Reinforcement Shrink-Mask for Text Detection

Chuang Yang, Mulin Chen, Yuan Yuan, *Senior Member, IEEE*, and Qi Wang, *Senior Member, IEEE*

***Abstract*—Existing real-time text detectors reconstruct text contours by shrink-masks only. Though they simplify the framework and can make the model run fast, the strong dependence on shrink-masks leads to unreliable detection results (e.g., miss detection and overdetection). Moreover, these methods ignore the information from surrounding pixels, which causes sensitive shrink-masks and accelerates the reliability decline of detection results. Considering the above problems, we construct an effective and efficient text detection network, termed as Reinforcement Shrink-Mask for Text Detection (RSMTD), which strengthens the model's ability to recognize texts while enjoying a high detection speed. Specifically, an effective text representation strategy (Reinforcement Shrink-Mask, RSM) is designed to decouple texts and shrink-masks. RSM builds texts through shrink-masks and reinforcement offsets to ensure stable detection results encountering shrink-masks that deviate from the ground-truth. It is worth noting that reinforcement offsets can force our method to focus on the foreground shapes to bring precise shrink-mask edges. For the robustness improvement of shrink-masks, Super-pixel Window (SPW) is proposed to encourage RSMTD to utilize the surroundings of each pixel to predict shrink-masks. Particularly, SPW treats the interval regions between texts and shrink-masks as background, which helps to suppress interval regions and to avoid text adhesion. Moreover, a lightweight feature merging branch is constructed to further accelerate the inference process. As demonstrated in the experiments, our method is superior to existing state-of-the-art (SOTA) methods in both detection accuracy and speed on multiple benchmarks.**

***Index Terms*—Text detection, arbitrary-shaped text, real-time text detector.**

I. Introduction

TEXT detection [1]–[3], a key technology to provide text location information for many text recognition (STR) [4]–[6] related applications, has become a hot topic recently. Existing text detection methods can be roughly divided into non-real-time methods and real-time methods. The former [7]–[9] focuses on achieving high detection accuracy. However, the complex framework leads to low detection speed and high memory requirements, which makes it hard to deploy them in the device. Though the latter [10], [11] can run faster than the non-real-time methods benefiting from a lightweight network and simple post-processing, the limited detection accuracy restricts their applicability. Specifically, these lightweight methods are trained on fewer official labels only, which leads to the models being hard to learn effective shrink-mask features. It further results in the weak model's ability to distinguish different shrink-masks and to suppress interferences (the shrink-mask generated by baseline in Fig. 1). Meanwhile, they rebuild text contour by the shrink-mask only (SMO). Concretely, these methods extend the shrink-mask by a specific offset outward. Since the offset is computed by the area and perimeter of the shrink-mask, the rebuilt contour is related to the shrink-mask only, which leads to the strong dependence on the shrink-mask. It aggravates the drawback of the weak recognition for shrink-mask and results in unreliability rebuilt contours. Therefore, how to improve the recognition of shrink-mask features without bringing extra computational cost, and to avoid the deep dependency of final results on the predicted shrink-masks in the reconstructing process is still explored.

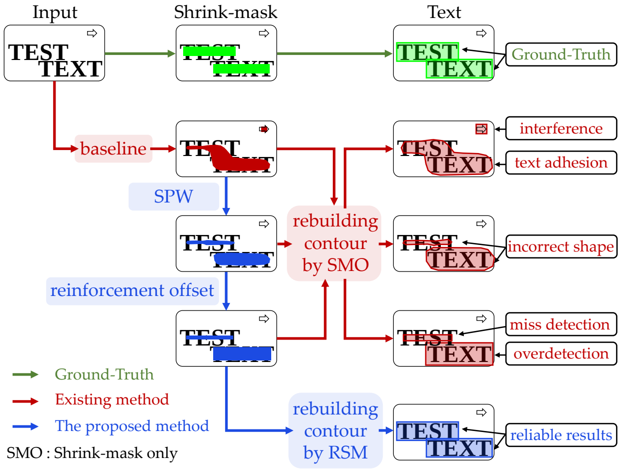


Fig. 1. Existing light-weight shrink-mask based text detectors are hard to predict shrink-mask accurately and detection results are sensitive to shrink-masks. The designed Super-pixel Window (SPW) helps to utilize surrounding pixels for enhancing the reliability of shrink-masks and for avoiding text adhesion. Meanwhile, reinforcement offsets force our method to focus on recognizing foreground shapes to improve the accuracy of shrink-mask edges. The above SPW and reinforcement offsets strengthen the shrink-mask recognition effectively. Furthermore, Reinforcement Shrink-Mask (RSM) rebuilds texts through the combination of shrink-masks and reinforcement offsets to bring the robustness improvement of detection results.

Considering the challenges above, we construct an effective and efficient text detection network based on Reinforcement Shrink-Mask (RSM) and Super-pixel Window (SPW), namely RSMTD, which can achieve comparable detection accuracy with non-real-time methods and detection speed with real-time methods. Specifically, the proposed RSM represents text instances by shrink-masks and reinforcement offsets to decouple shrink-masks and texts. Different from existing methods that compute the extending offsets according to the area and perimeter of the predicted shrink-mask directly, the proposed

This work was supported by the National Natural Science Foundation of China under Grant U21B2041, U1864204, 61825603, and 62106182.

Chuang Yang is with the School of Computer Science, and with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P. R. China. Mulin Chen, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China. E-mail: cyang113@mail.nwpu.edu.cn, chenmulin@mail.nwpu.edu.cn, y.yuan.ieee@gmail.com, crabwq@gmail.com.

Qi Wang is the corresponding author.

method models the computing offset as a regression task. It can ensure proper extending offset for the shrink-mask even though it deviates from the ground-truth. In the aspect of detection accuracy, it ensures the reliability of rebuilt text instances even when the predicted shrink-masks deviate from the corresponding ground-truth. Importantly, reinforcement offsets encourage our method to focus on the shrink-mask shapes in the training stage, which avoids bringing interference information into detection results. For detection speed, text contours can be rebuilt by extending the shrink-mask contours outward by reinforcement offsets directly, which simplifies the post-processing and makes it work that run faster, and deploy easier. SPW is designed to facilitate the pixelwise prediction tasks. It encourages RSMTD to utilize the surroundings of each pixel to help the shrink-masks recognition and the reinforcement offsets prediction. Meanwhile, SPW only treats shrink-masks as foreground, which helps to suppress interval regions between shrink-masks and texts and to present text adhesion. Particularly, SPW brings no computational cost to the inference process and can be integrated into other detectors seamlessly. Additionally, a lightweight feature merging branch is constructed to save computational cost to further speed up the inference process.

As depicted in Fig. 1, SPW helps RSMTD to separate adhesive shrink-masks and to suppress interference regions. Reinforcement offsets supervise our network to enhance the shapes accuracy of shrink-masks and texts effectively. Furthermore, reliable text contours can be obtained by the combination of revised shrink-masks and reinforcement offsets. Compared with baseline, the proposed RSMTD improves the robustness of detection results significantly while ensuring high detection speed. The experimental results demonstrate that our method is superior to existing state-of-the-art (SOTA) text detection methods (as shown in Fig. 2). The contributions of this work are summarized as follows:

1) Reinforcement Shrink-Mask is proposed to decompose texts into shrink-masks and reinforcement offsets, which ensures a simple framework and can rebuild text contours accurately even when the predicted shrink-masks deviate from the corresponding ground-truth. Particularly, reinforcement offsets help the proposed method to recognize shrink-mask edges precisely.
2) Super-pixel Window is designed to encourage the network to utilize the surroundings of each pixel for pixelwise prediction tasks, which facilitates the shrink-masks recognition, interval regions suppression, and reinforcement offsets prediction. Importantly, SPW brings no extra computational cost to the inference process.
3) An effective and efficient detector with a lightweight network and simple post-processing is proposed. It makes detection results more reliable, run faster, and deployment easier, which provides essential support for applications.

The rest of the paper is organized as follows. Related works about text detection are shown in Section II. The details of our method are presented in Section III. The ablation study and the comparison with state-of-the-art (SOTA) methods in Section IV demonstrate the superior performance of the proposed method. Meanwhile, some detection results are visualized and model efficiency is analyzed. Moreover, we have discusses the limitation of RSMTD in this section. The conclusion of this paper is given in Section V.

