Hindawi
Complexity
Volume 2021, Article ID 9958333, 8 pages
https://doi.org/10.1155/2021/9958333

Research Article

AEMF: An Attention-Based Efficient and Multifeature Fast Text Detector

Wanqi Ma,[1] **Chaoyu Yang**,[1] **Jie Yang**,[2] **and Jian Wu**[1]

[1]*School of Economics and Management, Anhui University of Science and Technology, Huainan 232001, China*
[2]*School of Computing and Information Technology, Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, NSW 2522, Australia*

Correspondence should be addressed to Chaoyu Yang; yangchy@aust.edu.cn

Received 11 March 2021; Accepted 30 June 2021; Published 12 July 2021

Academic Editor: Huihua Chen

The label from industrial commodity packaging usually contains important data, such as production date, manufacturer, and other commodity-related information. As such, those labels are essential for consumers to purchase goods, help commodity supervision, and reveal potential product safety problems. Consequently, packaging label detection, as the prerequisite for product label identification, becomes a very useful application, which has achieved promising results in the past decades. Yet, in complex industrial scenarios, traditional detection methods are often unable to meet the requirements, which suffer from many problems of low accuracy and efficiency. In this paper, we propose a multifeature fast and attention-based algorithm using a combination of area suggestion and semantic segmentation. This algorithm is an attention-based efficient and multifeature fast text detector (termed AEMF). The proposed approach is formed by fusing segmentation branches and detection branches with each other. Based on the original algorithm that can only detect text in any direction, it is possible to detect different shapes with a better accuracy. Meanwhile, the algorithm also works better on long-text detection. The algorithm was evaluated using ICDAR2015, CTW1500, and MSRA-TD500 public datasets. The experimental results show that the proposed multifeature fusion with self-attention module makes the algorithm more accurate and efficient than existing algorithms. On the MSRA-TD500 dataset, the AEMF algorithm has an F-measure of 72.3% and a frame per second (FPS) of 8. On the CTW1500 dataset, the AEMF algorithm has an F-measure of 62.3% and an FPS of 23. In particular, the AEMF algorithm has achieved an F-measure of 79.3% and an FPS of 16 on the ICDAR2015 dataset, demonstrating the excellent performance in detecting label text on industrial packaging.

1. Introduction

In recent years, with the continuous development of industry, the quality and safety of commodities have attracted more and more attention. Therefore, textual information from the product labels needs to be recognized. In particular, the information of production date, manufacturer, and location is essential indicators for the sampling supervision and management agency to monitor or regulate the product production standards. As such, the text detection of industrial product labels plays a significant role in ensuring industrial products' safety supervision and protection.

As for the reading of characters from product labels, the traditional method applies manual labor by the experienced workers, while label data are then manually input into the computer. This detection method is inefficient, labor intensive, and contains many errors, which fails to meet the needs of assembly line operations and automated production development. With the emergence of automate technology, such as deep learning, the label detection automation becomes reliable and efficient. Text detection technology is essential in the analysis and extraction of image information. The key lies in distinguishing and locating the complex text region and separating them from background area. Traditional text detection algorithms usually employ hand-crafted features such as edge gradients, directional gradient histograms, and local binarization to classify candidate regions into text and nontext areas. However, such hand-crafted features fail to accurately describe or capture complex textual domains in natural scenes.

Meanwhile, existing methods are also limited in terms of text localization in complex industrial environments.

To overcome these issues, this paper proposes a novel detection algorithm based on a combination of region suggestion and semantic segmentation, through multimodel integration. While the former is used to predict candidate text boxes, the latter is to detect the candidate text region. At last, we generate the prediction results by aggregating both results to obtain the final text box. The contribution of this paper is then summarized as follows:

(i) Compared with the existing algorithm that performs poorly for long texts, the proposed algorithm increases the number of channels in the back convolution layer, which improves the detection capability for long texts.

(ii) The algorithm consists of the segmentation branch and the detection branch. More precisely, the segmentation branch is based on self-attention strategy, while the detection branch utilizes the multiresolution feature fusion [8]. Finally, the mutual fusion of the two branches is used to detect the text in the image more quickly and efficiently.

