Received December 21, 2020, accepted February 1, 2021, date of publication February 4, 2021, date of current version February 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3057108

A Robust and Effective Text Detector Supervised by Contrastive Learning

RAN WEI[1], YAOYI LI[1], (Member, IEEE), HAIYAN LI[1], ZE TANG[1], HONGTAO LU[1,2], (Member, IEEE), NENGBIN CAI[3], AND XUEJUN ZHAO[3]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2]MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China
[3]Shanghai Key Laboratory of Crime Scene Evidence, Institute of Forensic Science, Shanghai Public Security Bureau, Shanghai 200083, China

Corresponding authors: Hongtao Lu (htlu@sjtu.edu.cn) and Nengbin Cai (13162056906@163.com)

This work was supported in part by the NSFC under Grant 61772330, Grant 61533012, and Grant 61876109; in part by the Shanghai Key Laboratory of Crime Scene Evidence under Grant 2017XCWZK01, and in part by the Interdisciplinary Program of Shanghai Jiao Tong University under Grant YG2019QNA09.

ABSTRACT Scene text detection is a task that detects the position of text in natural scenes. Due to the different sizes, arbitrary orientations, different colors of texts, as well as low contrast and resolution in the complex background, text detection in natural scene images is very challenging. So far, the detection results for text instances in motion blur, low-resolution images are still not satisfactory. In this paper, in order to solve the above problems, we propose an effective and robust text detection network that combines a state-of-the-art contrastive learning method SimCLR. Before being input to the feature extractor, the data is augmented in different methods, and then we calculate the similarity of the extracted corresponding feature pairs. This can significantly improve the performance of the detector in difficult conditions. We conduct a series of experiments on the public dataset ICDAR2013, ICDAR2015 and MSRA-TD500. On the ICDAR 2015 dataset, our method achieves F-measure of 0.840 and runs at 9.1 FPS at 720p resolution, demonstrating that the proposed method is effective and efficient.

INDEX TERMS Scene text detection, contrastive learning, data augmentation.

I. INTRODUCTION

Text is the essential medium for the human to transmit information, and it can be seen everywhere in natural scenes and contains rich and vital semantic information. Therefore, the detection and recognition of text in natural scenes could greatly promote the image understanding and processing. In recent years, the detection and recognition of text information in natural scenes have gradually become one of the most concerned research fields. As the prerequisite for text recognition, text detection aims to locate text boundaries. In the past ten years, many excellent methods [1]–[9] have been proposed to detect text in natural scenes.

Traditional text detection methods are usually based on sliding windows [2]–[4] or connected component extraction [5]–[9]. With the rapid development of deep learning, text detection methods based on it [11]–[16] sprang up. By training a deep neural network to extract features, instead of manually designing features, text detection accuracy is greatly improved. However, due to the huge differences in text size, orientation, color and contrast, resolution in complex background, text detection in natural scene images is still challenging. In particular, we find that if there is blur or the low contrast between the text and the background, the detection performance of the text area will be significantly worse.

In order to cope with the above problems, we propose a text detection network based on EAST proposed by Zhou *et al.* [17]. We use the commonly used VGG-16 network [18] to extract multi-scale features on the input image, and then use unpooling to enlarge the feature maps of the highest layer and sequentially merge the feature maps of corresponding scales. Merging different scales of features has the advantage of decreasing the computation while containing semantic information of various scales. However, there will still be some problems. For example, this merged feature is not effective in detecting large-size text instances, and the text in blurry and jittery images can not be well detected. Inspired

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

by self-supervised learning methods, especially SimCLR proposed by Chen *et al.* [19], we solve the above problems by performing pairwise augmentation of the image, including random color jittering, random Gaussian blur and random grayscale. And then, we calculate the similarity of the extracted corresponding feature pairs. The added contrastive learning branch effectively optimizes the accuracy of the detection module.

Our contributions are summarized as follows:

1) We propose an end-to-end trained natural scene text detection network. Our network architecture incorporates the structure of EAST and contrastive learning methods, which can make full use of multi-scale image features and significantly improve the effect of blur and low-contrast text detection.

2) As far as we know, the contrastive learning method is mostly used in Natural Language Processing (NLP), image classification and other tasks before. We apply this method to text detection tasks and make significant improvements.

3) We conduct a series of experiments to evaluate our model. On the ICDAR2013 dataset, our method achieves 0.904 precision, 0.888 recall and 0.896 F-measure. On the ICDAR2015 dataset, our model achieves 0.840 F-measure scores and run at 9.1 FPS, and on the MSRA-TD500 dataset, it achieves 0.784 F-measure scores, which proves the effectiveness and robustness of the model.

II. RELATED WORK

A. TEXT DETECTION

For a long period of time, scene text detection and recognition in natural scenes have been popular research topics in computer vision. Researchers have investigated a number of inspiring ideas and effective strategies [6], [16], [21]–[29]. Systematic comments and exhaustive analysis can be found in the survey papers [30]–[32]. Previous work on text detection can be roughly divided into two categories: the one is conventional methods, extracting features of scene text manually, which is based on sliding windows [2]–[4] or connected components [5]–[9], and the other is based on deep learning methods. The sliding window-based methods move a sliding window on each position of the image to detect text. The methods based on connected components first extract character candidates and then perform post-processing to eliminate non-text noise and connect those extracted candidates.