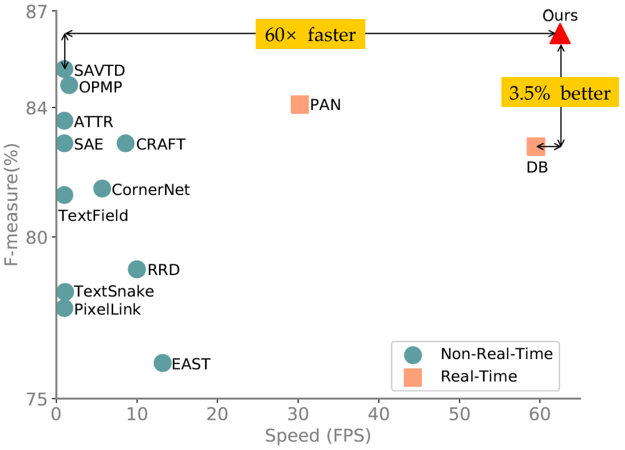


Fig. 2. Performance of detection accuracy and speed on MSRA-TD500 dataset. Our method outperforms the best non-real-time method SAVTD [8] in detection accuracy and runs faster than the best real-time method DB [12].

II. Related Work

In recent years, scene text detection technique is rapidly developing. Existing text detection methods can be roughly classified into non-real-time and real-time methods.

A. Non-Real-Time Text Detection Methods.

Non-real-time text detection methods are composed of complicated network and post-processing. They focus on achieving high detection accuracy. Liao *et al.* [13] proposed rotation-sensitive features for detecting oriented text instances. Zhou *et al.* [14] proposed dense prediction for abandoning the anchor mechanism. Law *et al.* [15] represented text instances by four corner regions and combined them to rebuild text regions. However, they failed to detect irregular-shaped text instances. Though Wang *et al.* [16] modeled text contours by multiple contour points, highly curved texts could not be fitted accurately. Some works [17]–[20] decomposed text instances into a number of character-level boxes and connected them to rebuild text contours. Zhu *et al.* [21] converted text instance contours from point sequences into Fourier signature vectors. Liu *et al.* [22] represented text contours by Bezier-Curve. Zhang *et al.* [23] detected text instances by an effective pyramid lengthwise and sidewise residual sequence model. Zhang *et al.* [24] and Wang *et al.* [25] predicted quadrilateral bounding boxes at first and then they segmented text regions in the range of the boxes. To improve the detection performance of very long sentences, Tian *et al.* [26] segmented an embedding map to connect multiple tiny text instances as integral ones. Xu *et al.* [27] predicted pixel classes and directions to detect texts from the background and distinguish different text instances, respectively. Although these methods can fit arbitrary-shaped text contours accurately, the low detection

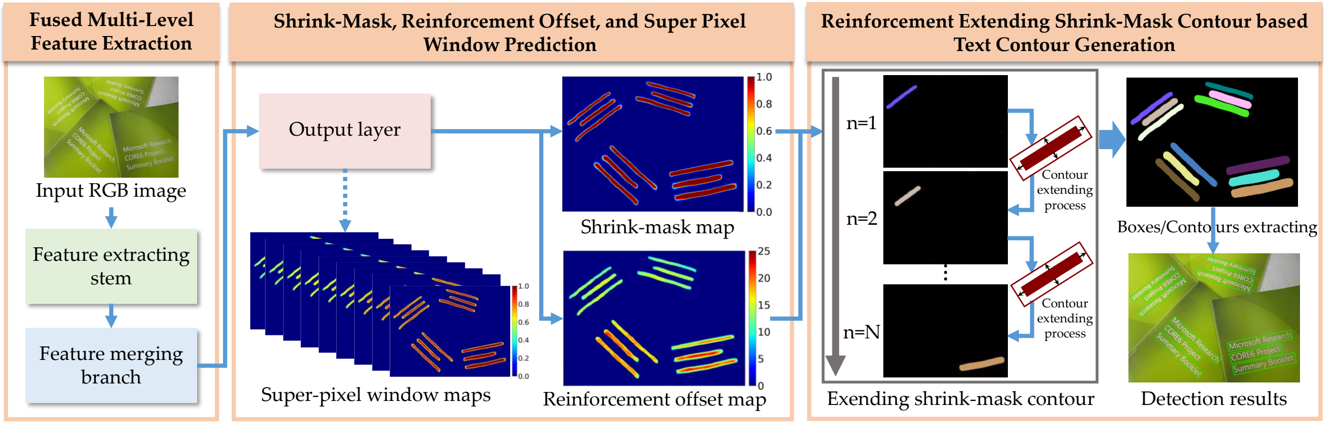

Fused Multi-Level Feature Extraction
Input RGB image
Feature extracting stem
Feature merging branch
Shrink-Mask, Reinforcement Offset, and Super Pixel Window Prediction
Output layer
Shrink-mask map
Super-pixel window maps
Reinforcement offset map
1.0
0.8
0.6
0.4
0.2
0.0
25
20
15
10
5
0
Reinforcement Extending Shrink-Mask Contour based Text Contour Generation
n=1
n=2
n=N
Contour extending process
Contour extending process
Boxes/Contours extracting
Exending shrink-mask contour
Detection results

Fig. 3. Overall architecture of the proposed RSMTD, which consists of feature extracting module (Fused Multi-Level Feature Extraction), prediction headers (Shrink-Mask, Reinforcement Offset, and Super Pixel Window Prediction), and post-processing (Reinforcement Extending Shrink-Mask Contour based Text Contour Generation). Dashed arrow is the training only operator. “n=1, n=2, ..., n=N” indicates the 1th, 2th, ..., Nth shrink-mask in the image, respectively.

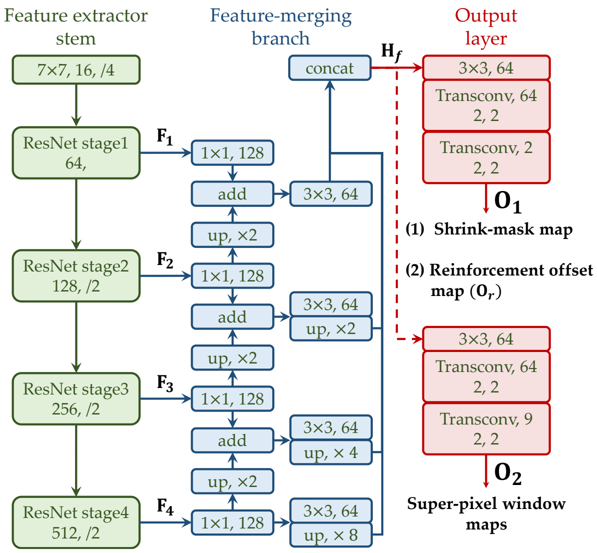


Fig. 4. Details of feature extracting network. Red dashed arrow is the training only operator. ‘/2’ indicates downsample operator. $\mathbf{O_1}$ is responsible for predicting shrink-mask and reinforcement offset maps. $\mathbf{O_2}$ used for generating Super-pixel window maps.

speed and high memory requirement hinder providing support for STR-related [28]–[30] applications.

B. Real-Time Text Detection Methods.

To obtain fast detection speed and reduce memory requirements, many real-time text detection methods are proposed recently. These methods are equipped with a lightweight backbone and adopt the shrink-mask based segmentation framework to detect text instances. For example, Wang *et al.* [31] and Liao *et al.* [12] rebuilt text contours by extending shrink-mask regions. Specifically, Wang *et al.* [31] segmented shrink-masks and texts simultaneously. In the inference stage, the authors extended shrink-mask regions to text regions through pixel-level post-processing, which was time-consuming. Since Liao *et al.* [12] could compute the distances between shrink-masks and text masks according to shrink-masks through the algorithm proposed in [32], they only needed to segment shrink-masks and to extend them through patch-level post-processing, which further improved the detection speed. However, though these existing real-time text detection methods enjoy fast detection speed, the detection accuracy is still far behind non-real-time methods.

III. Our Method

In this section, the overall architecture of RSMTD is introduced at first. Then, the Reinforcement Shrink-Mask (RSM) and Super-pixel Window (SPW) are described in detail, respectively. Next, the label generation process is illustrated. In the end, the optimization function is elaborated.