(iii) The proposed algorithm combines area suggestion and semantic segmentation for test detection, thereby improving the detection accuracy and speed.

2. Related Work

Scene text detection receives more and more attention in computer vision, with the technology of text detection being introduced into industrial product label inspection. Currently, many methods have been proposed for the label detection task, which are mainly divided into two aspects: character localization (based on area suggestion) and character positioning (using semantic segmentation). As for the natural scene text detection, a network structure is proposed based on Faster-RCNN [2]. The algorithm consists of rotation region proposal (RRP), rotation region-of-interest pooling (RRoIP), and intersection over union (IoU) to enhance the detection of tilted text; later, a text-attentional CNN is applied to improve the classification of candidate text components by extracting convolutional features from the candidate text component regions instead of hand-crafted features. Another work from [3] utilizes high and low dimensional features to simultaneously detect text segments of different sizes, which improves the detection robustness for both large and small texts. In addition, Liao et al. [11] proposed an adjustable network structure (TextBoxes) for text detection with different size ratios. The network structure can effectively detect different font sizes according to the multiscale features of varied convolutional layers.

However, most of the existing methods, based on text region suggestions or text component suggestions, are limited to detect text fields in fixed rectangular or quadrilateral regions [5]. Another problem is that texts in natural scenes usually come with different shapes, font sizes (aspect ratio), and candidate frame sizes. Yet, traditional methods are unable to approximately match actual text instances, while their employed frame regression can only make minor adjustments to the candidate frame's position, thereby reducing the detection performance.

Some methods have been proposed to tackle natural scene text detection using the concept of semantic segmentation. For instance, Zhang and Zhang [16] used fully convolutional networks (FCNs) to predict the text segmentation map and then applied the MSER text detection method to filter and obtain more accurate textual contents; Zhou [10] proposed the EAST model using upsampling strategy to fuse the features from different layers to generate the global feature map, which is helpful to detect the text box of the size scale; the PixelLink text detection model is proposed in [17], which generates the text location using the most minor circumscribed matrix from minAreaRect; Xu and Wang [20] used the text direction field to represent the image feature and proposed a detection model for the irregular text vector field. The main idea is to obtain the representative pixels of each text instance before expanding finally to obtain text instances. This method improves the accuracy of text detection from multiple angles. Xue and Lu [19] designed a multiscale shape regression prediction model. To begin with, a dense text instance boundary is obtained to locate text in different directions and shapes. However, the drawback is that it fails to locate the text box's position utilizing semantic segmentation accurately.

3. Main Approach

Our main motivation for this work is to improve the performance of text segmentation and character detection through joint training. First, we describe the network structure of the proposed algorithm. Then, we propose the self-attention mechanism under the segmentation branch. Subsequently, we propose the multiresolution feature fusion under the detection branch. Finally, the objective function is formulated by fusing the two branches.

3.1. Network Architecture. The proposed method consists of two parts, including regional suggestion and semantic segmentation. Accordingly, the employed network structure also has two components, one for text segmentation and the other one is for character detection. In addition, the exact baseNet is used as the backbone model, while both two components are designed to share convolutional and pooling layers (see Figure 1).

The segmentation branch is similar to the EAST algorithm [10]. The problem with EAST is that the maximum text instance size is proportional to the received field of the network, which limits the ability of the network to predict longer text regions. To overcome this issue, in this paper, a self-attention module is added to each convolutional layer, which is used to effectively expand the detection field by considering the case of long text.

The detection branch is inspired by the Faster-RCNN algorithm [2]. The core process of Faster-RCNN

