandong University of Science and Technology, Taian 271019, China

Corresponding author: Guangcun Wei (weigc@sdust.edu.cn)

This work was supported in part by the Key Research Program of Shandong Province under Grant 2018GGX101011.

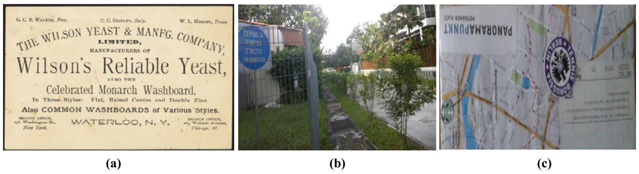
**ABSTRACT** At present, text spotting in natural scenes has become one of the research hotspots. Among them, curvilinear text and long text are the main difficulties of text spotting in natural scenes. To better solve these two types of problems, we propose a novel end-to-end text spotting model. The model includes three parts: shared convolution module, text detector module and text recognizer module. For the problem of long text, we adopt the corner attention mechanism to extract the features of long text more effectively. For the problem of curve text, we feed the rectification feature map into the SA-BiLSTM decoder to recognize the curve text more effectively. More importantly, the joint optimization strategy realizes the mutual promotion function of the text detection task and the text recognition task. Experimental results on TotalText, ICDAR2015, ICDAR2013, CTW1500, COCO-Text and MLT datasets prove that our method achieves excellent performance and robustness in text spotting tasks based on end-to-end natural scenes.

**INDEX TERMS** Natural scene text spotting, SA-BiLSTM, end-to-end, joint optimization.

## I. INTRODUCTION

Text spotting in natural scenes has high research value and a wide range of application scenarios, and has become one of the hot topics of research. On the one hand, text is an important carrier for the spread of human civilization; On the other hand, rich high-level semantic information helps us understand the world better. More importantly, text information in natural scenes has a very wide range of application scenarios, such as image search, instant translation, robot navigation, blind assisted reading, and industrial automation. Therefore, text spotting in natural scenes has important research value.

However, the inherent feature of text in natural scenes increase the difficulty of text spotting. First of all, the diversity of text, such as text in different languages, fonts, font sizes, shapes, etc., as shown in Figure 1.a; In addition, the complexity of the text background, the background may contain many objects similar to the text, such as leaves, bricks, windows, fences, etc., as shown in Figure 1.b; Finally, unsatisfactory data quality, imperfect imaging conditions often lead to low data quality, such as low resolution, distortion and blur, as shown in Figure 1.c.


THE WILSON YEAST & MANFG. COMPANY,
LIMITED,
MANUFACTURERS OF
Wilson's Reliable Yeast,
Celebrated Monarch Washboard,
Also COMMON WASHBOARDS of Various Styles.
WATERLOO, N. Y.
PANORAMAPUNKT

(a) (b) (c)

**FIGURE 1. Characteristics of text in natural scenes. (a) Text diversity. (b) Complex background. (c) Poor image quality.**

Traditional OCR (optical character recognition) processing methods generally decompose text spotting in natural scenes into two independent subtasks: text detection task and text recognition task [1]. The text detection is used to detect whether there is a text instance in the picture; text recognition is to recognize the content of text. The OCR processing method has achieved good results in text spotting problems in natural scenes. However, this method ignores the inherent connection between text detection and text recognition. First, the accumulation of training errors, the errors in the text detection stage will be passed to the text recognition process, which will lead to a worse text recognition performance; second, the text detection task and the text recognition task cannot be optimized at the same time.

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

To address the issues of current OCR methods, end-to-end OCR processing has been proposed by Li *et al.* [2], Liu *et al.* [3], He *et al.* [4], Sun *et al.* [5] and Lyu *et al.* [6]. Their common idea is that the text detection branch and the text recognition branch share a feature extraction network. Therefore, the text detector and the text recognizer can be optimized jointly, which can effectively solve the problem of error accumulation.

The end-to-end method proposed by Li *et al.* [2] can achieve good performance on horizontal text datasets, but it cannot handle curvilinear text problems in natural scenes well. In order to better deal with curvilinear text, Liu *et al.* [3], He *et al.* [4] and Sun *et al.* [5] proposed similar solutions. Their common ideas are: first, the feature extraction network extracts the features of the text area; second, rectify feature maps; finally, feed the rectified feature maps into the text recognizer.

Although Lyu *et al.* [6] can improve the performance of text detection, it has lost the potential order information between characters. On the one hand, even if each character is detected correctly, it is difficult to connect them into words correctly. On the other hand, the author's default text is recognized from left to right, so it cannot handle non-traditional text directions. The paper [7] adopts a joint optimization strategy which makes good use of the potential internal connection between text detection and text recognition, but it cannot well avoid the interference of complex backgrounds.

In order to better solve the above problems, we propose a novel end-to-end text spotting framework, which uses a joint optimization strategy to effectively utilize the inherent connection between text detection tasks and text recognition tasks. Firstly, the text detector uses a corner attention mechanism [8], which can better solve the problem of long texts; Secondly, the TPDM (Text Point Detection Module) can better avoid the interference of complex backgrounds. In addition, a feature rectification network is used to rectify the feature maps, and then the rectified feature maps is fed into the recognizer, which is beneficial to the recognition of curve text. Most importantly, SA-BiLSTM (spatial attention mechanism with BiLSTM) is a text recognition model based on the combination of spatial attention mechanism and BiLSTM, which can more effectively extract semantic information between characters.