In recent years, deep learning has achieved rapid development, and text detection algorithms based on it have gradually become the mainstream of text detection. Furthermore, the widespread application of convolutional neural networks (CNN) has greatly promoted the development of text detection [14], [21], [25], [26], [33]–[35]. Object detection and text detection have similarities. The recently proposed object detection methods are mainly divided into two categories: two-stage methods, which mainly rely on region proposal network, such as regions with CNN (R-CNN) [26], and Faster R-CNN [36]. The other is the one-stage method, which predicts the position of the object directly, such as YOLO [37] and SSD [38]. Regarding text as a special object, Huang *et al.*[25] first proposed MSER to retrieve candidate texts, followed by CNN to classify text or non-text regions. DeepText [39], based on Faster R-CNN, proposed Inception-RPN and further optimize it to adapt to text detection. Tian *et al.* [14] designed Connectionist Text Proposal Network (CTPN), which connects Long Short Term Memory networks (LSTM) to CNN, predicts a series of small-scale text components and splices text lines through context information. EAST first uses Fully Convolutional Networks (FCN) [40] to extract different scales of features and generates text predictions with rotating rectangles or quadrilaterals and then connects to its own locality-aware NMS (LANMS) algorithm to replace non-maximum suppression (NMS) to produce the final result. Yao *et al.* [28] use FCN to predict text/non-text, character classes, and character linking orientations respectively, then apply a series of processes for text detection. In order to separate adjacent text instances, PixelLink [55] performs pixel-level text/non-text and link prediction, and excludes noise by performing post-processing to obtain text boxes. TextSnake [56] uses ordered disks and text centerlines to model text instances, which is an earlier method for arbitrary text shapes.

B. SELF-SUPERVISED LEARNING

Self-supervised learning is one of the two basic learning paradigms of machine learning. Compared with supervised learning, self-supervised learning requires fewer data and labels, and it can also improve the generalization performance of the model. For each picture, the supervised learning model will make classification predictions or give region proposals based on the labeled training data, while the self-supervised learning model will learn the details of each part of the image to give more prediction information. This is consistent with the pixel-wise prediction network architecture used in this article to avoid missing possible text regions. The primary purpose of self-supervised learning is to learn a general feature expression for downstream tasks. It supervises itself, such as removing some parts of the picture and relying on the surrounding information to predict the missing patch. Currently, there are two mainstream methods for self-supervised learning: Generative Methods [41] and Contrastive Methods [42], [43]. Compared with Generative Methods, Contrastive Methods do not require the model to reconstruct the original input but to attain the goal of distinguishing different inputs in the feature space.

There has always been a problem of how to learn effective visual representations without human supervision. Chen *et al.* [19] proposed a simple framework for contrastive learning of visual representations (SimCLR)to solve it. This work proposes different data augmentation methods for an input sample. Different augmentations of the same sample are regarded as positive samples, and other different samples are regarded as negative samples. Compared with another contrastive learning method, the Momentum Contrast (MoCo)