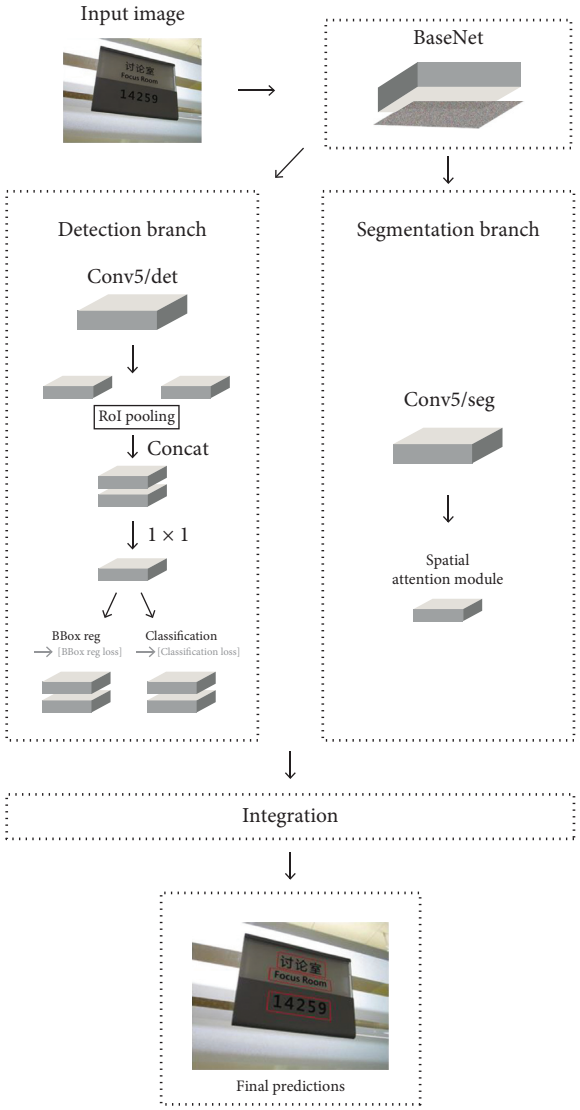

Figure 1: The proposed network structure of AEMF.

character detection is to use the anchor frame to fit the real frame through the region proposal network. The purpose is to locate a number of regions that contain characters with higher probability, before extracting the features of these regions to detect characters. The problem with Faster-RCNN is that the detection performance is determined by anchor frames and feature selection [1]. In this paper, at first, we introduce RoI (region-of-interest) pooling to aggregate information from different regions. Meanwhile, the multiresolution features are employed to extract more convolutional features of the candidate regions via avoiding misleading and false detection. Second, the detection branch is also composed of a region proposal and a segmentation part, which are used to predict character category and location simultaneously [27]. At last, the detection is determined via considering the trade-off between the current target detection and the desired text. Ultimately, the proposed AEMF algorithm combines the segmentation branch with the target detection branch.

3.2. Attention Mechanism. In the segmentation branch, we consider to increase the maximal size of text instance. By contrast, the existing algorithm, such as EAST, has the limit on the size that is proportional to the network's received fields, reducing the prediction ability to include more text regions.

To this end, our method introduces an attention mechanism [21] for extracted text and location features, to ensure the coverage of the target detection field. More precisely, the attention module is generated according to the spatial relationship between the text features. Let a spatial attention map be $M_s(F) \in R^{H\times W}$ that encodes the places to be emphasized or suppressed.

In this paper, we use two pooling operations to summarize the channel information operations of the feature maps before generating two 2D maps: $F^s_{\text{avg}} \in R^{1\times H\times W}$ and $F^s_{\max} \in R^{1\times H\times W}$, which represent the average pooling feature and the maximum pooling feature for the entire channel, respectively. Then, the 2D spatial attention map is generated by interconnecting two 2D maps and convolving them in a standard convolutional layer [12]. In short, the spatial attention is computed as follows:

$$\begin{aligned} M_s(F) &= \sigma\left(f^{7\times 7}\left([\text{AvgPool}(F); \text{Max}(F)]\right)\right), \\ M_s(F) &= \sigma\left(f^{7\times 7}\left(\left[F^s_{\text{avg}}; F^s_{\max}\right]\right)\right), \end{aligned} \tag{1}$$

where σ is the sigmoid activation function and $f^{7\times 7}$ is a convolutional operation with a convolutional kernel of 7×7 layers. The two operations of average pooling and maximum pooling are then forwarded to a shared network to produce our channel attention map $M_c \in R^{C\times 1\times 1}$, while this shared network is composed of a multilayer perceptron (MLP) with a hidden layer. To reduce the overhead of the parameters, the hidden activation size is set to $M_c \in R^{C/r\times 1\times 1}$, where r is the reduction rate [13]. In short, the channel attention is computed by first applying each descriptor through the shared network, and then the element summation method is utilized to merge the output feature vectors as follows:

$$\begin{aligned} M_c(F) &= \sigma\left(\text{MLP}\left(\text{AvgPool}(F)\right) + \text{MLP}\left(\text{MaxPool}(F)\right)\right), \\ M_c(F) &= \sigma\left(W_1\left(W_0\left(F^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\max}\right)\right)\right), \end{aligned} \tag{2}$$

where $W_0 \in R^{C/r\times C}$ and $W_1 \in R^{C\times C/r}$ are the trainable weights, respectively. Finally, in the process of feature fusion, an intermediate feature map $F \in R^{C\times H\times W}$ is used as input. Then, the final refined output F'' is calculated using the 1D channel attention map $M_c \in R^{C\times 1\times 1}$, and the 2D spatial attention map $M_c \in R^{1\times H\times W}$ is as follows:

$$\begin{aligned} F' &= M_c(F) \otimes F, \\ F'' &= M_s(F') \otimes F', \end{aligned} \tag{3}$$

where $\otimes$ is the multiplication of the corresponding matrix elements. Note that the value of the attention channel is between [0, 1], which helps to reinforce the valuable image information and suppress the useless information.

3.3. Multiresolution Feature Fusion. In the detection branch, in the region-of-interest pooling layer, the frame generated by RPN (region proposal network) is mapped from the coordinates of the input image to the conv_5 (which is the 5th convolutional layer). Then, the extracted corresponding region is divided into 7 equal parts horizontally and vertically. As a result, the maximum pooling is performed on each part to obtain a $7 \times 7 \times 512$ fixed-size region. However, conv_5 has high-level semantic information but the low resolution of the feature map, while conv_3 (the 3rd convolutional layer) has high resolution and detailed location information. If only the features of the proposed frame are extracted from the conv_5 feature map, the high resolution features in the conv_3 shallow feature map will be wasted [23]. Therefore, this paper improves the original region of the interest pooling layer by the combination of the features extracted by conv_5 and conv_3. First, the pooling operation is applied to the region of interest for conv_5 and conv_3 to obtain feature maps of size $7 \times 7 \times 512$ and $7 \times 7 \times 256$. Second, the two feature maps are stitched in the same dimension to obtain a feature map of size $7 \times 7 \times 768$. At last, we use a 1×1 convolution kernel to convolve the stitched feature map, before joining them with the operation of Concat and Add [26]. In particular, Concat feature fusion achieves superposition on the number of dimensions [25]. On the other hand, the Add feature fusion is used to increase the amount of information in each dimension, which is done by adding the corresponding feature maps before proceeding to the next convolution operation (see Figure 2). As such, the convolution of Concat and Add is shown in equations (4) and (5), respectively:

$$D_{\text{concat}} = \sum_{i=1}^{c} X_i * K_i + \sum_{i=1}^{c} Y_i * K_{i+c}, \tag{4}$$

$$D_{\text{add}} = \sum_{i=1}^{c} (X_i + Y_i) * K_i, \tag{5}$$

where D_{concat} and D_{add} are individual output channels, X_i and Y_i are input channels, and K_i is the convolution kernel of the corresponding channel.

Note that the Add feature fusion shared convolution has less number of parameters and computational effort. Furthermore, the increase in dimension can not only mitigate the gradient disappearance and enhance the feature transfer but also realize the feature reuse [4]. Therefore, Concat is used in the feature fusion part of the region of interest.

Overall, the proposed method pools region of interest in the multichannel convolutional layers, and then the feature maps are summed on the feature channel dimension via concat splicing [6]. The same convolution is performed on the stitched feature map, which combines the stitched features, high-level semantic information, and shallow detail location information [7]. This proposed convolution, as a result, helps in maintaining the size of the feature map and preserving the feature information.

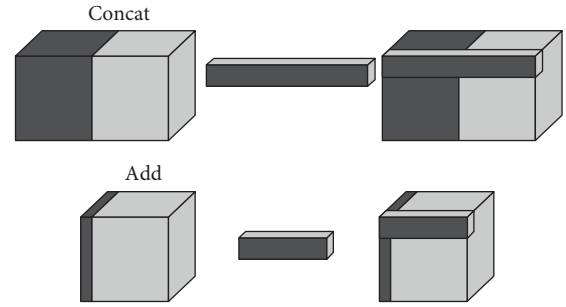

Figure 2: Feature fusion of Concat and Add.