## II. RELATED WORK

### A. TEXT DETECTION IN NATURAL SCENE

The main difficulty of text detection in natural scenes is caused by the characteristics of texts in natural scenes, such as the diversity of text directions and the diversity of text languages. The paper [9] transformed the text detection task into a series of text box detection and introduces RNN (recurrent neural networks) to improve the effect of text detection. It has a significant effect on detecting horizontal text data, but it is not suitable for non-horizontal text. The thesis [10] first be cut each word into more directional small text segments that are easier to detect, and then connects each small text block into a word with a neighboring link, which is conducive to recognizing a wide range of lengths with directions Words and lines of text. The paper [11] first uses a FCN (fully convolutional network) to generate multi-scale fusion feature maps, and then directly performs pixel-level text block prediction on this basis, supporting two types of text area annotations: rotating rectangle box and arbitrary quadrilateral.

### B. TEXT RECOGNITION IN NATURAL SCENE

Natural scene text recognition includes two categories: the CTC-based method and the method based on the attention mechanism. The paper [12] used the CTC-based method for the first time in recognition system and achieved good results. The papers [13], [14], [15], [16] and [17] also adopted the improved CTC method to further verify the effect of CTC method. Paper [18] first proposed the attention mechanism to solve the problem of machine translation, and it is now widely used in text recognition task in natural scenes. Paper [19] proposed an attention mechanism based on encoding and decoding, which can better adapt to the problem of text recognition. For irregular text recognition (warped and curved text), the paper [20] proposed a combination of attention mechanism and spatial transformation network to improve the performance of irregular text recognition.

### C. TEXT SPOTTING BASED ON END-TO-END IN NATURAL SCENE

Text detection and text recognition are often regarded as two independent sub-problems, ignoring the intrinsic connection between text detection and text recognition. Therefore, end-to-end text spotting has become one of the research trends. The paper [21] uses a text detector based on SSD (single shot multibox detector) [22] and a text recognizer based on CRNN (convolutional recurrent neural network) [23]. Paper [2] uses a text detector based on RPN (Region Proposal Network) [24] and a text recognizer based on the attention LSTM (long short-term memory) mechanism. Papers [2] and [7] use a strategy based on joint optimization of text detectors and text recognizers to achieve overall performance improvement.

We propose the advantages of an end-to-end text spotting framework as follows. First, the joint optimization model effectively uses the potential internal connection between the two tasks of text detection and text recognition to improve the overall performance. Second, the corner attention mechanism can better solve the problem of long texts; in addition, the TPDM can better avoid the interference of complex backgrounds. Finally, the rectified feature map is fed into the SA-BiLSTM recognizer, which can more effectively extract the semantic information between characters and is conducive to text recognition.

# III. MODEL DESIGN

## A. MODEL

We propose a text spotting model based on end-to-end natural scenes. The model consists of three parts: shared convolutional network, text detector and text recognizer. The scene text spotting flow chart is shown in Figure 2. First, feed the preprocessed image into the shared convolutional network to extract the shared feature maps. Then feed the shared feature maps into the text detector and text recognizer, and the text detector and text recognizer promote each other through Boxes and TPDM. Finally, a joint optimization strategy is used to make full use of the inherent connection between the text detection task and the text recognition task to improve the overall performance of text spotting in natural scenes.

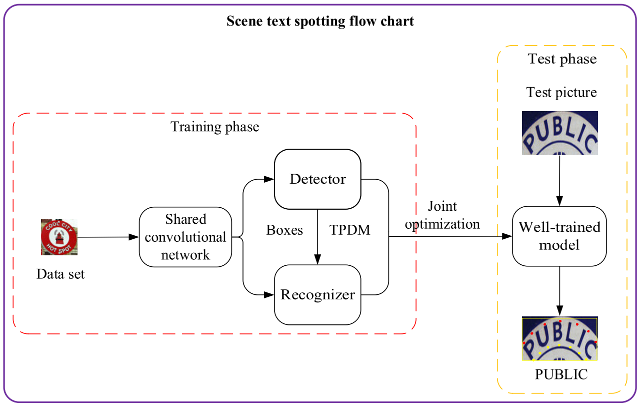


**FIGURE 2. Scene text spotting flow chart.**

## B. MODEL FRAMEWORK

This paper proposes an end-to-end text spotting framework structure, as shown in Figure 3. It includes three parts: shared convolution feature extraction network, text detector and text recognizer. First, the picture is fed into the shared convolution feature extraction network for feature learning, and the obtained feature map is input to a text detector and a text recognizer. The IRM (Text Regressor Module) in the paper [8] can better adapt to the detection of long texts. In addition, the text detector uses boundary points to represent text instances, which is more suitable for detecting and recognizing text of any shape than rectangle box. The text recognition model based on SA-BiLSTM decoder can more effectively extract the semantic information between characters. The following describes each process separately.

### 1) SHARED CONVOLUTION MODULE

The shared convolution module adopts ResNet-50 [25] structure to extract shared features. Since texts in natural scenes usually have different sizes. In order to better adapt to texts of various sizes, it is necessary to maintain a large receptive field and richer features. Use dilated convolution to maintain a larger receptive field. Inspired by FPN (feature pyramid networks) [26], we use the method of concatenating low-resolution feature maps and high-resolution feature maps to extract richer text features. The size of the final output feature map is 1/4 of the input picture.