A. Overall Architecture

The overall architecture of the proposed RSMTD is shown in Fig. 3, which consists of feature extracting stem, feature merging branch, output layer, and reinforcement extending strategy based post-processing. In the Inference stage, a fused multi-level feature map is extracted through feature extracting stem and feature merging branch at first. Then, the output layer conducts on the fused feature map to predict shrink-mask, reinforcement offset, and SPW. The SPW is a training only operator. It helps to utilize the information of surrounding pixels to facilitate the pixelwise prediction tasks and brings no extra computational cost to the inference process. For shrink-mask and reinforcement offset, the pixel values of them denote the probability that whether the pixel belongs to text and the minimum distance between the pixel and text contour, respectively. In the end, to reconstruct all text contours in the image, all shrink-mask contours are extended by the predicted reinforcement offsets through post-processing one by one. Benefiting from the advantages of reinforcement offset, RSMTD can rebuild text contours even when the predicted shrink-mask deviates from the corresponding ground-truth, which brings significant improvement in detection accuracy.

The details of feature extracting stem, feature merging

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3209022

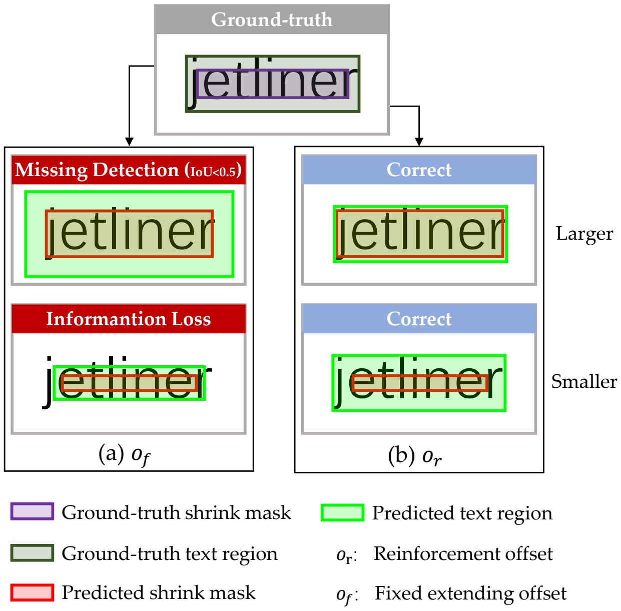


Fig. 5. Illustration of essential differences between traditional fixed extending strategy and reinforcement extending strategy. The fixed extending offset o_f is computed by the area and perimeter of the shrink-mask, which leads to miss detection or information loss when the predicted shrink-mask deviates from the ground-truth. For the proposed reinforcement offset o_r, which can effectively avoid the above problems.

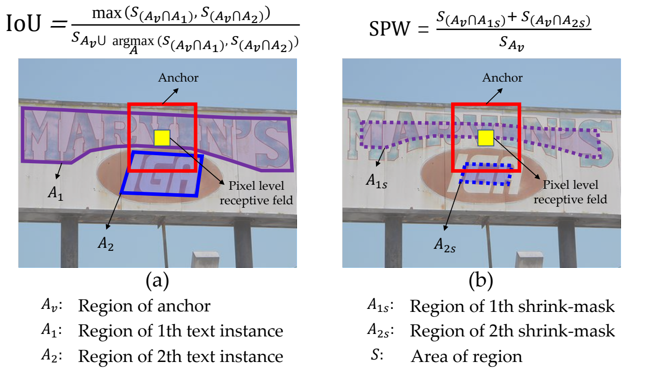

$\text{IoU} = \frac{\max(S_{(A_v \cap A_1)}, S_{(A_v \cap A_2)})}{S_{A_v} \cup \operatorname{argmax}_A (S_{(A_v \cap A_1)}, S_{(A_v \cap A_2)})}$
$\text{SPW} = \frac{S_{(A_v \cap A_{1s})} + S_{(A_v \cap A_{2s})}}{S_{A_v}}$
Anchor
Pixel level receptive feld
A_1
A_2
A_{1s}
A_{2s}
(a)
(b)
A_v: Region of anchor
A_1: Region of 1th text instance
A_2: Region of 2th text instance
A_{1s}: Region of 1th shrink-mask
A_{2s}: Region of 2th shrink-mask
S: Area of region

Fig. 6. Illustration of essential differences between Intersection of Union (IoU) and Super-pixel Window (SPW).

branch, and output layer are illustrated in Fig. 4. ResNet [33] is adopted as feature extracting stem directly. It generates multi-level feature maps whose sizes are $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ and $\frac{1}{32}$ of the input image, respectively. To reduce the computational cost to speed up the inference process, we follow the idea of Feature Pyramid Network (FPN) [34] and design a lightweight feature merging branch. It is used for concatenating the multi-level feature maps to generate a fused feature map, which contains high-level semantic information and low-level apparent feature simultaneously. Specifically, in the feature merging branch, we gradually merge multi-level feature maps as follows:

$$\mathbf{H_i^1} = \text{conv}_{1\times1,128}\left(\mathbf{F_i}\right), \ i = 1,2,3,4, \tag{1}$$

$$\mathbf{H_i^2} = \begin{cases} \mathbf{H_i^1}, & i = 1 \\ \text{up}_{\times2}\left(\mathbf{H_i^1}\right) + \mathbf{H_{i-1}^1}, & i = 2,3,4 \end{cases}, \tag{2}$$

$$\mathbf{H_i^3} = \begin{cases} \text{conv}_{3\times3,64}\left(\mathbf{H_i^2}\right), & i = 1 \\ \text{up}_{\times2^{(i-1)}}\left(\text{conv}_{3\times3,64}\left(\mathbf{H_i^2}\right)\right), & i = 2,3,4 \end{cases}, \tag{3}$$

$$\mathbf{H_f} = \text{concatenate}\left(\mathbf{H_i^3}\right), \ i = 1,2,3,4, \tag{4}$$

where $\mathbf{F_i}$ is the feature map from feature extracting stem. $\mathbf{H_i^1}$, $\mathbf{H_i^2}$, and $\mathbf{H_i^3}$ are the hidden feature maps. $\mathbf{H_f}$ is the fused multi-level feature map outputted from feature merging branch. $\text{up}_{\times2^{(i-1)}}(\cdot)$ denotes upsampling the corresponding feature map to $2^{(i-1)}$ times of original size .

For output layer, which consists of two similar convolutional neural network structures and is responsible for predicting the shrink-mask, reinforcement offset, and SPW. Specifically, the output layer takes $\mathbf{H_f}$ as input and generetes $\mathbf{O_1}$ and $\mathbf{O_2}$. The $\mathbf{O_1}$ is a tensor with the size of $\frac{H}{4} \times \frac{W}{4} \times 2$, It consists of predicted shrink-mask map and reinforcement offset map ($\mathbf{O_r}$), where both map sizes are $\frac{H}{4} \times \frac{W}{4} \times 1$. The $\mathbf{O_r}$ is a tensor with the size of $\frac{H}{4} \times \frac{W}{4} \times 9$, which is composed of 9 SPW maps.

B. Reinforcement Shrink-Mask

To avoid missing detection and improve the information integrity of detected text instances, we propose a Reinforcement Shrink-Mask (RSM) to represent text instances. The RSM-based model can rebuild text contours by extending shrink-mask contours outward by reinforcement offsets accurately, even when the predicted shrink-mask deviates from the corresponding ground-truth. Particularly, since the prediction of reinforcement offset forces our method to recognize text shapes and shrink-masks enjoy the same shapes as texts, it facilitates RSMTD to generate accurate shrink-mask shapes to avoid bringing interference information into detection results.

As we can see from Fig. 5 (a), existing real-time methods (such as [12]) rebuild text contour by extending the predicted shrink-mask contour outward by a fixed offset o_f:

$$o_f = \frac{S_s}{L_s}\delta_t, \tag{5}$$

where S_s and L_s are the area and perimeter of the predicted shrink-mask, respectively. δ_t is the extending coefficient.