3.4. Loss Functions. The loss function can be expressed as follows:

$$L = L_s + \lambda_g L_g, \tag{6}$$

where L_s and L_g denote the loss of fractional and geometric graphs, respectively, and λ_g expresses the importance between the two losses (in this paper, λ_g is set to 1). In most advanced detection pipelines, the problem of unbalanced target object distribution is addressed by balanced sampling and hard negation. Although this may improve network performance, it also introduces a more tuned and complex pipeline of parameters. To facilitate a simpler training process, we employ the class balanced cross-entropy [15] to have the following:

$$\begin{aligned} L_s &= \text{balanced} - \text{xent}\left(\widehat{Y}, Y^*\right) \\ &= -\beta Y^* \log\left(\widehat{Y}\right) - (1-\beta)\left(1 - Y^*\right)\log\left(1 - \widehat{Y}\right), \end{aligned} \tag{7}$$

where $\widehat{Y} = F_s$ is the predicted value of the fractional plot and Y^* is the actual underlying value. Parameter β is the balance factor between positive and negative samples:

$$\beta = 1 - \frac{\sum_{y^* \in Y^*} y^*}{\left|Y^*\right|}. \tag{8}$$

Furthermore, to generate accurate text geometry predictions for large and small text regions, we introduce the regression part of the rotating rectangular box as the IoU loss function, that is, fixed for objects of different scales. For the IoU loss, we have

$$L_R = -\log\left(\text{IoU}\left(\widehat{R}, R^*\right)\right) = -\log\frac{\left|\widehat{R} \cap R^*\right|}{\left|\widehat{R} \cup R^*\right|}, \tag{9}$$

where $\widehat{R}$ is the predicted geometry, R^* is its corresponding actual shape, and the width ω_i and height h_i of the intersecting rectangle $|\widehat{R} \cap R^*|$ are calculated as follows:

$$\begin{aligned} \omega_i &= \min\left(\widehat{d}_2, d_2^*\right) + \min\left(\widehat{d}_4, d_4^*\right), \\ h_i &= \min\left(\widehat{d}_1, d_1^*\right) + \min\left(\widehat{d}_3, d_3^*\right), \end{aligned} \tag{10}$$

where d_1, d_2, d_3, and d_4 are the distances from a pixel to the top, right, bottom, and left borders of its corresponding rectangle. The following formula gives the union zone:

$$\left|\widehat{R} \cup R^*\right| = \left|\widehat{R}\right| + \left|R^*\right| - \left|\widehat{R} \cap R^*\right|. \tag{11}$$

Next, the rotation angle loss is calculated as follows:

$$L_\theta\left(\widehat{\theta}, \theta^*\right) = 1 - \cos\left(\widehat{\theta} - \theta^*\right), \tag{12}$$

where $\widehat{\theta}$ is the prediction of the angle of rotation and θ^* is the actual value. Finally, the total geometric loss can be calculated as follows:

$$L_g = L_R + \lambda_\theta L_\theta. \tag{13}$$

In this paper, λ_θ is set to 10 during the experiment.

4. Experiments

4.1. Experimental Environment. This experiment is conducted with the following settings: hardware configuration CPU is an eight-core 16-threaded i9-9900k, with main frequency 5 GHz and memory is 32 GB; as for GPU setting, we use NVIDIA RTX 2080Ti with 11 GB of video memory.

4.2. Benchmark Datasets. Our experiment for label text detection in industrial complex environments is conducted using three public datasets, including ICDAR2015 and MSRA-TD500 and CTW1500. Their detailed description is as follows.

ICDAR2015. The data set has a total of 1500 pictures. We take 1000 of them for training and the rest for testing. This data set is provided by the incidental scene text reading competition from ICDAR in 2015. The background scene is a random street or shopping mall picture taken by Google Glass without focusing. It is designed to help text detection and recognition models improving the generalization performance.