### 2) TEXT DETECTOR

The text detector is composed of three parts: text regression module, iterative optimization module and text point detection module, as shown in Figure 3.

#### a: TEXT REGRESSOR MODULE

Inspired by [11], TRM (Text Regressor Module) uses a fully convolutional sub-network as a text regressor. Based on the shared convolutional feature map, two prediction channels are output by pixel-wise: text and non-text. We use a similar approach to others: the pixels in the text area are defined as positive samples, and the pixels in the non-text area are defined as negative samples. For each positive sample, there are 8 channels to predict the four corners of the text box. TRM has two functions, one is the classification task of text and non-text, and the other is to locate the text.

For the classification task, we use the scale-invariant dice-coefficient function proposed in [8], which is defined as follows:

$$L_{cls} = 1 - \frac{2 * sum\left(y \cdot \hat{y} \cdot w\right)}{sum\left(y \cdot w\right) + sum\left(\hat{y} \cdot w\right)} \tag{1}$$

where $sum$ is a cumulative function in a two-dimensional space, $y$ is a binary label value, $\hat{y}$ is a predicted value, and $w$ is a two-dimensional weight space.

For the text localization regression task, due to the better robust performance of smooth $L_1$[26],we use smooth $L_1$ to optimize the text regression task $L_{loc}$.

Therefore, the loss function of TRM is defined as follows:

$$L_{trm} = L_{loc} + \alpha L_{cls} \tag{2}$$

where $\alpha$ is the hyper-parameter. The $\alpha$ parameter is used to balance the two sub-loss functions. In the experiment, $\alpha$ is set to 0.01.

#### b: ITERATIVE REFINEMENT MODULE

To better detect long texts, we adopt the IRM (iterative refinement module) proposed in the paper [8]. Since the position close to the corner of the text area can obtain more accurate boundary information in the same receptive field, a corner attention mechanism is introduced to return to the coordinate offset of each corner point.

The loss function of IRM is defined as:

$$L_{irm} = \frac{1}{K * 8} \sum_{k=1}^{K} \sum_{j=1}^{8} smooth_{L1}\left(C_k^j + C_k^{\hat{j}}\right) \tag{3}$$

where $K$ represents selecting $K$ detected text boxes from the output of the TRM step, $C_k^j$ represents the offset of the $j$-th coordinate of the $k$-th text box, and $C_k^{\hat{j}}$ is the corresponding predicted value.

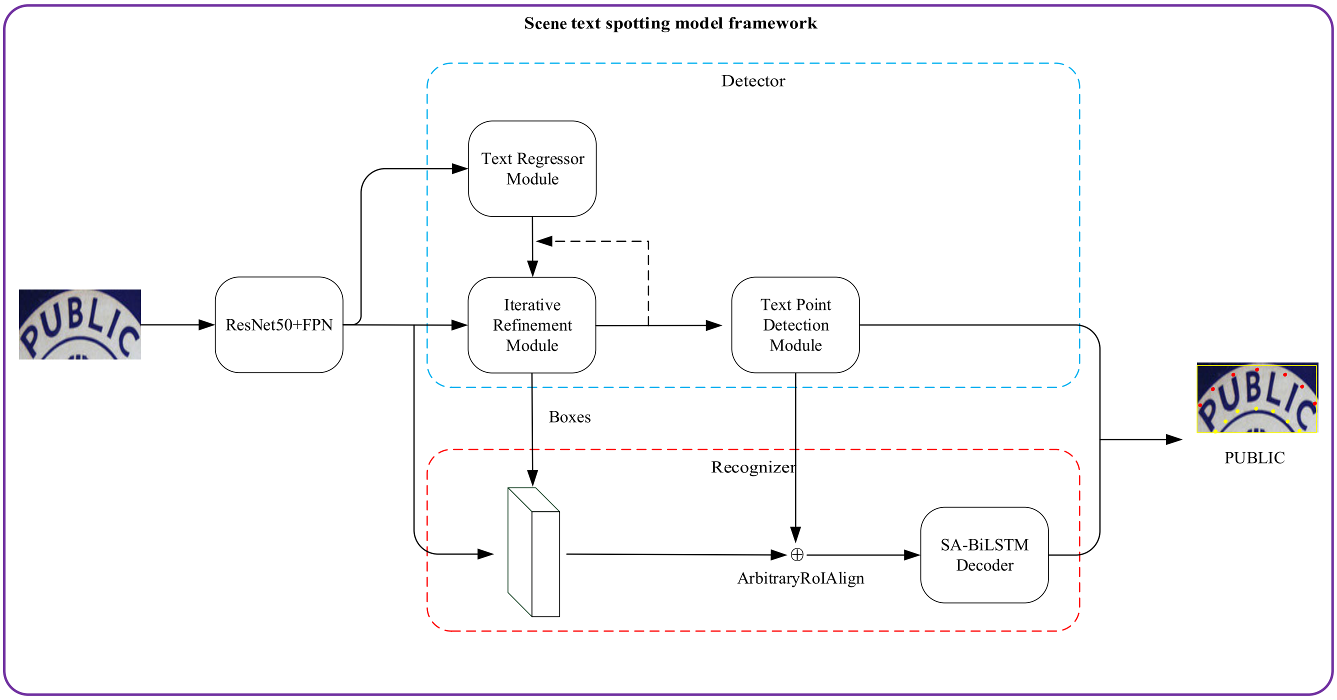


**FIGURE 3.** **Scene text spotting model framework.**

### *c: TEXT POINT DETECTION MODULE*

Using ROI-Align to extract the features of the text quadrilateral will extract a lot of background noise, which will affect the recognition network. The use of boundary points to represent arbitrary-shaped text can effectively avoid such problems. First, the boundary points can describe the precise text shape and eliminate the impact of background noise. Secondly, the boundary points are easy to rectify any shape text into horizontal text, which is beneficial to the text recognition network.