As shown in Eq. (5), Since o_f deeply depends on the area and perimeter of the predicted shrink-mask, the o_f will deviate from the ground-truth a large when the predicted shrink-mask is larger or smaller than the ground-truth, which further leads to missing detection or text information loss (Fig. 5 (a)) and influences model detection accuracy. Considering these problems, we propose to extend the shrink-mask contour by a reinforcement offset o_r :

$$\mathbf{O_r} = \text{Tconv}_{2\times2,1}\left(\text{Tconv}_{2\times2,64}\left(\text{conv}_{3\times3,64}\left(\mathbf{H_f}\right)\right)\right), \tag{6}$$

$$o_r = \frac{\mathbf{O_r^{i,j}} + \mathbf{O_r^{k,v}}}{2}, \tag{7}$$

where $\text{Tconv}\left(\text{Tconv}\left(\text{conv}\left(\cdot\right)\right)\right)$ denotes the output layer in Fig. 4. $\mathbf{O_r}$ is the predicted reinforcement offset map. For

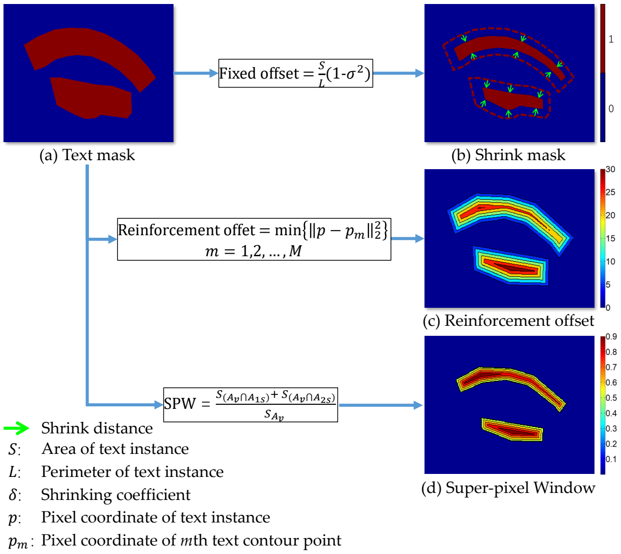

(a) Text mask
Fixed offset = $\frac{S}{L}(1-\sigma^2)$
(b) Shrink mask
1
0
Reinforcement offet = $\min\{\|p-p_m\|_2^2\}$
$m = 1,2,\ldots,M$
30
25
20
15
10
5
0
(c) Reinforcement offset
SPW = $\frac{S_{(A_v\cap A_{1s})}+S_{(A_v\cap A_{2s})}}{S_{A_v}}$
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
(d) Super-pixel Window
Shrink distance
S: Area of text instance
L: Perimeter of text instance
δ: Shrinking coefficient
p: Pixel coordinate of text instance
p_m: Pixel coordinate of mth text contour point

Fig. 7. Illustration of the label generation process.

```
Algorithm 1: Label Generation
Data: text mask $B_t$, shrinking coefficient $\sigma$, Area of text
      instance $S$, perimeter of text instance $L$, width and
      height of input image $W$ and $H$, coordinates of
      current pixel $(i, j)$, anchor region $A_v$, shrink-mask
      region $A_s$
Result: binary mask map of shrink-mask $B_s$, heat map
        of reinforcement offset $B_r$, heat map of
        Super-pixel Window $B_w$
initializing $B_s \in \mathbb{R}^{W,H}$, $B_r \in \mathbb{R}^{W,H}$, and $B_s \in \mathbb{R}^{W,H,9}$;
for k$th$ text instance $in$ $B_t$ do
    fixed offset$_k$ $\leftarrow \frac{S_k}{L_k}(1-\delta^2)$;
    shrink-mask$_k$ $\leftarrow$ shrinking text contour inward by
      fixed offset$_k$;
    drawing shrink-mask$_k$ on $B_s$;
    for v$th$ pixel$_v^{i,j}$ $in$ text instance$_k$ do
        min $\leftarrow$ 1e6;
        for m$th$ pixel$_m$ $in$ text contour$_k$ do
            if $\|\text{pixel}_v^{i,j} - \text{pixel}_m\|_2^2 < \min$ then
                min $\leftarrow \|\text{pixel}_v^{i,j} - \text{pixel}_m\|_2^2$;
            end
        end
        $B_r^{i,j} \leftarrow$ min;
    end
end
for k$th$ shrink-mask $in$ $B_t$ do
    for l$th$ pixel$_l^{i,j}$ $in$ shrink-mask$_k$ do
        for n$th$ anchor $on$ pixel$_l^{i,j}$ do
            SPW$_n$ $\leftarrow \frac{S_{(A_v^n\cap A_{1s})}+S_{(A_v^n\cap A_{2s})}}{S_{A_v^n}}$;
            $B_w^{i,j,n} \leftarrow$ SPW$_n$;
        end
    end
end
```

a specific text instance, i, j and k, v are the coordinates of the two closest points to the text contour, respectively. Different from o_f, o_r is independent of the predicted shrink-mask, which means the distance between the predicted shrink-mask contour and text contour can be evaluated accurately, even when the predicted shrink-mask deviates from the corresponding ground-truth. Benefiting from the advantages of o_r, our method can avoid missing detection and improve text information integrity when rebuilding text contour based on incorrect shrink-mask (Fig. 5 (b)), which brings significant improvement for detection accuracy. Moreover, since the text contours can be reconstructed by extending the shrink-mask contour outward directly, the RSMTD enjoys a simpler network and post-processing than non-real-time methods (such as [9], [21]), which facilitates our model run times faster.

C. Super-pixel Window

Since some background regions enjoy similar low-level features (such as color, texture, and gradients) with text instances, existing real-time methods are hard to distinguish the shrink-mask from them according to the pixel-level features (the yellow regions in Fig. 6). Therefore, we design Super-pixel Window (SPW) to encourage RSMTD to utilize the surrounding information of each pixel for enhancing pixelwise prediction tasks.

The existing way to encourage CNN model to utilize the surroundings of each pixel is to optimize the network under the supervision of the intersection of union (IoU). As shown in Fig. 6, the IoU is defined as:

$$\text{IoU} = \frac{\max\left(S_{(A_v\cap A_1)}, S_{(A_v\cap A_2)}\right)}{S_{A_v\cup \underset{A}{\text{argmax}}\left(S_{(A_v\cap A_1)}, S_{(A_v\cap A_2)}\right)}}, \tag{8}$$

where A_v indicates the region of pre-defined anchor. We set the aspect ratios of anchors to $\frac{1}{2}$, 1, 2 and the scales of them to 2, 4, 8. A_1 and A_2 are the regions of text instances. $\cap$ and $\cup$ are the intersection and union operators. $S_{(\cdot)}$ is the area of region. $\underset{A}{argmax}(\cdot)$ is the text region that enjoys maximum intersection with A_v.

However, since IoU considers the whole text region, it brings interference to the training process when meeting some very long text instances (such as A_1 in Fig. 6 (a)) that are far beyond the network receptive field. Moreover, smaller text instances are treated as background for IoU, which further leads to semantic ambiguity between text and background. Considering these problems, we design SPW as follows:

$$\text{SPW} = \frac{S_{(A_v\cap A_{1s})} + S_{(A_v\cap A_{2s})}}{S_{A_v}}, \tag{9}$$

where A_{1s} and A_{2s} are shrink-mask regions.

Different from IoU, as shown in Fig. 6 (b), the proposed SPW enjoys the following advantages: (1) only the shrink-masks within the range of anchor are recognized, which effectively avoids the interference brought by very long text instances. At the same time, SPW treats the interval region between shrink-mask contour and text contour as background, which helps to distinguish shrink-masks and predict reinforcement offsets; (2) all shrink-masks within the range of anchor are considered no matter their scales are large or small, which avoids the semantic ambiguity brought by treating the small ones as background. Additionally, the SPW brings no extra computational cost because it can be removed from the inference process, which ensures a high detection speed.

D. Label Generation

The label generation processes for shrink-mask map, reinforcement offset map, and SPW map are visualized in Fig. 7.

For shrink-mask (as shown in Fig. 7 (b)), the corresponding contour is generated through moving the text instance contour inward by shrink offset o_s that is computed by the Vatti clipping algorithm [32]:

$$o_s = \frac{S_t}{L_t}\left(1 - \delta_s^2\right), \tag{10}$$

where S_t and L_t are the area and perimeter of text instance, respectively. δ_s is the shrinking coefficient, which is set to 0.4 empirically.

For reinforcement offset o_r (as shown in Fig. 7 (c)), which is defined as the minimum distance between the pixels on text instance and text contour:

$$\begin{aligned} o_r = \min\left\{\|p - p_m\|_2^2\right\}, \\ m = 1, 2, ..., M, \end{aligned} \tag{11}$$

where p and p_m are the coordinates of pixels of text instance and text contour, respectively. $\min(\cdot)$ indicates the minimum operator. M is the number of contour points.