MSRA-TD500. The data set consists of 300 training and 200 test images. It is a text detection data set provided by Huazhong University of Science and Technology in 2012. The dataset images contain pictures taken in scenes such as offices, shopping malls, and streets. The target texts are composed of Chinese and English in different directions.

CTW1500. This data set is a Chinese natural scene text image data set provided by Tsinghua University. It is collected from Tencent Street View and has a high degree of diversity. The pictures include flat text, urban street view text, township street view text, the text under weak lighting conditions, small text, and partial display text.

4.3. Backbone Networks. In this paper, we use two backbone networks for training, VGG16 and ResNet50. Among them, VGG16 comes from VGGNet [29], which is a convolutional neural network. The main contribution of VGGNet is to explore the relationship between the depth of a convolutional neural network and its performance. By repeatedly stacking 3×3 convolutional kernels, VGGNet greatly increases the speed of the network. VGGNet has six networks, A, A-LRN, B, C, D, and E. Among them, D and E are often referred to as VGG16 and VGG19. VGG16 has 5 convolutional segments, while each segment contains 2 or 3 convolutional layers, and each segment has a maximum pooling layer at the end to reduce the image size. The number of convolution kernels in each segment is the same (from front to back), which is 64-128-256-512. The perceptual field is increased by stacking multiple convolution kernels of size 3×3.

ResNet50 is part of the Residual Networks [30] (ResNet), which is a deep convolutional neural network based on residual units. Due to its simplicity and practicality, many subsequent studies have been done based on ResNet50 or ResNet-101. The advantage is that this backbone network enables to avoid gradient disappearance when the layers are very deep, making the model training and converging more easily.

4.4. Quantitative Results. The following diagrams show the example performance of the AEMF algorithm under different datasets from various environments, including the ICDAR2015 dataset (see Figure 3), the MSRA-TD500 dataset (see Figure 4), and the CTW1500 dataset (see Figure 5). As observed, the AEMF algorithm can perform text detection in ambiguous, uneven, and multilingual scenarios, as well as different angles. Clearly, the results show that the AEMF algorithm achieves a highly accurate and stable detection.

4.5. Comparison with State-of-the-Art. The following three tables show the comparison of the recall, accuracy, average score, and frames per second (FPS) for the AEMF algorithm, using the VGG16 and ResNet50-based backbone network, against existing methods. More precisely, we have R for recall, P used for Precision Accuracy, and F for F-measure.

Table 1 shows the results of text detection using the AEMF algorithm in the ICDAR2015 dataset. In terms of recall accuracy (R), we observe that our algorithm performs the best among all algorithms. It achieves 77.3% recall under the ResNet50-based backbone network, which is 3.7% higher than that of [10]. However, under the VGG16-based backbone network, our algorithm has a recall of 72.3%, which is lower than the recall of EAST and SegLink but 20.4% higher than Faster-RCNN. Meanwhile, for the Precision Accuracy (P), the proposed algorithm performs the best compared with existing algorithms, which leads to the 84.2% accuracy under the backbone network of ResNet50. We also observe a 1.5% improvement over EAST and a 10% improvement over Faster-RCNN. Compared with CTPN, the proposed algorithm also improves by 10.2% significantly. Moreover, the AEMF algorithm is higher than SegLink by 10.1%. In addition, for F-measure, the algorithm has the highest average score among all algorithms using the ResNet50 backbone network. In addition, the proposed algorithm also achieved an FPS of 16 with the ResNet50-based backbone network, which again outperforms all other algorithms.

Table 2 reports the results of all the algorithms on MSRA-TD500-based dataset. As to recall accuracy (R), we observe that our algorithm performs the best among all algorithms with 65.3% recall under the ResNet50-based

(a) (b)

Figure 3: Recognition results of the AEMF algorithm from the ICDAR2015 dataset.

(a) (b)

Figure 4: Recognition results of the AEMF algorithm from the MSRA-TD500 dataset.

(a) (b)

Figure 5: Recognition results of the AEMF algorithm from the CTW1500 dataset.

backbone network, which is 3.7% higher compared with [10]. Nevertheless, using the VGG16-based backbone network, our algorithm has a recall of 62.3%, which is higher than the recall of EAST and CTPN; for the Precision Accuracy (P), the AEMF algorithm performs the best among all comparison algorithms with 83.3% accuracy with ResNet50. We also observe a 1.1% improvement over [9] and a 1.6% improvement over [10]. At last, for F (F-measure), the proposed algorithm achieves an average score of 72.3%, which is 2.1% higher than that of [10]. Moreover, its FPS is the best compared with other detection algorithms.