TPDM consists of four stacked 3 × 3 convolutional layers and a fully connected layer. Inspired by RPN where proposals are regressed based on default anchors, we use a similar method to set a set of default points for the text boundary. Specifically, $N$ points are sampled at equal distances on each long side of the text instance as target boundary points. The corresponding default points are placed equidistantly along the long side of the smallest quadrilateral. Instead of directly predicting the coordinates of the boundary point, the offset of the default point associated with it is first generated. The module predicts a $4N$-d vector which is coordinate offsets (2-d) of $2N$ boundary points. Given the coordinate offsets $(\Delta x', \Delta y')$, the boundary points $(x_b, y_b)$ can be passed Calculated:

$$\begin{cases} x_b = x'_d + w_0 \Delta x' \\ y_b = y'_d + h_0 \Delta y' \end{cases} \tag{4}$$

where $(x'_d, y'_d)$ is the set default point, $w_0, h_0$ are the width and height of the text box output by the IRM, respectively. The loss function $L_{tpdm}$ of TPDM is defined as:

$$L_{tpdm} = \frac{1}{2N} \sum_{i=1}^{2N} \left( Smooth_{L1} \left( \hat{x}_{b,i}, x_{b,i} \right) + Smooth_{L2} \left( \hat{y}_{b,i}, y_{b,i} \right) \right) \tag{5}$$

where $(x_{b,i}, y_{b,i})$ is the $k$-th predicted text point,,whose associated target boundary point is $(\hat{x}_{b,i}, \hat{y}_{b,i})$.

### 3) TEXT RECOGNIZER

### *a: ArbitraryRoIAlign*

In order to better adapt to the curve text, we use a rectification network similar to the paper [18] to the features. Specifically, TPS (Thin-Plate-Spline) can rectify the deformed image (affine, perspective, curve arrangement, etc.) to obtain the rectified feature map, which is convenient for text recognition.

### *b: SA-BiLSTM DECODER*

SAM is proposed in the paper [27]. In contrast, we combine the spatial attention mechanism with BiLSTM to better extract the semantic information between texts. The structure of SA-BiLSTM is shown in Figure 4.

Suppose $T$ iterations are needed, and the predicted character sequence is $y = (y_1, \ldots, y_T)$. There are three inputs at step t: the input feature F, the hidden state $s_{t-1}$ of the previous iteration and the character category $y_{t-1}$ predicted by the previous iteration.

Firstly, expand the $s_{t-1}$ vector into a feature map of shape $(V, H_p, W_p)$. V represents the size of the RNN hidden layer

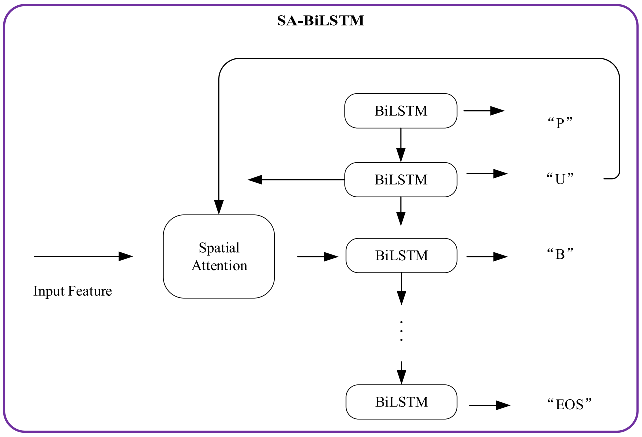


**FIGURE 4.** **SA-BiLSTM module structure diagram.**

and is set to 256.

$$S_{t-1} = expand\_\mathrm{dim}\left(s_{t-1}, H_p, W_p\right) \tag{6}$$

Secondly, calculate the weight $\alpha_t$ of attention:

$$e_t = W_t \times \tanh\left(W_s \times S_{t-1} + W_f \times F + b\right) \tag{7}$$

$$\alpha_t\,(i,j) = \exp\left(e_t\,(i,j)\right) / \sum\nolimits_{i'=1}^{H_p} \sum\nolimits_{j'=1}^{W_p} \exp\left(e_t\left(i', j'\right)\right) \tag{8}$$

The shapes of $e_t, \alpha_t$ are $(H_p, W_p)$, $W_t, W_s, W_f$ and b are training weights and bias.

Thirdly, calculate the weighted feature $g_t$:

$$g_t = \sum\nolimits_{i=1}^{H_p} \sum\nolimits_{j=1}^{W_p} \alpha_t\,(i,j) \times F(i,j) \tag{9}$$

Embed the characters of the character type $y_{t-1}$ predicted by the previous iteration and perform a concat operation with $g_t$ to calculate the input $r_t$ of the RNN:

$$f\,(y_{t-1}) = W_y \times onehot\,(y_{t-1}, N_c) + b_y \tag{10}$$

$$r_t = concat(g_t, f\,(y_{t-1})) \tag{11}$$

$W_y$, $b_y$ are weights and bias, and $N_c$ is the number of types of sequence decoders. We set it to 79, including English letter case, Arabic numerals, and several special characters.