For SPW (as shown in Fig. 7 (d)), it treats shrink-masks as valid regions and the corresponding pixel values of the map are computed by the Eq. (9).

Giving a text mask B_t, the corresponding binary map label of shrink-mask, heat map labels of reinforcement offset and SPW can be generated by Algorithm 1.

E. Optimization

The proposed RSMTD can be regarded as a multi-task network aiming at shrink-mask segmentation, reinforcement offset prediction, and SPW prediction. For shrink-mask segmentation, the classic dice loss [35] is used for evaluating the difference between the segmentation binary masks and the corresponding labels, which is formulated as:

$$\mathcal{L}_{sm} = 1 - \frac{2 \times |Y \cap \widehat{Y}| + 1}{|Y| + |\widehat{Y}| + 1}, \tag{12}$$

where Y and $\widehat{Y}$ are the ground-truth and predicted shrink-masks. Since the serious imbalance of positive (text instances) and negative samples (background) and plenty of simple samples, Online Hard Example Mining (OHEM) [36] is adopted when calculating $\mathcal{L}_{sm}$.

TABLE I
INFORMATION OF THE TRAINING SAMPLES ON MULTIPLE BENCHMARKS. $\mathrm{N_i}$, $\mathrm{N_{sm}}$, $\mathrm{N_{spw}}$, AND $\mathrm{N_r}$ ARE SAMPLE NUMBERS OF INSTANCE, SHRINK-MASK, SPW, AND REINFORCEMENT OFFSET.

Dataset	Type	$\mathrm{N_i}$	$\mathrm{N_{sm}}$	$\mathrm{N_{spw}}$	$\mathrm{N_r}$
SynthText	Train	7266866	$\mathrm{N_i}$	90×$\mathrm{N_i}$	30×$\mathrm{N_i}$
MSRA-TD500	Train	1973	$\mathrm{N_i}$	90×$\mathrm{N_i}$	30×$\mathrm{N_i}$
	Test	582	–		
Total-Text	Train	9290	$\mathrm{N_i}$	90×$\mathrm{N_i}$	30×$\mathrm{N_i}$
	Test	2215	–		
CTW1500	Train	7692	$\mathrm{N_i}$	90×$\mathrm{N_i}$	30×$\mathrm{N_i}$
	Test	3068	–		

For reinforcement offset and SPW prediction, the ratio loss $\mathcal{L}_{ratio}$ is adopted to compute their gradients:

$$\mathcal{L}_{ratio}(P,\ \widehat{P}) = \log\frac{\max\left(P,\ \widehat{P}\right)}{\min\left(P,\ \widehat{P}\right)}, \tag{13}$$

where P and $\widehat{P}$ are the ground-truth and prediction, respectively. Therefore, $\mathcal{L}_{ratio}$ based reinforcement offset loss $\mathcal{L}_{o_r}$ and SPW loss $\mathcal{L}_{SPW}$ can be defined as:

$$\mathcal{L}_{o_r} = \mathcal{L}_{ratio}(o_r,\ \widehat{o_r}), \tag{14}$$

$$\mathcal{L}_{SPW} = \mathcal{L}_{ratio}(SPW,\ \widehat{SPW}), \tag{15}$$

where o_r and $\widehat{o_r}$ are the ground-truth (computed by the Eq. (7)) and predicted o_r. SPW and $\widehat{SPW}$ are the ground-truth (computed by the Eq. (9)) and predicted SPW.

The final loss function $\mathcal{L}$ used for training the proposed network is given by:

$$\mathcal{L} = \lambda_1\mathcal{L}_{sm} + \lambda_2\mathcal{L}_{o_r} + \lambda_3\mathcal{L}_{SPW}, \tag{16}$$

where λ is hyper-parameter and used to balance multiple losses weights. In this paper, λ_1, λ_2, and λ_3 are set to 1, 0.25, and 0.25 respectively.

IV. EXPERIMENTS

To demonstrate the effectiveness of the RSM and SPW, we conduct ablation studies on the MSRA-TD500 and Total-Text datasets. Moreover, we also compare the proposed RSMTD with related state-of-the-art (SOTA) methods on multiple public datasets to show the superiority in both detection accuracy and speed.

A. Datasets

SynthText [37] is composed of 800k synthetic images generated by combining varied text instances with 8k natural images. This dataset is used for pre-training the proposed RSMTD.

MSRA-TD500 [38] consists of arbitrary-oriented and long text sentences. It includes 300 training images and 200 test images. Since the training images are rather less, we include 400 images from HUST-TR400 [39] as training data.

Total-Text [40] contains various shapes word-level text instances (such as horizontal, multi-oriented, and curved shapes) simultaneously. It is composed of 1255 training images and 300 testing images.

CTW1500 [41] is a dataset for testing the model performance on line-level curved text instances, which has 1000 training images and 500 testing images.

TABLE II
DETECTION RESULTS WITH DIFFERENT SETTINGS ON MSRA-TD500 AND TOTAL-TEXT DATASETS. "S: 736" AND "S: 640" MEAN THAT THE SHORTER SIDE OF EACH TESTING IMAGE IS RESIZED TO BE 736 AND 640 PIXELS. "BASELINE" MEANS THE FRAMEWORK IS EQUIPPED WITH A TRADITIONAL SHRINK-MASK ONLY. "RSM" INDICATES REINFORCEMENT SHRINK-MASK. "SPW" MEANS SUPER-PIXEL WINDOW.

#	Model	RSM	SPW	MSRA-TD500 (S : 736)				Total-Text (S : 640)			
				P(%)	R(%)	F(%)	FPS	P(%)	R(%)	F(%)	FPS
1	baseline			87.2	81.6	84.3	64.2	87.1	80.9	83.9	73.2
2	baseline+	✓		88.9	82.5	85.6	62.5	87.9	82.7	85.2	70.9
3	baseline+	✓	✓	89.8	83.1	86.3	62.5	88.5	83.8	86.1	70.9

TABLE III
DETECTION RESULTS WITH DIFFERENT SETTINGS ON TOTAL-TEXT THAT EQUIPPED WITH SPW. "RSM" INDIATES REINFORCEMENT SHRINK-MASK. "P_{50}", "R_{50}", AND "F_{50}" INDICATE THE PRECISION, RECALL, AND F-MEASURE THAT IOU IS SET AS 50%. "P_{75}", "R_{75}", AND "F_{75}" ARE THE PRECISION, RECALL, AND F-MEASURE THAT IOU IS SET AS 75%.

	TotalText					
	P_{50}(%)	R_{50}(%)	F_{50}(%)	P_{75}(%)	R_{75}(%)	F_{75}(%)
w/o RSM	87.6	82.3	84.9	86.6	82.3	84.4
with RSM	88.5	83.8	86.1	87.3	83.8	85.5

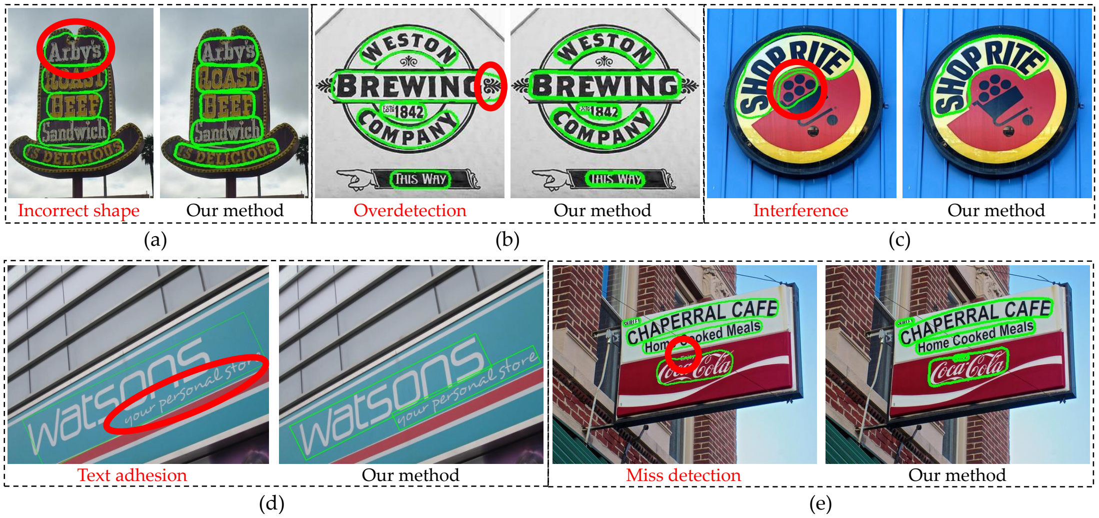

Incorrect shape Our method
(a)
Overdetection Our method
(b)
Interference Our method
(c)
Text adhesion Our method
(d)
Miss detection Our method
(e)

Fig. 8. Qualitative comparisons with traditional shrink-masks based rebuilt text contours and reinforcement shrink-masks based rebuilt text contours.