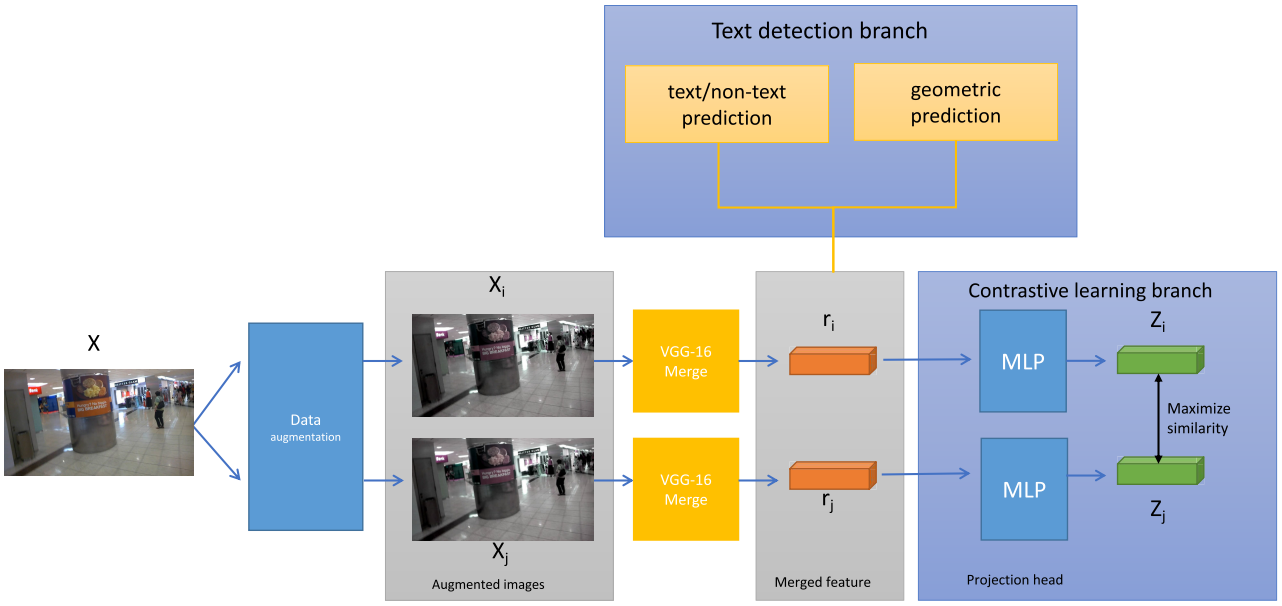


FIGURE 1. Overall structure of our method, containing text detection branch and contrastive learning branch.

proposed by He *et al.* [20], the data storage queue is omitted in SimCLR. They add a non-linear mapping between the representation layer and the final loss layer, which can greatly improve the quality of the learned representation. In addition, data augmentation is beneficial to self-supervised learning, and the different combinations of data augmentation methods will have different effects. Section IV details the results of our comparative experiments on different augmentation methods.

III. METHOD

The schematic diagram of our model is shown in Figure 1. We use EAST as the backbone network, which directly makes dense pixel-wise predictions of text lines or words in the input images and shows the corresponding geometric shapes. Compared with other methods mentioned, we simplify some steps, such as the application of the regional proposal network and the generation of text regions. Therefore, the computation cost of the entire network is much lower. Inspired by self-supervised learning and SimCLR, we add a contrastive learning branch to supervise the detection effect in order to improve the accuracy of detection.

A. NETWORK DESIGN

Since the size of the text region varies greatly in different images, feature maps of different scales are required to correspond to text instances of different sizes. More precisely, the existence of large text instances requires high-level features in neural networks, while predicting the precise geometry of smaller text instances requires early low-level information. Therefore, the proposed network must use different scales of features to meet these requirements. In previous work, the EAST network architecture satisfies this requirement well.

However, through experiments, we find that EAST is not ideal in detecting text instances in motion blurry and jittery pictures. Inspired by contrastive learning, especially the SimCLR method, we perform pairwise augmentation on the image. We hope to solve the problem above through augmentation methods such as adding random Gaussian blur and random color jittering. Then, after merging the feature, we add a contrastive learning branch, where we pass the merged feature pair through a nonlinear fully connected layer to obtain a better representation. Finally, we calculate the similarity of each pair of feature maps, which is included in the loss function for training.

B. DATA AUGMENTATION AND CONTRASTIVE LEARNING BRANCH

First, we perform two different random data augmentations on the training set image. In this case, each image sample can be randomly converted to generate two related image pairs for this sample, denoted as x_i and x_j. In the next step, we feed them into VGG-16 for feature extraction.

In this work, we successively analyze and experiment with a variety of simple enhancements: randomly cropping and resizing, horizontal flipping, random Gaussian blur, random greyscale, random color jitter. Then we conduct a series of combination comparison tests to find out the most effective data enhancement combinations among them.

After feature extraction and feature merging, we feed merged feature r_i and r_j as the representation of x_i and x_j, into the projection head to obtain the corresponding image representations z_i and z_j. The projection head here is a Multi-Layer Perceptron(MLP) with one hidden layer we use to obtain z_i, $z_i = g(r_i) = W^{(2)}\sigma W^{(1)}r_i$ where σ is a ReLU activate

function. It is beneficial to define the contrastive loss on z_i rather than r_i.

C. FEATURE EXTRACTION AND LABEL GENERATION

We respectively pass two sets of augmented related image pairs x_i and x_j through the VGG-16 network to extract four different sizes of feature maps f_1, f_2, f_3 and f_4 at different scales, which are respectively $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$ of the original image size. Then we upsample the feature map of the highest layer (f_4) and concat it with f_3. After that we use $conv_{1\times1}$ to reduce the number of channels to decrease computation, finally pass the result through the $conv_{3\times3}$ to get the merged feature map m_1, and use the same operation to get m_2 and m_3. Here, m_3 is the final merged feature map, and then we output it to the next step. In the contrastive learning branch, we flatten the m_3 of the last step to get r_i and r_j. The schematic diagram of the specific process is shown in Figure 2.

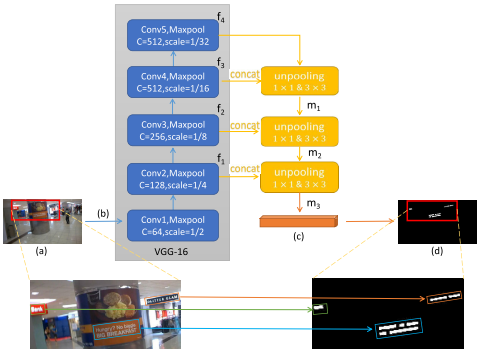


FIGURE 2. (a) is the input image. (b) stands for the augmentation. (c) is the merged feature. (d) shows the generated feature map. C in the blue boxes means number of channels. 1 × 1 in the yellow boxes stands for $conv_{1\times1}$, and it is same for 3 × 3.