The $r_t$, $s_{t-1}$ is fed into the RNN (Bi-LSTM):

$$(x_t, s_t) = rnn\,(s_{t-1}, r_t) \tag{12}$$

Finally, the prediction result of the t-th iteration is as follows:

$$p\,(y_t) = softmax\,(W_o \times x_t + b_o) \tag{13}$$

$$y_t \sim p\,(y_t) \tag{14}$$

The loss function $L_{recog}$ of the text recognizer is defined as:

$$L_{recog} = -\frac{1}{T} \sum\nolimits_{t=1}^{T} logp\,(y_t) \tag{15}$$

where $T$ represents the length of the tag sequence.

### 4) JOINT OPTIMIZATION AND LOSS FUNCTION

Our proposed text spotting framework uses a joint optimization strategy: text detection tasks and text recognition tasks share features and optimize at the same time. it saves computing time and can make better use of the internal connection between text detection and text recognition tasks. Therefore, the loss function $L$ is defined as follows:

$$L = \sigma_1 L_{trm} + \sigma_2 L_{irm} + \sigma_3 L_{tpdm} + \sigma_4 L_{recog} \tag{16}$$

where $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ are used to balance the four submodules, all set to 1.0 in the experiment.

## IV. EXPERIMENTAL DESIGN AND ANALYSIS

### A. DATA SET

The data set used in our experiment and its related introduction are as follows:

SynthText is a synthetic data set, contains 800,000 synthetic images and has a large number of multi-directional text examples.

TotalText is a text dataset of comprehensive scenes. The dataset contains 1255 training datasets and 300 test datasets, with a variety of texts such as horizontal, directional and curved texts. The data set provides word-level annotations.

Different from the TotalText data set, CTW1500 proposed in 2017 is a scene text data set containing arbitrary shape Chinese and English texts, with 1000 training images and 500 test images.

ICDAR2015 is a text dataset of natural scenes proposed in the ICDAR 2015 competition. These images are multi-directional text data sets, including 1000 training data sets and 500 test sets. All pictures provide character-level and word-level annotations.

The COCO-Text dataset has a total of 6368 images, it contains 43686 training data sets and 10,000 test sets.

The ICDAR2013 dataset contains only horizontal text. The training data set contains 229 images, and the test set contains 233 images. The data set provides both character-level and word-level annotations.

MLT is a scene text data set in multiple languages. It contains 7200 training data sets, 9000 test sets and 1800 verification data sets.

### B. EXPERIMENTAL DETAILS

Different from the previous strategy of independent training or alternating training of text detection and text recognition, we used joint optimization based on the end-to-end text spotting model. The entire training process includes two stages: first, pre-training on SynthText dataset, and finally fine-tuning on actual data (TotalText, ICDAR2015, ICDAR2013, CTW1500, COCO-Text and MLT).

The experiment uses the SGD optimization algorithm, the weight attenuation value is 0.0001, and the momentum value is 0.9. During the pre-training phase, 300K iterative training was carried out for the model, the default value of the initial learning rate was 0.01, and in the 150K and 300K iterations, the learning rate dropped by one tenth.

During the fine-tuning phase, The default value of the initial learning rate was set to 0.001 and then reduced to one-tenth in 150k iterations. Fine-tuning process stops at 200k times. Our experimental model is based on Pytorch.

**LabelGeneration** Since the training stage requires equidistant boundary points to train TPDM, we use the algorithm in [28] to sample on the long side of the text boundary. N is also set to 7.

### C. EXPERIMENTAL ANALYSES

#### 1) CURVED TEXT

We have conducted experiments on the TotalText data set to verify the effectiveness of the model on arbitrary-shaped text. In the test phase, the long side of the picture is set to 1100. To be fair, this article follows the evaluation protocol in the latest method [29].

The performance of the experimental scheme proposed in this paper on the Total-Text dataset is shown in Table 1. As can be seen from Table 1, our method achieved the most advanced performance in both text detection tasks and end-to-end text recognition tasks. In particular, compared with the method in [28], the performance of the method in this paper is 0.9% and 2.4% higher than that of Boundary in text detection tasks and end-to-end text spotting (without lexicon) tasks respectively. Compared with [27], it improved by 3.0% in text detection tasks. It should be noted that [27] requires character-level annotations. The reasons for the performance improvement are as follows, first, the corner attention mechanism is conducive to the detection and recognition of long text; second, the text decoder based on SA-BiLSTM can better extract the semantic information of the text; finally, TPDM can better avoid interference from complex background.

**TABLE 1.** Results on totaltext.

| Method | Detection | | | E2E | |
|---|---|---|---|---|---|
| | P | R | F | None | Full |
| Baseline[29] | 40.0 | 33.0 | 36.0 | - | - |
| Textboxes[21] | 62.1 | 45.5 | 52.5 | 36.3 | 48.9 |
| P. Lyu MaskTextSpotter[6] | 87.0 | 80.2 | 83.4 | 52.9 | 71.8 |
| M. Liao MaskTextSpotter[27] | 88.3 | 72.4 | 85.2 | 65.3 | **77.4** |
| Boundary[28] | 88.9 | 85.0 | 87.0 | 65.0 | 76.1 |
| Our(det only) | 89.0 | 86.4 | 87.8 | - | - |
| Our(end-to-end) | **89.2** | **87.3** | **88.2** | **67.4** | 76.8 |

"P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "E2E" means end-to-end, "None" means recognition without any lexicon, "Full" lexicon contains all words in test set.