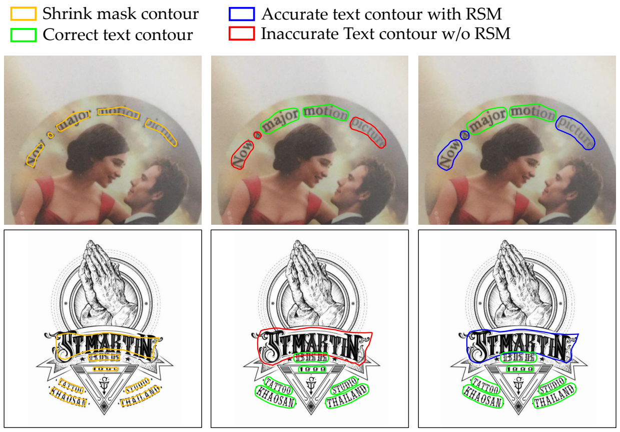
B. Implementation Details

The feature extracting stem is pre-trained on ImageNet [42] and the whole network is pre-trined on SynthText [37]. In the fine-tuning stage, Adam [43] is employed to train the model. The initial learning rate is set to 0.001 and is adjusted by the 'poly' strategy used in [44]. The training batch size is set to 16. We initialize the weights of feature merging branch and output layers with the strategy in [45]. For all datasets, the blurred text regions labeled as DO NOT CARE are ignored during the training. We visualize the details of training and testing samples of SynthText, MSRA-TD500, Total-Text, and CTW1500 in Table. I for understanding the optimization and evaluation process, where N_{i}, N_{sm}, $\mathrm{N}_{\mathrm{spw}}$, and N_{r} are sample numbers of instance, shrink-mask, SPW, and reinforcement offset respectively. All the experiments are conducted on Pytorch using a workstation with two 1080Ti GPUs. To increase the training data and avoid over-fitting, we adopt the following data augmentation strategies: (1) random horizontal flipping, (2) random scaling and cropping, (3) random rotation with an angle range of (-10, 10).

C. Ablation Study

The ablation study is conducted to show the effectiveness of the proposed Reinforcement Shrink-Mask (RSM) and Super-pixel Window (SPW).

Reinforcement Shrink-Mask (RSM). As mentioned above, RSM can bring significant improvement to the reliability of detection results by decoupling texts into shrink-masks and reinforcement offsets. In this section, ablation experiments on MSRA-TD500 and Total-Text are conducted to demonstrate the superiority of the proposed RSM. We can see from Table. II #2, benefitting from the advantage that RSM can rebuild text contours even when predicted shrink-masks deviate from the corresponding ground-truth, our method brings 1.3% improvement in terms of F-measure on MSRA-TD500 and Total-Text simultaneously compared with baseline. Some examples are shown in Fig. 9, it is found that traditional shrink-masks based rebuilt text contours (the red colored geometries in


Shrink mask contour
Correct text contour
Accurate text contour with RSM
Inaccurate Text contour w/o RSM

(a) Shrink mask (b) Text contours w/o RSM (c) Text contours with RSM

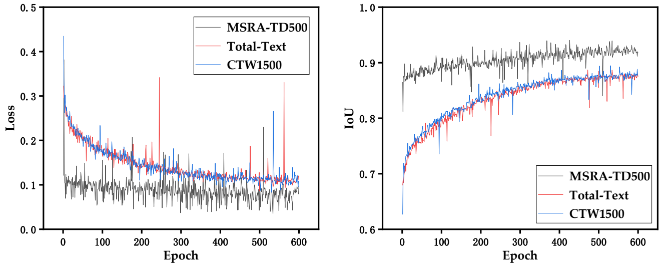
Fig. 9. Comparison with traditional shrink-mask based rebuilt text contour and reinforcement shrink-mask based rebuilt text contour.

Fig. 9) are smaller than ground-truth, which leads to text information loss. Reinforcement shrink-masks based rebuilt text contours (the blue colored geometries in Fig. 9) avoid these problems effectively. It can be found from the second row in Fig. 9 that reinforcement shrink-masks based rebuilt text contours are reliable even when the shrink-masks are larger than ground-truth. To further verify the effectiveness of RSM, we evaluate our method that equipped with SPW on Total-Text benchmark with different settings. Specifically, as shown in Table. III, RSM achieves 1.2% and 1.1% gains in F-measure when the corresponding IoU is set as 50% and 75%, respectively. Particularly, for the RSM based model, the corresponding F-measure with 75% IoU achieves 85.5%, which outperforms the F-measure of baseline with 50% IoU by 0.6%. This experimental result verifies that RSM can improve the integrity of rebuilt text contours encountering inaccurate shrink-masks. As depicted in Fig. 8, compared with baseline, the proposed RSMTD can effectively correct the shrink-mask shapes (Fig. 8 (a)) while avoiding miss detection (Fig. 8 (e)) and overdetection (Fig. 8 (b)) with the help of RSM. These experiments demonstrate the superiority of the proposed RSM for detecting text instances.

Super-pixel Window (SPW). For enhancing pixelwise prediction tasks (the recognition of shrink-masks and the prediction of reinforcement offsets) and suppressing interval regions between shrink-masks and texts, SPW is designed in this work. Since SPW can encourage the proposed method to utilize the surrounding information of each pixel, which facilitates to suppress interference regions (Fig. 8 (c)) to improve the robustness of the predicted shrink-masks. It makes our method achieve significant performance gain in terms of F-measure. As shown in Table. II #3, SPW brings 0.7% improvement in F-measure compared with 'baseline+RSM' (Table. II #2). Meanwhile, as the definition of SPW (Eq. 9), it treats the interval regions between shrink-masks and texts as background, which helps to avoid text adhesion effectively. It can be found in Fig. 8 (d), RSMTD with SPW successfully separates those adhesive texts. Moreover, since SPW does not participate in the reconstructing process of text contours, it brings no extra complexity to the framework and computational cost to the inference process (as shown in Table. II #2–#3), which facilitates the deployment of the text detection technique. The experimental results prove that SPW not only can improve the detection accuracy but also can ensure a high detection speed.

TABLE IV
COMPARISON WITH RELATED METHODS ON MSRA-TD500. "NRT" AND "RT" INDICATE NON-REAL-TIME AND REAL-TIME TEXT DETECTION METHODS, RESPECTIVELY. "**BLUE**" AND "**RED**" ARE THE BEST RESULTS OF NON-REAL-TIME AND REAL-TIME TEXT DETECTION METHODS, RESPECTIVELY.

Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [14]	87.3	67.4	76.1	**13.2**
	PixelLink [20]	83.0	73.2	77.8	-
	TextSnake [18]	83.2	73.9	78.3	1.1
	RRD [13]	87.0	73.0	79.0	10
	TextField [27]	87.4	75.9	81.3	-
	CornerNet [15]	87.6	76.2	81.5	5.7
	CRAFT [19]	88.2	78.2	82.9	8.6
	SAE [26]	84.2	81.7	82.9	-
	MaskTextSpotter-v3 [46]	90.7	77.5	83.5	-
	STD-SL [7]	87.7	80.8	84.1	6.4
	OPMP [23]	86.0	83.4	84.7	1.6
	SAVTD [8]	89.2	81.5	**85.2**	-
RT	DB [12]	90.4	76.3	82.8	62.0
	DB++ [11]	87.9	82.5	85.1	55.0
	PAN [31]	84.4	83.8	84.1	30.2
	PAN++ [10]	85.3	84.0	84.7	32.5
	Ours	89.8	83.1	**86.3**	**62.5**



(a) Curves of training loss (b) Curves of training IoU

Fig. 10. Convergence analysis of our method on MSRA-TD500, Total-Text, and CTW1500 datasets. IoU means the Intersection of Union between predicted shrink-mask and the corresponding ground-truth.