Since this network is designed for multi-orientation text detection problems, we believe that the geometric shapes of text boxes are all quadrangles. Following the idea of EAST, we design the positive region of the quadrangle on the score map as a shrunk version of the original image.

For each quadrangle $Q = \{v_i \mid i \in \{1; 2; 3; 4\}\}$, where $v_i = \{x_i, y_i\}$ are vertices on the quadrangle in clockwise order. To shrink Q, we first compute a reference length l_i for each vertex v_i as:

$$l_i = \min(D(v_i, v_{(i \bmod 4)+1}), \quad D(v_i, v_{((i+2) \bmod 4)+1})) \quad (1)$$

where $D(v_i, v_j)$ is the L2 distance between v_i and v_j.Then we shrink the long edges and the short edges of the quadrangle. For each edge $\langle v_i, v_{(i \bmod 4)+1}\rangle$, we move two endpoints respectively inward along the edge by $0.3 \times l_i$ and $0.3 \times l_{(i \bmod 4)+1}$ to shrink it.

D. LOSS FUNCTION

The loss can be formulated as:

$$L = L_s + \lambda L_g + \mu L_c \quad (2)$$

where L_s represents the loss for the text score map of both x_i and x_j, L_g represents the loss for the geometry of both x_i and x_j, and L_c represents the loss for the contrastive loss respectively. λ and μ are the weight of the importance between 3 parts of loss, we set λ to 1 and μ to 1 in this paper. The specific forms of the three parts are described below.

1) LOSS FOR SCORE MAP

In some of the previous effective methods, how to design the loss function to make it closer to the data distribution has become a key part of training an excellent model. We need a suitable loss function to balance the label and background in the training image. This can significantly improve the generalization performance of the network, but the complex loss function will inevitably introduce more hyper-parameters that need to be manually set. These hyper-parameters depend heavily on human experience and even result in worse generalization ability. Therefore, the class-balanced cross-entropy and dice-loss are now more used as the loss function of the score map. In the proposed method, we use dice-loss, which has a faster training convergence rate, given by:

$$\begin{aligned} L_s &= DiceLoss(\widehat{Y}, Y^*) \\ &= 1 - \frac{2\left|\widehat{Y} \cap Y^*\right|}{\left|\widehat{Y}\right| + |Y^*|} \end{aligned} \quad (3)$$

where $\widehat{Y}$ represents the prediction of the score map, and Y^* is the ground truth.

2) LOSS FOR GEOMETRIES

Generally, the geometric map regression is calculated by using $L1$ norm or $L2$ norm as the standard measuring loss. Since the geometric map in this model merges the features of various scales, it is very hard to measure it accurately using these two methods. Because simple distance metrics cannot be used arbitrarily in features of different sizes. In order to solve this problem, we use the IoU loss [52] to measure the geometric regression between the prediction region and the ground truth region since it is invariant against targets of different scales. The formula is as follow:

$$L_g = L_e + \lambda_\theta L_\theta \quad (4)$$

The overall geometry loss consists of L_e, which stands for edge loss, and L_θ is the angle loss. We set λ_θ to 10 in our experiments. L_e is given by:

$$\begin{aligned} L_e &= -\log IoU(\widehat{R}, R^*) \\ &= -\log \frac{\left|\widehat{R} \cap R^*\right|}{\left|\widehat{R} \cup R^*\right|} \end{aligned} \quad (5)$$

where $\widehat{R}$ represents the predicted edge geometry and R^* is its corresponding ground truth. L_θ is as follow:

$$L_\theta(\hat{\theta}, \theta^*) = 1 - \cos(\hat{\theta} - \theta^*) \quad (6)$$

where $\hat{\theta}$ is the prediction of the rotation angle and θ^* represents the ground truth.

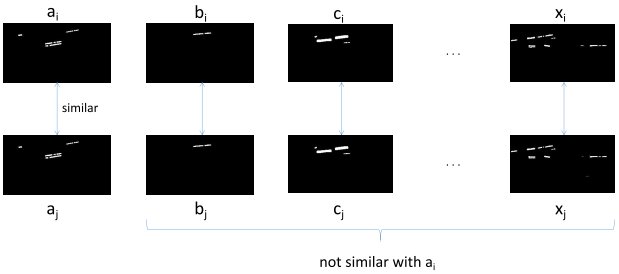


FIGURE 3. Positive and negative samples in one mini-batch.

3) LOSS FOR CONTRASTIVE LEARNING BRANCH

In this part we continue to use the normalized temperature-scaled cross-entropy loss proposed in SimCLR, given by:

$$L = \frac{1}{2N}\sum_{k=1}^{N}[l(2k-1,2k) + l(2k,2k-1)] \tag{7}$$

We randomly sample a mini-batch of N samples each time and define a comparison prediction task on the paired augmented instance pairs derived from the mini-batch to obtain $2N$ data. Similar to SimCLR, we do not sample negative examples, but take the other $2(N-1)$ instances in the mini-batch as negative examples. As Figure 3 shows, there is only one similar feature map a_j and all the others are negative samples in each mini-batch for a_i. This is same for a_j, b_i, b_j, c_i, c_j, $\ldots$, x_j.