Similarly, Table 3 summarizes the results of text detection based on the CTW1500 data set for all detection algorithms. For the recall accuracy (R), the recall rate of the AEMF algorithm using the backbone network of ResNet50 is 80.3%, ranking as the first place among all detection

Table 1: Results on the ICDAR2015 dataset.

Method	*R*	*P*	*F*	FPS
Ours-VGG16	72.8	80.4	76.4	6.5
Ours-ResNet50	**77.3**	**84.2**	**79.3**	**16**
CTPN [24]	52	74	61	7.7
EAST [10]	73.5	83.6	78.2	13.2
Faster-RCNN [9]	52.4	73.2	61.8	—
SegLink [22]	76.8	73.1	75	—

Table 2: Results on the MSRA-TD500 dataset.

Method	*R*	*P*	*F*	FPS
Ours-VGG16	62.3	82.1	70.8	7.5
Ours-ResNet50	**65.3**	**83.3**	**72.3**	**8**
CTPN [24]	53.8	60.4	56.9	1.1
EAST [10]	61.6	81.7	70.2	6.5
Faster-RCNN [9]	62.3	82.1	70.8	7.5
SegLink [22]	—	—	—	—

Table 3: Results on the CTW1500 dataset.

Method	*R*	*P*	*F*	FPS
Ours-VGG16	61.4	48.5	61.2	17
Ours-ResNet50	**80.3**	50.4	**63.1**	**23**
CTPN [24]	60.4	**53.8**	56.9	7.14
EAST [10]	78.7	49.1	60.4	21.2
Faster-RCNN [9]	—	—	—	—
SegLink [22]	42.3	40	40.8	10.7

algorithms. Again, the FPS of the AEMF algorithm is better than most current participating detection algorithms. In particular, the proposed method is 23% higher than all participating algorithms.

Overall, the AEMF algorithm achieves the best performance in the ICDAR2015 data set based on the backbone network of ResNet50. Again, the proposed algorithm achieved the recall rate of 77.3%, the accuracy rate of 84.2%, the average score of 79.3%, and the frame rate per second of 16. For the rest two dataset, our method achieves the second best detection performance. However, our FPS result is always the highest, which indicates that the proposed algorithm is very efficient to process the upcoming video frames. The high processing rate compensates for our detection accuracy using the dataset of MSRA-TD500 and CTW1500. Overall, it is empirically confirmed that the proposed algorithm is able to detect text in varying directions from different environments. Another advantage is that our method is capable for long-text detection with different shapes.

5. Conclusion and Future Work

Due to the poor performance of text detection from the current industrial complex environment, this paper proposes an attention-based efficient and multifeature fast text detector (AEMF). This algorithm is based on the fusion of the region suggestion and semantic segmentation features [28]. In the region suggestion, the feature information is extracted and preserved through multifeature fusion. Meanwhile, the semantic segmentation is implemented using an attention mechanism, which helps in refining helpful information and suppressing useless contents. At last, the combination of region suggestion and semantic segmentation is aggregated to form the final detection. Experimental results show that the proposed algorithm improves the performance of text detection in multiple directions and shape recognition from complex environments; in addition, the proposed algorithm also improves the long-text detection accuracy simultaneously.

The future work mainly includes the following aspects: (1) the detection of long curved text can be considered by applying other data fusion strategy; (2) we could adopt a lighter model to improve the detection speed; and (3) we can also apply the proposed algorithm to other detection datasets for a large-scale experiment.

Data Availability

ICDAR2015 has a total of 1500 images, 1000 for training and the rest for testing. The dataset is a public dataset provided by the incidental scene text (INCENTAL SCENE TEXT) reading competition added to the RRC by ICDAR2015. The dataset is a random image of a street or mall taken unfocused by Google Glass, designed to help text detection and recognition models improve generalization performance. MSRA-TD500 is a dataset of 300 training images and 200 test images of text detection provided by Huazhong University of Science and Technology in 2012. The images in the dataset consist of pictures taken in offices, shopping malls, and streets, and the text in the pictures is composed of Chinese and English in different directions. CTW1500 is a textual image dataset of natural scenes in Chinese provided by Tsinghua University. The images include flat text, urban street scene text, township street scene text, text under low lighting conditions, distant text, and partial display text. The images are collected from Tencent Street View and are highly diverse.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61873004).