D. Comparison with State-of-the-Art Methods

Existing state-of-the-art (SOTA) text detection methods can be divided into non-real-time (NRT) and real-time (RT) methods. The former and latter focus on detection accuracy and speed, respectively. In this section, the proposed RSMTD is compared with them on MSRA-TD500, Total-Text, and CTW1500 datasets to show the comprehensive superiority in both detection accuracy and speed.

MSRA-TD500: Long Straight Text Benchmark. We evaluate RSMTD for detecting multi-language long straight text instances in the MSRA-TD500 dataset. The shorter side of

TABLE V
Comparison with related methods on Total-Text. "NRT" and "RT" indicate Non-Real-Time and Real-Time text detection methods, respectiely. "**Blue**" and "**Red**" are the best results of Non-Real-Time and Real-Time text detection methods, respectively.

Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [14]	50.0	36.2	42.0	-
	TextSnake [18]	82.7	74.5	78.4	-
	TextField [27]	81.2	79.9	80.6	-
	TextRay [16]	83.5	77.9	80.6	-
	OPMP [23]	88.5	82.9	85.6	1.4
	LOMO [24]	87.6	79.3	83.3	-
	FCENet [21]	87.4	79.8	83.4	-
	CRAFT [19]	87.6	79.9	83.6	-
	ReLaText [17]	84.8	83.1	84.0	-
	ContourNet [25]	86.9	83.9	85.4	3.8
	DRRG [9]	86.5	84.9	85.7	-
	STD-SL [7]	89.7	83.5	**86.5**	**6.1**
RT	DB [12]	88.3	77.9	82.8	50.0
	DB++ [11]	87.4	79.6	83.3	48.0
	PAN [31]	89.3	81.0	85.0	39.6
	PAN++ [10]	89.9	81.0	85.3	38.3
	Ours	88.5	83.8	**86.1**	**70.9**

TABLE VI
Comparison with related methods on CTW1500. "NRT" and "RT" indicate Non-Real-Time and Real-Time text detection methods, respectively. "**Blue**" and "**Red**" are the best results of Non-Real-Time and Real-Time text detection methods, respectively.

Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [14]	78.7	49.1	60.4	**21.2**
	TextSnake [18]	67.9	85.3	75.6	1.1
	TextField [27]	83.0	79.8	81.4	-
	TextRay [16]	82.8	80.4	81.6	-
	OPMP [23]	85.1	80.8	82.9	1.4
	LOMO [24]	85.7	76.5	80.8	-
	FCENet [21]	85.7	80.7	83.1	-
	CRAFT [19]	86.0	81.1	83.5	-
	ReLaText [17]	86.2	83.3	84.8	10.6
	ContourNet [25]	83.7	84.1	83.9	4.5
	DRRG [9]	85.9	83.0	84.5	-
	STD-SL [7]	87.2	85.0	**86.1**	5.3
RT	DB [12]	84.8	77.5	81.0	55.0
	DB++ [11]	86.7	81.3	83.9	40.0
	PAN [31]	86.4	81.2	83.7	39.8
	PAN++ [10]	87.1	81.1	**84.0**	36.0
	Ours	87.8	80.3	83.9	**72.1**

each testing image is resized to 736 and the detection results are evaluated by the evaluation metrics in [41].

The results on MSRA-TD500 are shown in Table. IV. It is found that RSMTD achieves the F-measure of 86.3% at an astonishing detection speed (62.5 FPS). For real-time text detection methods, benefiting from the advantage of RSM that text contours can be rebuilt accurately even when the predicted shrink-masks deviate from the corresponding ground-truth, our method outperforms DB [12] by 3.5% about F-measure. Moreover, the lightweight feature merging branch facilitates RSMTD to run 2x times faster than PAN [31]. Compared with OPMP [23] and SAVTD [8], since the proposed RSMTD is equipped with a simpler framework and is encouraged to enhance pixelwise prediction tasks by SPW, our method surpasses them by 1.6% and 1.1% in F-measure respectively and runs 40x times faster than them at least. Moreover, we visualize the training process in Fig. 10, it is found that the model enjoy fast converge speed on the MSRA-TD500 dataset.

The detection results demonstrate the superiority of our method for detecting long straight text instances. We also illustrate some qualitative results in Fig. 11 (a).

Total-Text: Word-Level Curved Text Benchmark. We test the proposed RSMTD for detecting word-level curved text instances in the Total-Text dataset. To ensure a fair comparison environment, the shorter side of each testing image is resized to 640 in this experiment.

Since existing real-time methods simplify the CNN model to pursue high detection speed, the lightweight network leads to sensitive detection results (e.g., DB [12] and PAN [31]). To improve the robustness of rebuilt text contours and bring no extra computational cost to the inference process, we design SPW to encourage our method to utilize the surrounding information of each pixel. As we can see from Table. V, SPW helps RSMTD achieve 86.1% in F-measure while enjoying 70.9 FPS in detection speed, which outperforms previous real-time methods and demonstrates the effectiveness of the introduced SPW. For non-real-time methods, they enhance the ability to recognize text instances by the complicated CNN model. However, some methods ignore the text adhesion problem, which makes it difficult to separate adhesive text instances and causes bad performance on Total-Text benchmark. Benefiting from the strong ability of SPW to suppress interval regions, RSMTD can avoid the text adhesion problem effectively (as shown in Fig. 8 (d)), which helps our method surpass previous SOTA non-real-time methods in detection accuracy. Specifically, RSMTD outperforms ContourNet [25] and DRRG [9] by 0.7% and 0.4% in F-measure. Moreover, RSM based text representation method saves much computational cost, which makes our method can run 10x times faster than the fastest non-real-time method (STD-SL [7]).

The results demonstrate the effectiveness of RSMTD for detecting word-level curved text instances. Moreover, as shown in Fig. 11 (b), though many text instances are close to each other, our method still can distinguish them effectively because of the superiority of RSM based text representation method.

CTW1500: Line-Level Curved Text Benchmark. The proposed RSMTD is evaluated on CTW1500 dataset to show the model's robustness for detecting long curved text instances. The shorter sides of images in this benchmark are set to 640.

RSM is proposed to represent texts by shrink-masks and reinforcement offsets in this work, which helps to improve the robustness of detection results and to simplify the framework to speed up the inference process. Moreover, the lightweight feature merging branch further accelerates the detection speed while keeping comparable feature fusion ability with the traditional feature merging module. As shown in Table. VI,

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3209022

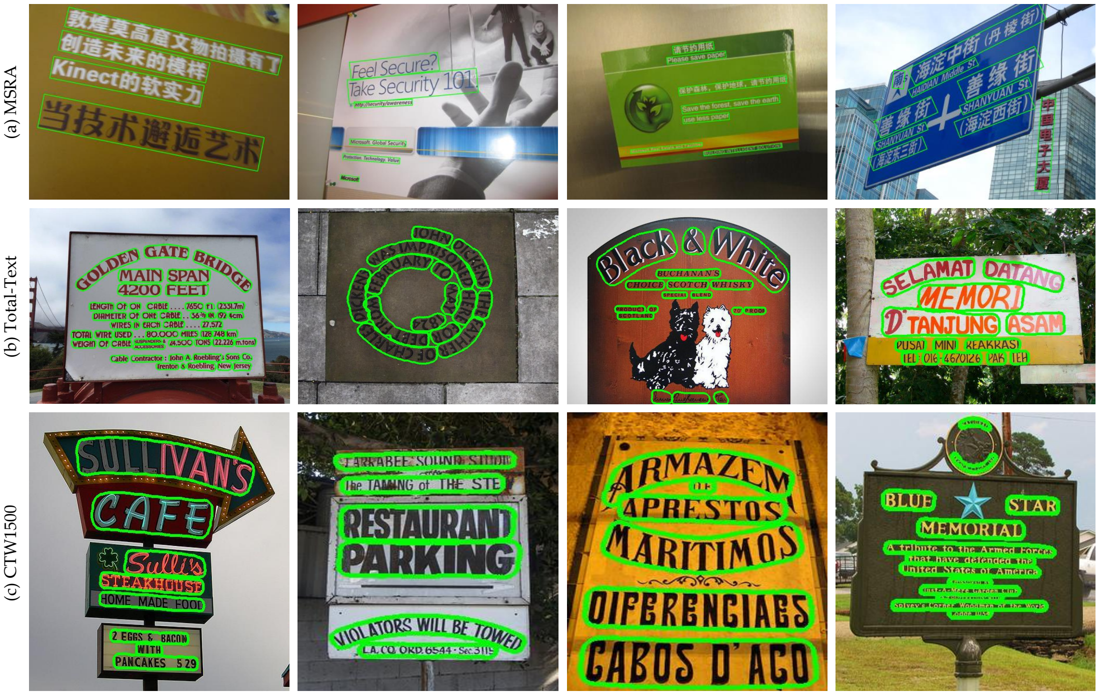


Fig. 11. Illustration of some qualitative detection results of the proposed RSMTD.