We further verified the effectiveness of our method on the CTW1500 data set. The experimental results are shown in Table 2. It can be seen from Table 2 that we have also achieved very good performance on the CTW1500 data set. Especially in the text detection task, it is 1.1% higher than [30].

#### 2) ORIENTED TEXT

The experimental scheme proposed in this paper was tested on the ICDAR2015 dataset to confirm the validity of the oriented text. The results are shown in Table 3. Compared with [28], our method showed an improvement of 1.2% and 3.7% in text detection tasks and the end-to-end text spotting with strong lexicon, respectively. In addition, compared with [27], our method improves text detection performance by 2.8%.

**TABLE 2.** Results on CTW1500.

| Method | Detection | | | E2E | |
|---|---|---|---|---|---|
| | P | R | F | None | Full |
| SegLink[10] | 42.3 | 40.0 | 40.8 | - | - |
| EAST[11] | 49.1 | 78.8 | 60.4 | - | - |
| FOTS[3] | 79.5 | 52.0 | 62.8 | 21.1 | 39.7 |
| CTD+TLOC[31] | 77.4 | 69.8 | 73.4 | - | - |
| TextSnake[32] | 67.9 | 85.3 | 75.6 | - | - |
| TextDragon[30] | 84.5 | 82.8 | 83.6 | 39.7 | 72.4 |
| ABCNet[33] | - | - | - | **45.2** | 74.1 |
| Our(det only) | 85.2 | 80.6 | 82.7 | - | - |
| Our(end-to-end) | **86.3** | 83.1 | **84.7** | 38.9 | **74.5** |

"P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "E2E" means end-to-end, "None" means recognition without any lexicon, "Full" lexicon contains all words in test set.

**TABLE 3.** Results on ICDAR2015.

| Method | Detection | | | E2E | | |
|---|---|---|---|---|---|---|
| | P | R | F | S | W | G |
| TextBoxes++[34] | 87.2 | 76.7 | 81.7 | 73.3 | 65.9 | 51.9 |
| He* et al[4] | 87.0 | 86.0 | 87.0 | 82.0 | 77.0 | 63.0 |
| P.Lyu MaskTextSpotter[6] | 91.6 | 81.0 | 86.0 | 79.3 | 73.0 | 62.4 |
| TextNet[5] | 89.4 | 85.4 | 87.4 | 78.7 | 74.9 | 60.5 |
| FOTS[3] | 91.0 | 85.2 | 88.0 | 81.1 | 75.9 | 60.8 |
| Boundary[28] | 89.8 | 87.5 | 88.6 | 79.7 | 75.2 | **64.1** |
| M.Liao MaskTextSpotter[27] | 86.6 | 87.3 | 87.0 | 83.0 | **77.7** | **73.5** |
| Our(det only) | 90.4 | 87.0 | 88.7 | - | - | - |
| Our(end-to-end) | **91.9** | **87.8** | **89.8** | **83.4** | 75.1 | 63.3 |

"P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "S","W" and "G" mean recognition with strong , weak and generic lexicon respectively. "*" denotes that training dataset of MLT2017 is used for training .

**TABLE 4.** Results on COCO-Text.

| Method | Detection | | | E2E | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| EAST[11] | 50.0 | 32.4 | 39.5 | - | - | - |
| WordSup[35] | 45.2 | 30.9 | 36.8 | - | - | - |
| RRD MS[36] | 64.0 | 57.0 | 61.0 | - | - | - |
| Lyu et al [37] | 72.5 | 52.9 | 61.1 | - | - | - |
| M.Liao MaskTextSpotter[27] | 66.8 | **58.3** | 62.3 | **65.8** | 37.3 | 47.6 |
| Our(det only) | 71.3 | 55.4 | 62.4 | - | - | - |
| Our(end-to-end) | **72.8** | 56.3 | **63.5** | 63.6 | **39.2** | **48.5** |

"P", "R" and "F" mean Precision, Recall and F-measure respectively.

On the COCO-Text data set, we further verify the effectiveness of the oriented text. The results are shown in Table 4. Compared with [27], our method improves 1.2% and 0.9% in text detection tasks and end-to-end text spotting tasks, respectively.

### 3) HORIZONTAL TEXT

We conducted tests on the ICDAR2013 dataset to verify the effectiveness of the model on the horizontal text dataset. The results are shown in Table 5. It can be seen from Table 5 that the method proposed in this paper also achieves good performance on the horizontal data set. It should be noted that [27] requires character-level annotations.

**TABLE 5. Results on ICDAR2013.**

| Method | Detection | | | E2E | | |
|---|---|---|---|---|---|---|
| | P | R | F | S | W | G |
| TextBoxes++[34] | 88.0 | 74.0 | 81.0 | 93.0 | 92.0 | 85.0 |
| He* et al[4] | 91.0 | 89.0 | 90.0 | 91.0 | 89.0 | 86.0 |
| Boundary[28] | 93.1 | 87.3 | 90.1 | 88.2 | 87.7 | 84.1 |
| P.Lyu MaskTextSpotter[6] | 95.0 | 88.6 | 91.7 | 92.2 | 91.1 | 86.5 |
| M.Liao MaskTextSpotter[27] | **94.8** | **89.5** | **92.1** | **93.3** | **91.3** | **88.2** |
| Our(det only) | 93.7 | 87.4 | 90.4 | - | - | - |
| Our(end-to-end) | 94.3 | 88.2 | 91.1 | 91.3 | 90.8 | 85.8 |

"P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "S","W" and "G" mean recognition with strong , weak and generic lexicon respectively. "*" denotes that training dataset of MLT2017 is used for training .