We use cosine similarity to calculate the similarity between two augmented images, given by:

$$sim(a,b) = \frac{a^{\top}b}{\|a\|\|b\|} \tag{8}$$

Then the loss function for a positive pair of examples (i, j) is defined as:

$$l_{i,j} = -\log\frac{\exp(sim(z_i,z_j)/\tau)}{\sum_{k=1}^{2N} 1_{(k\neq i)}\exp(sim(z_i,z_k)/\tau)} \tag{9}$$

where $1_{(k\neq i)} \in \{0,1\}$ is an indicator function evaluating to 1 if $k \neq i$ and τ denotes a temperature parameter. The final loss is computed across all positive pairs, both(i, j) and (j, i), in a mini-batch.

IV. EXPERIMENTS

We compare the proposed method with recent algorithms and conduct quantitative and qualitative experiments on the public benchmark: ICDAR2013, ICDAR2015 and MSRA-TD500. All the training data we use is available publicly.

A. IMPLEMENT DETAILS

We use Pytorch [45] to implement our method. The feature extraction part of the proposed method is based on VGG-16, we use its pre-trained model on ImageNet[54]. We use Radam [46] and Lookahead [47] to optimize our model with the learning rate set to $1e^{-3}$. We first train our text detection branch for 100 iterations, and then the whole model for 700 iterations.

All our experiments are conducted on a server (CPU: Intel E5-2620 v4 @ 2.10GHz; GPU: NVIDIA GTX1080 TI; RAM: 64GB). We train our model with the batch size of 20 on 2 GPUs in parallel and evaluate our model on 1 GPU.

B. BENCHMARK DATASETS

ICDAR2013 [59] is the dataset proposed in Challenge 2 of the 2013 Robust Reading Competition. It contains 229 training images and 233 testing images in different resolutions. Most images focus on horizontal texts and also have some slightly oriented texts.

ICDAR2015 [44] is the dataset used in Challenge 4 of the ICDAR 2015 Robust Reading Competition. It consists of 1500 images, 1000 of which are used for training and the other 500 are for testing. The text regions are annotated by four vertices of the quadrangle. These images are taken incidentally by Google Glass. Therefore, the text in the scene may be in arbitrary orientations, motion blur and lower resolution.

MSRA-TD500 [57] comprises 300 training images and 200 test images. It is a dataset with arbitrary-oriented and long text lines. Different from the ICDAR2015 dataset, it contains both English and Chinese text instances. Since its training images are too few to learn a deep network, we follow the previous works [17], [55], [56] to harness 400 images from HUST-TR400 [58] as training data.

C. EVALUATION PROTOCOL

Text detection is usually divided into character-level detection and word-level detection. The ICDAR2015 dataset uses word-level detection. In order to verify the effectiveness of our model, we used its standard as the verification standard, which is based on the notions of precision and recall. Precision is defined as the number of correct estimates divided by the total number of estimates:

$$Precision = \frac{\sum_{r_d\in D} m(r_d, G)}{|D|} \tag{10}$$

where r means a rectangle in the set, G represents a ground-truth set of targets and D stands for a set of detection results of proposed model. For text detection, it is unrealistic to expect the model to be exactly the same as the bounding rectangle of the word identified by the human marker. The match between two matching bounding boxes is defined as the area of intersection divided by the union of area of two bounding boxes (IOU). For identical bounding boxes, the value of IOU is 1, and for bounding boxes that do not have intersection, the value of IOU is 0. For each rectangle in D, we find the closest match in G, and vice versa.

$$m(r,R) = IOU(r,r')|r' \in R \tag{11}$$

$m(r, R)$ stands for the best match for a rectangle r in a set of bounding boxes R. Recall is defined as the number of correct detections divided by the total number of targets:

$$Recall = \frac{\sum_{r_g\in G} m(r_g, D)}{|G|} \tag{12}$$

FIGURE 4. Some successful detection examples of our method on ICDAR2013 dataset.

Precision and Recall evaluate the model from two aspects. A model that over-estimates bounding boxes will be penalized by a lower Precision score. A model that under-estimates the number of bounding boxes will be penalized by a lower Recall score.

In order to facilitate representation and unify standards, we use the standard F-measure to combine the precision and recall numbers into a single measure of quality.

$$\text{F-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

D. RESULTS ON ICDAR2013

We test our model on the ICDAR2013 dataset to evaluate its performance to detect horizontal texts. For a fair comparison, we also mix the 1000 training pictures of ICDAR2015 and the training pictures of ICDAR2013 together as the training set. Our method achieve 0.904 precision, 0.888 recall and 0.896 F-measure.

Figure 4 shows a set of examples of successful text detection on the ICDAR2013 dataset. It can be seen that the dense text can be well detected by our method, while the text-like patterns and textures are not misdetected. Successful detection in dark and reflective places proves the robustness of the model.