References

[1] C. Yao, X. Bai, W. Liu et al., "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, Providence, RI, USA, June 2012.

[2] C. Liu, C. Wang, and R. Dai, "Text detection in images based on unsupervised classification of edge-based features," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 610–614, Seoul, Korea (South), September 2005.

[3] M. Buta, L. Neumann, and J. Matas, "FASText: efficient unconstrained scene text detector," in *Proceedings of the IEEE*

International Conference on Computer Vision, pp. 1206–1214, Santiago, Chile, December 2015.

[4] P. Shivakumara, P. Trung Quy, and T. Chew Lim, "A laplacian approach to multi-oriented text detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 412–419, 2011.

[5] J. Wang, Z. Wang, and X. Tian, "Review of natural scene text detection and recognition based on deep learning," *Journal of Software*, vol. 31, no. 5, pp. 1465–1496, 2020.

[6] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou et al., "ICDAR 2015 competition on robust reading," in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, August 2015.

[7] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition CVPR*, Providence, RI, USA, July 2012.

[8] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognition*, vol. 90, pp. 337–345, 2019.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] X. Zhou, C. Yao, H. Wen, Y. Wang et al., "EAST: an effificient and accurate scene text detector," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[11] M. Liao, B. Shi, X. Bai et al., "TextBoxes: a fast text detector with a single deep neural network," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 4161–4167, Honolulu, HI, USA, July 2017.

[12] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision—ECCV 2016*, pp. 21–37, 2016.

[13] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2529–2541, 2016.

[14] J. Ma, W. Shao, H. Ye et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[15] S. Xie and Z. Tu, "Holistically-nested edge detection," *International Journal of Computer Vision*, vol. 125, no. 75, pp. 3–18, 2015.

[16] Z. Zhang and C. Zhang, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[17] D. Deng and H. Liu, "PixelLink: detecting scene text via instance segmentation," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 6773–6780, Vancouver, Canada, February 2018.

[18] C. Xue, S. Lu, and F. Zhan, "Accurate scene text detection through border semantics awareness and bootstrapping," in *Computer Vision—ECCV 2018*, pp. 370–387, Springer, Berlin, Germany, 2018.

[19] C. Xue and S. Lu, "MSR: multi-scale shape regression for scene text detection," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Yokohama, Japan, August 2019.

[20] Y. Xu and Y. Wang, "TextField: learning a deep direction field for irregular scene text detection," *IEEE Trans. on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2018.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Computer Vision—ECCV 2018*, pp. 3–19, Springer, Berlin, Germany, 2018.

[22] B. Shi, X. Bai, and SJ. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[23] P. He, W. Huang, T. He et al., "Single shot text detector with regional attention," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3047–3055, Venice, Italy, October 2017.

[24] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Computer Vision—ECCV 2016*, pp. 56–72, Springer, Berlin, Germany, 2016.

[25] J.-B. Liu, M. Arockiaraj, M. Arulperumjothi et al., "Distance based and bond additive topological indices of certain repurposed antiviral drug compounds tested for treating COVID-19," *International Journal of Quantum Chemistry*, vol. 121, no. 10, 2021.

[26] Z. Yang, J. Yang, B. Yecies, and W. Li, "Multi-view learning for context-aware extractive summarization," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 1762–1769, Canberra, Australia, December 2020.

[27] J. Yang and J. Ma, "Feed-forward neural network training using sparse representation," *Expert Systems with Applications*, vol. 116, pp. 255–264, 2019.

[28] J. Yang and J. Ma, "A structure optimization framework for feed-forward neural networks using sparse representation," *Knowledge-Based Systems*, vol. 109, pp. 61–70, 2016.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-sclae image recognition," 2014, https://arxiv.org/abs/1409.1556.

[30] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.