### 4) MULTI-LANGUAGE

In order to verify the reliability of our method, we conduct experiments on MLT data. The experimental results are shown in Table 6. Our method also achieves good performance on the MLT dataset.

**TABLE 6. Results on MLT.**

| Method | Det-R | E2E-R | P |
|---|---|---|---|
| Busta et al.[38] 2+ | 68.4 | 42.9 | 53.7 |
| Busta et al.[38] 3+ | 69.5 | 43.3 | 59.7 |
| M.Liao MaskTextSpotter[27] 2+ | 80.0 | 47.9 | 68.3 |
| M.Liao MaskTextSpotter[27] 3+ | 82.8 | 48.5 | 60.5 |
| Our 2+ | 81.3 | 48.7 | 66.2 |
| Our 3+ | 82.1 | 51.3 | 61.1 |

"Det-R": detection recall; "E2E-R": end-to-end recognition recall; "P": precision. "2+" and "3+" mean that words whose length are large than 2 and 3 are counted respectively. "

### 5) VISUALIZATION

Figure 5 is the visualization results of some data. As can be seen from the first two lines, the model can handle arbitrarily shaped text well. The third line, due to the complexity of the text background and the blurred image quality, leads to misdetection and missed detection.

**TABLE 7. Ablation experimental results.**

| Datasets | Methods | Detection | | | E2E |
|---|---|---|---|---|---|
| | | P | R | F | None |
| ICDAR2015 | **No-TPDM** | 86.9 | 84.4 | 85.6 | 62.6 |
| | **SAM** | 89.2 | 86.6 | 87.9 | **64.7** |
| | **FULL** | **91.9** | **87.8** | **89.8** | 64.5 |
| TotalText | **No-TPDM** | 84.1 | 83.5 | 83.8 | 61.3 |
| | **SAM** | 88.4 | 86.1 | 87.5 | 66.1 |
| | **FULL** | **89.2** | **87.3** | **88.2** | **67.4** |

"P", "R" and "F" mean Precision, Recall and F-measure in detection task respectively. "E2E" means end-to-end, "None" means recognition without any lexicon.

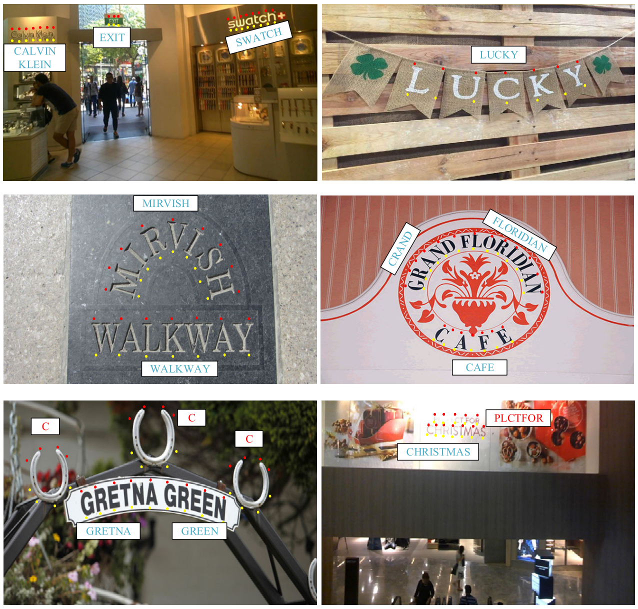


**FIGURE 5. Visualization of results.**

### 6) ABLATION EXPERIMENT

Ablation experiments can better verify our proposed model.

**FULL**: It is our end-to-end text discovery framework.

**No-TPDM**: we train a model named "No-TPDM" which removes the TPDM part from FULL. It is used to verify the effectiveness of TPDM.

**SAM**: We train a model and name it "SAM", using the SAM proposed in the paper [27] as the text recognizer of our model. It is used for comparison with our proposed SA-BiLSTM.

As shown in Table 7, on the ICDAR2015 data set, compared with No-TPDM, FULL has increased by 4.2% and 1.9% in text detection tasks and end-to-end text spotting tasks, respectively. For the TotalText dataset, compared with No-TPDM, FULL has increased by 4.4% and 6.1% respectively in text detection tasks and end-to-end text spotting tasks. Therefore, TPDM can effectively share features between text detection and text recognition, and then make full use of the inherent relationship between text detection and text recognition to improve the overall performance of text spotting.

We propose a text recognizer based on SA-BiLSTM and compare it with the text recognizer based on SAM proposed in the paper [27]. As shown in Table 7, our proposed SA-BiLSTM text recognizer achieves good performance. Especially on the TotalText data set, compared with SAM, FULL has increased by 0.7% and 1.3% respectively in text detection tasks and end-to-end text spotting tasks.

## V. CONCLUSION

Aiming at the problem of arbitrarily shaped text spotting in natural scenes, we propose an end-to-end text spotting framework and adopt a joint optimization strategy. Experiments show that the text decoder based on the SA-BiLSTM mechanism can better extract the semantic information of the text, TPDM can better avoid the interference of complex backgrounds. Our method achieves the most advanced performance in both text detection tasks and end-to-end text recognition tasks. However, the image background with high similarity to the text still cannot accurately remove the interference. Therefore, text spotting in a complex background is one of the future work in text spotting in natural scenes.