E. RESULTS ON ICDAR2015

We conduct quantitative and qualitative experiments on the authoritative ICDAR2015 data set to verify the effectiveness and robustness of our method in multi-oriented text detection tasks. The evaluation criteria of the ICDAR2015 dataset uses a quadrilateral box to test the model's ability to detect multi-orientation text. Moreover, the background of the images in the ICDAR2015 data set is more complex, and there are more interference conditions, such as illumination and blur caused by motion, object occlusion. Therefore, this dataset is a huge challenge to the text detection model. It can be seen from Table 1 that we achieve 0.860 precision, 0.822 recall and 0.840 F-measure. Among them, recall and F-measure are better than other methods, the precision is competitive.

TABLE 1. Results on ICDAR2013. MS stands for multi-scale testing.

Method	Precision	Recall	F-measure
Ours	0.904	**0.888**	**0.896**
PixelLink (MS) [55]	0.886	0.875	0.881
Textboxes++ (MS) [13]	0.91	0.84	0.88
SSTD [50]	0.89	0.86	0.88
SegLink (MS) [35]	0.920	0.844	0.88
CTPN [14]	**0.930**	0.830	0.877
EAST+PVANET 2x RBOX (MS) [17]	0.926	0.827	0.874
Gao *et al.* [60]	0.90	0.80	0.85

Figure 5 illustrates some detection results on the test set. It can be seen that whether they are traffic signs on the roadside, tiny texts on the subway, shopping mall texts with motion blur, or texts with low contrast and dark light, our model can accurately detect the texts and locate them by bounding boxes.

A set of comparison charts with other methods is shown in Figure 6. As can be seen, our proposed method can accurately locate the text and distinguish each word correctly in the small and dense texts (a); with poor light conditions and object occlusion (b), our model does not produce missing detection; when in scene (c), our detector successfully locates all the text, including the blurred text on the wall and the reflection on the ground; in (d), the proposed model performs well, while other methods made more or less missing detections. It proves that our method can accurately and robustly handle difficult cases, such as object occlusion, dark light, low contrast, motion blur, and jitter when taking pictures.

FIGURE 5. Some successful detection examples of our method on ICDAR2015 dataset.

FIGURE 6. Qualitative comparisons of text detection results on ICDAR2015 with other methods. (a) Our methods (b) EAST (c) PAN (d) CTPN.

F. RESULTS ON MSRA-TD500

In order to verify the robustness of our method in long text and mixed language (English and Chinese) text detection, we perform an evaluation on the MSRA-TD500 dataset. As can be seen in Table 2 that we achieve the highest recall (0.7873) and F-measure (0.7840) among listed methods.

Figure 7 shows some of the test results of our method on the MSRA-TD500 dataset. There is usually a larger gap between

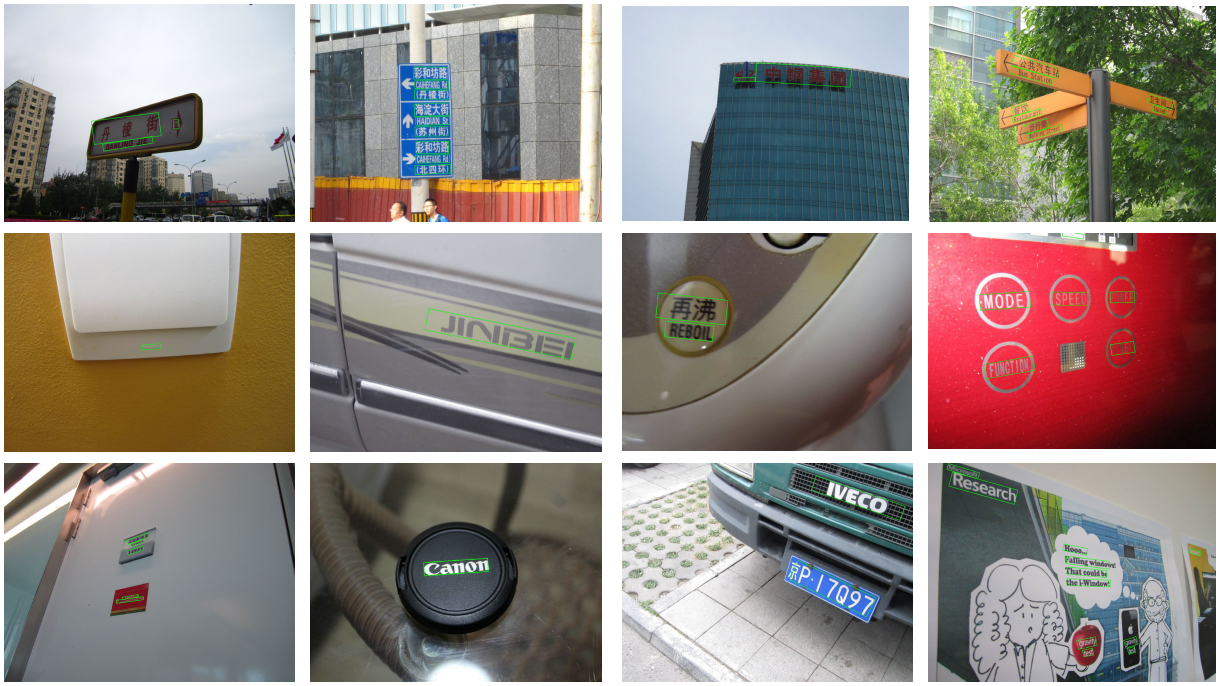


FIGURE 7. Some successful detection examples of our method on MSRA-TD500 dataset.

TABLE 2. Results on ICDAR2015. MS stands for multi-scale testing.

Method	Precision	Recall	F-measure
Ours	0.860	**0.822**	**0.840**
Qin *et al.* [62]	0.868	0.798	0.832
PAN [53]	0.840	0.819	0.829
Textboxes++ (MS) [13]	**0.878**	0.785	0.829
Jiang *et al.* [61]	0.858	0.797	0.826
R2CNN [49]	0.8562	0.7968	0.8254
DRFCN [48]	0.82	0.80	0.81
EAST+PVANET 2x RBOX (MS) [17]	0.8327	0.7833	0.8072
Gao *et al.* [60]	0.804	0.784	0.794
WordSup [51]	0.793	0.770	0.782
EAST+VGG16 RBOX [17]	0.8046	0.7275	0.7641
RRPN [12]	0.7323	0.8217	0.7744
SSTD [50]	0.8023	0.7386	0.7691
SegLink [35]	0.731	0.768	0.75
CTPN [14]	0.742	0.516	0.609
baseline	0.660	0.447	0.533

Chinese characters than between English words, which is a difficult point in detecting Chinese text lines. It can be seen that our method successfully detects both Chinese and English text lines.

G. DIFFERENT AUGMENTATION METHODS

Different from the classification tasks, we take into account the particularity of the text location in the text detection, and remove some improper augmentation methods, such as random cropping and horizontal flipping. The reason is that after these two augmentations, the feature regions on the representation will be hardly corresponding, which leads to a low similarity between z_i and z_j. These low similarity pairs will have a negative impact on the contrastive learning branch. In order to further explore, we also conduct these two augmentations experiments. The experimental results confirm our assumptions. Therefore, we use different combinations of random color jittering, random Gaussian blur, and random grayscale to conduct comparative experiments.

As shown in Table 2, when using random color jittering alone as the data augmentation method, we achieve the best experimental results, on average more than 1% higher than other methods. On the other hand, when using random Gaussian blur + random grayscale (without random color jittering) as the augmentation method, we achieve the worst experimental results, which verify the effectiveness of random color jittering from the side. It also verifies that the contrastive learning branch significantly improves the effect of text detection.

TABLE 3. Results on MSRA-TD500.

Method	Precision	Recall	F-measure
Ours	0.7807	**0.7873**	**0.7840**
TextSnake [56]	0.832	0.739	0.783
PixelLink [55]	0.830	0.732	0.778
SegLink [35]	0.860	0.700	0.770
EAST+PVANET 2x [17]	**0.8728**	0.6743	0.7608
Yao *et al.* [28]	0.7651	0.7531	0.7591
Gao *et al.* [60]	0.86	0.67	0.75
RRPN [12]	0.820	0.680	0.740
DeepReg [48]	0.770	0.700	0.740
EAST+VGG16 [17]	0.8167	0.6160	0.7023

H. SPEED COMPARISON

The result of speed comparison is demonstrated in Table 3. We use our best-performing networks to run through the ICDAR 2015 test set (containing 500 images with 1280×720 resolution) and report the average speed. These experiments

are conducted on a server using a single NVIDIA GTX1080 TI graphic card and Intel E5-2620 v4. For other methods, we directly list the results of their original papers. As is shown in Table 3, our model runs at 9.1 FPS, which is the fastest except TextBoxes++. While maintaining the highest F-measure score, the speed of our proposed model is still competitive.

TABLE 4. Results of different augmentation methods on ICDAR2015 dataset. "CJ" represents random color jittering. "GB" stands for random Gaussian blur. "GS" means random grayscale.

Augmentation	Precision	Recall	F-measure
CJ	**0.860**	**0.822**	**0.840**
GB	0.849	0.809	0.829
GS	0.836	0.816	0.826
CJ + GB	0.838	0.819	0.828
CJ + GS	0.834	0.821	0.827
GB + GS	0.840	0.806	0.823
CJ + GB + GS	0.838	0.816	0.827

TABLE 5. Results of speed comparison.

Method	F-measure	FPS
Ours	**0.840**	9.1
Textboxes++(MS)[13]	0.829	2.3
Textboxes++[13]	0.817	**11.6**
WordSup[51]	0.782	2
SSTD[50]	0.770	7.7
EAST VGG16[17]	0.764	6.52
SegLink[35]	0.75	8.9
CTPN[14]	0.609	7.1
baseline	0.533	4.5

V. CONCLUSION

In this paper, we propose a text detection model with a contrastive learning branch. We make full use of the feature pyramid in FPN to extract multi-scale features, then upsample and merge them, and finally use the similarity calculated by the contrastive learning branch to optimize the effect of the detection branch. The network can be used for multi-orientation, multi-scale scene text detection tasks and achieves good scores on multiple public datasets. However, the model can still be improved. One is that in addition to straight lines, text in natural scenes also has text arranged in curves. Since the label generated of this model only contains positions and rotation angles of 4 vertices, our method is not suitable for curved text. In the follow-up plan, we will study more deeply and improve the method to adapt to this situation. In addition, we only use English and Chinese datasets to train and validate our method in this period. We plan to improve the model in the future and challenge more multilingual datasets.