## REFERENCES

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," 2018, *arXiv:1811.04256*. [Online]. Available: http://arxiv.org/abs/1811.04256

[2] H. Li, P. Wang, and C. Shen, "Towards End-to-End text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5238–5246.

[3] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[4] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5020–5029.

[5] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, and E. Ding, "TextNet: Irregular text reading from images with an end-to-end trainable network," in *Proc. Asian Conf. Comput. Vis.* Perth, WA, Australia: Springer, 2018, pp. 83–99.

[6] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 67–83.

[7] S. Qin, A. Bissaco, M. Raptis, Y. Fujii, and Y. Xiao, "Towards unconstrained end-to-end text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4704–4714.

[8] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10552–10561.

[9] 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.

[10] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3482–3490.

[11] 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. 5551–5560.

[12] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 577–584.

[13] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 35–48.

[14] W. Liu, C. Chen, K.-Y. Wong, Z. Su, and J. Han, "STAR-net: A SpaTial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2016, p. 7.

[15] 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.

[16] Y. Gao, Y. Chen, J. Wang, M. Tang, and H. Lu, "Reading scene text with fully convolutional sequence modeling," *Neurocomputing*, vol. 339, pp. 161–170, Apr. 2019.

[17] F. Yin, Y.-C. Wu, X.-Y. Zhang, and C.-L. Liu, "Scene text recognition with sliding convolutional character models," 2017, *arXiv:1709.01727*. [Online]. Available: http://arxiv.org/abs/1709.01727

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[19] Z. Liu, Y. Li, F. Ren, W. Goh, and H. Yu, "Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network," in *Proc. AAAI*, 2018, pp. 7194–7201.

[20] 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)*, Jun. 2016, pp. 4168–4176.

[21] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4161–4167.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[23] 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.

[24] 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.

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

[26] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[27] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 26, 2019, doi: 10.1109/TPAMI.2019.2937086.

[28] H. Wang, P. Lu, H. Zhang, M. Yang, X. Bai, Y. Xu, M. He, Y. Wang, and W. Liu, "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proc. AAAI*, 2020, pp. 1–9.

[29] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 935–942.

[30] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9075–9084.

[31] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," 2017, *arXiv:1712.02170*. [Online]. Available: http://arxiv.org/abs/1712.02170

[32] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–35.

[33] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," 2020, *arXiv:2002.10200*. [Online]. Available: http://arxiv.org/abs/2002.10200

[34] 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.

[35] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting word annotations for character based text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4940–4949.

[36] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5909–5918.

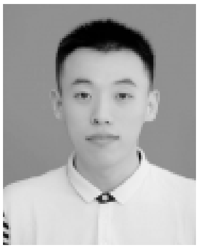
[37] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

[38] M. Bušta, Y. Patel, and J. Matas, "E2E-MLT—An unconstrained end-to-end method for multi-language scene text," 2018, *arXiv:1801.09919*. [Online]. Available: http://arxiv.org/abs/1801.09919

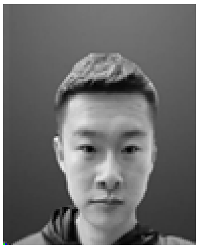
**GUANGCUN WEI** (Member, IEEE) received the B.S. degree from Shandong University (formerly Shandong University of Technology), Jinan, China, in 1994, and the M.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2006. He is currently an Associate Professor with the College of Intelligent Equipment, Shandong University of Science and Technology. He has authored over ten articles in journals and conference proceedings. He has led or participated in many projects supported by the National Innovation Fund for Small and Medium-sized Technology-based Firms, the Science Foundation of Shandong, the Key Research Program of Shandong Province, and other important projects. His research interests include the Internet of Things, software engineering, artificial intelligence, and computer vision. He was a recipient of the Science and Technology Advancement Awards at the Shandong province power, the Soft Science Award at the province, Shandong, China, and the Google Teacher Award, in 2018.

**WANSHENG RONG** was born in Shandong, China, in 1993. He received the B.S. degree from Liaocheng University, China, in 2018. He is currently pursuing the M.S. degree with the Shandong University of Science and Technology. His research interests include image processing and deep learning.

**YONGQUAN LIANG** was born in Shandong, China. He received the B.Sc. degree in applied mathematics and software from the Shandong Institute of Mining, in 1989, the M.Sc. degree in computer software from Beihang University, in 1992, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 1999. From 1989 to 1998, he was a Lecturer with the Department of Applied Mathematics and Software, Shandong Institute of Mining. From 1998 to 2001, he was an Assistant Professor with the College Computer Science and Engineering, Shandong University of Science and Technology, where he has been a Professor, since 2001, and also the Dean of the College of Computer Science and Engineering. His research interests include artificial intelligence, machine learning, data mining, cloud computing, big data analytics, and decision making.



**XINGUANG XIAO** was born in Shandong, China, in 1995. He received the B.S. degree from the Shandong University of Science and Technology, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include image processing and deep learning.



**XIANG LIU** was born in Qingdao, China, in 1995. He received the B.S. degree from the Taishan Institute of Technology, Shandong University of Science and Technology, China, in 2018. He is currently pursuing the M.S. degree with the Shandong University of Science and Technology. His research interests include deep learning, neural networks, computer vision, and other artificial intelligence applications.