REFERENCES

[1] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.

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

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

[4] J. Yan, J. Li, and X. Gao, "Chinese text location under complex background using Gabor filter and SVM," *Neurocomputing*, vol. 74, no. 17, pp. 2998–3008, Oct. 2011.

[5] 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.*, Jun. 2010, pp. 2963–2970.

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

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

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

[9] Y. Liu, "A contour-based robust algorithm for text detection in color images," *IEICE Trans. Inf. Syst.*, vol. 89, no. 3, pp. 1221–1230, Mar. 2006.

[10] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1962–1969.

[11] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," 2016, *arXiv:1611.06779*. [Online]. Available: http://arxiv.org/abs/1611.06779

[12] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*. [Online]. Available: http://arxiv.org/abs/2002.05709

[20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[21] 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.

[22] A. Mishra, K. Alahari, and C. V. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2687–2694.

[23] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.

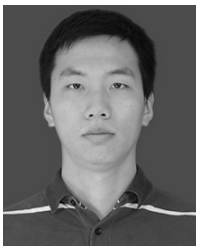
[24] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 752–765.

[25] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 497–511.

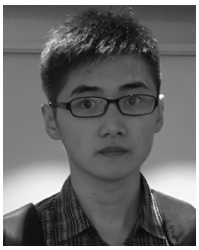
[26] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, Jan. 2016.

[27] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild, "Toward integrated scene text reading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 375–387, Feb. 2014.

[28] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016, *arXiv:1606.09002*. [Online]. Available: http://arxiv.org/abs/1606.09002
[29] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4042–4049.
[30] S. Uchida, "25–text localization and recognition in images and video," in *Handbook of Document Image Processing and Recognition*. 2014, pp. 843–883.
[31] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.
[32] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 19–36, Feb. 2016.
[33] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 440–445.
[34] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 512–528.
[35] 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. 2550–2558.
[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[37] 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.
[38] 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.
[39] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A unified framework for text proposal generation and text detection in natural images," 2016, *arXiv:1605.07314*. [Online]. Available: http://arxiv.org/abs/1605.07314
[40] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[41] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 649–666.
[42] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748
[43] R. Devon Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*. [Online]. Available: http://arxiv.org/abs/1808.06670
[44] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.
[45] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*. [Online]. Available: https://arxiv.org/abs/1912.01703
[46] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," 2019, *arXiv:1908.03265*. [Online]. Available: http://arxiv.org/abs/1908.03265
[47] M. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead optimizer: K steps forward, 1 step back," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9597–9608.
[48] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 745–753.
[49] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: http://arxiv.org/abs/1706.09579
[50] 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)*, Oct. 2017, pp. 3047–3055.
[51] 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.
[52] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 516–520.
[53] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8440–8449.
[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[55] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," in *Proc. AAAI*, Apr. 2018, vol. 32, no. 1.
[56] 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.*, 2018, pp. 20–36.
[57] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.
[58] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.
[59] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.
[60] J. Gao, Q. Wang, and Y. Yuan, "Convolutional regression network for multi-oriented text detection," *IEEE Access*, vol. 7, pp. 96424–96433, 2019.
[61] X. Jiang, S. Xu, S. Zhang, and S. Cao, "Arbitrary-shaped text detection with adaptive text region representation," *IEEE Access*, vol. 8, pp. 102106–102118, 2020.
[62] X. Qin, J. Jiang, C.-A. Yuan, S. Qiao, and W. Fan, "Arbitrary shape natural scene text detection method based on soft attention mechanism and dilated convolution," *IEEE Access*, vol. 8, pp. 122685–122694, 2020.



RAN WEI received the B.E. degree from Information Engineering University, China, in 2014. He is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include computer vision and deep learning.



YAOYI LI (Member, IEEE) received the B.E. degree from the Department of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include computer vision and deep learning.

HAIYAN LI received the M.S. degree in computer science from Xinjiang University, China, in 2007. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include computer vision, deep learning, and machine learning.

ZE TANG is currently pursuing the B.E. degree with the ACM Class, Shanghai Jiao Tong University. His research interests include computer vision and deep learning.

HONGTAO LU (Member, IEEE) is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He has authored or coauthored more than 100 papers in journals and premier conferences. His current research interests include computer vision, deep learning, and machine learning.

NENGBIN CAI is currently a Senior Engineer with the Criminal Technology Center of the Criminal Investigation Corps, Shanghai Public Security Bureau.

XUEJUN ZHAO is currently an Assistant Researcher with the Shanghai Institute of Criminal Science and Technology.