TABLE VII
CROSS-DATASET EVALUATIONS ON LINE-LEVEL DATASETS.

Evaluation dataset	Training on MSRA-TD500		
CTW1500	P (%)	R (%)	F (%)
	82.7	74.3	78.3
Evaluation dataset	Training on CTW1500		
MSRA-TD500	P (%)	R (%)	F (%)
	82.5	77.8	80.1

benefiting from the advantages of RSM and lightweight feature merging branch, our method outperforms DB [12] 2.9% in terms of F-measure and runs 32.3 FPS faster than PAN [31]. The superior comprehensive performance on multiple public benchmarks verifies that the proposed RSMTD is the best real-time text detection framework so far. For non-real-time methods, SPW and RSM help our method surpasses most of them in detection accuracy. Although RSMTD is a little lower (0.6% and 0.9%) than DRRG [9] and ReLaText [17] in F-measure, our method runs 7x times faster speed than it.

The evaluation results on the CTW1500 show the model's robustness for detecting long curved text instances. Moreover, some qualitative detection results on CTW1500 are illustrated. As we can see from the second column in Fig. 11 (c) even though there are some interference regions that enjoy similar low-level features (such as color and texture), our method still can distinguish them from texts effectively.

E. Cross Dataset Text Detection

The above experiments demonstrate the superior comprehensive performance of our method on multiple public benchmarks. In this section, we further verify the generalization ability of the proposed RSMTD by cross evaluation experiments. Considering both MSRA-TD500 and CTW1500 are line-level benchmarks, we design two groups experiments on them. Specifically, we train the proposed RSMTD on MSRA-TD500 and test it on CTW1500 at first. Then, the proposed RSMTD is trained on CTW1500 and evaluated on MSRA-TD500. As shown in Table. VII, benefiting from the proposed text representation method (RSM), RSMTD enjoys superior ability to fit irregular-shaped text instances. Although CTW1500 includes many curved text instances, the model trained on straight texts still can detect them with comparable performance. Specifically, our method achieves 78.3% in F-measure on CTW1500, which outperforms the TextSnake [18] (in Table. VI) by 2.7%. Moreover, SPW encourages the proposed method to utilize the surrounding information of each pixel to facilitate the pixelwise prediction tasks, which helps our approach performs better for detecting long texts. As we can see from Table. VII, RSMTD trained on CTW1500 surpasses EAST [14], PixelLink [20], and TextSnake [18] (in Table. IV) by 1.8% F-measure at least on MSRA-TD500. The experimental results verify the proposed RSMTD enjoys strong generalization and robustness for diverse shaped text instances in different benchmarks.

F. Speed Analysis

Benefiting from the advantages of the proposed ASM, SPW, and lightweight feature merging branch, our method not only runs faster than existing real-time methods but also achieves a comparable detection accuracy with non-real-time methods. In this section, to explore the superiority in terms of detection speed and computational resources of the inference process, we show the time consumption of each module of the proposed

TABLE VIII
TIME CONSUMPTION OF RSMTD ON THREE PUBLIC BENCHMARKS. THE TOTAL TIME CONSISTS OF BACKBONE, HEAD AND POST-PROCESSING. "S" MEANS THAT THE SHORTER SIDE OF EACH TESTING IMAGE. "HEAD" CONTAINS THE FEATURE MERGING BRANCH AND PREDICTION HEADERS . "POST" REPRESENTS POST-PROCESSING.

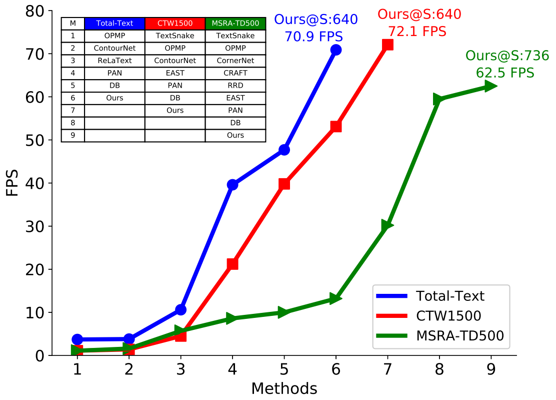
Dataset	S	Time consumption (ms)			F(%)	FPS
		Backbone	Head	Post		
MSRA-TD500	736	8.1	5.8	2.1	86.3	62.5
Total-Text	640	7.1	5.1	1.9	86.1	70.9
CTW1500	640	7.0	5.0	1.9	83.9	72.1

TABLE IX
COMPARISON OF COMPUTATIONAL COST AND PERFORMANCE OF DIFFERENT REAL-TIME DETECTORS. "GFLOPS" INDICATES FLOATING POINT OF OPERATIONS.

Methods	GFLOPs	F-measure (%)		
		MSRA-TD500	Total-Text	CTW1500
PAN [31]	63.88	84.1	85.0	83.7
DB [12]	52.54	82.8	82.8	81.0
Ours	46.96	86.3	86.1	83.9

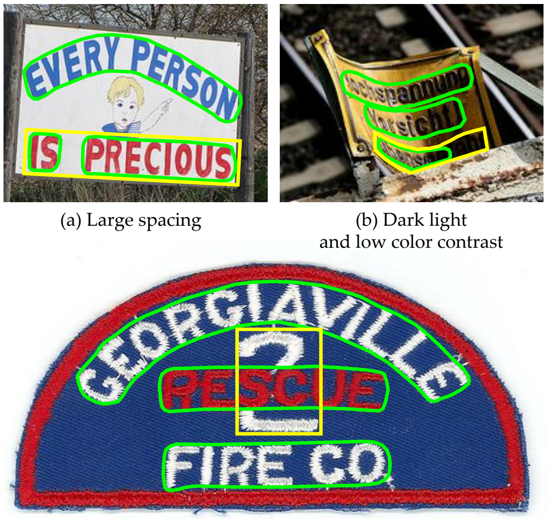
detection framework and compare the computational cost with other SOTA methods.

RSM based text representation strategy makes it that text contours can be rebuilt by extending shrink-mask contours outward by reinforcement offsets directly, which simplifies the post-processing and saves much computational cost. As illustrated in Table. VIII, the post-processing of our method only takes about 14% of the total time consumption. At the same time, since SPW does not participate in the inference process and the designed lightweight feature merging branch has fewer parameters compared with the previous feature merging module, the head only takes about 36% of the total time consumption (as shown in Fig. 4). The experimental results demonstrate the effectiveness of RSM, SPW, and lightweight feature merging branch for improving the model detection speed. Additionally, we verify the superiority of RSMTD in detection speed. As shown in Fig. 12, our method outperforms other related SOTA methods a lot on multiple public text detection datasets. Moreover, we show the computational cost of existing real-time methods in Table. IX. Because RSMTD only consists of two prediction headers and is equipped with a lightweight feature merging branch, our method enjoys the least floating point of operations (FLOPs) and the highest detection accuracy, which either demonstrates the efficiency of the designed detector.



M	Total-Text	CTW1500	MSRA-TD500
1	OPMP	TextSnake	TextSnake
2	ContourNet	OPMP	OPMP
3	ReLaText	ContourNet	CornerNet
4	PAN	EAST	CRAFT
5	DB	PAN	RRD
6	Ours	DB	EAST
7		Ours	PAN
8			DB
9			Ours

Fig. 12. Comparison of detection speed on multiple benchmarks. "S" is the shorter side of each testing image for our method.



(a) Large spacing

(b) Dark light and low color contrast

(c) Text occlusion

Fig. 13. Illustration of some challenging samples. The green bounding boxes are the detection results from our method. The yellow ones are labels.

G. Limitations of Our Algorithm

The experiments before have demonstrated the effectiveness of the proposed RSM and SPW, and the superior comprehensive performance of RSMTD on multiple public benchmarks. They have analyzed the details of the framework's efficiency. Although our method discusses some problems (e.g., the strong dependence on shrink-masks and the ignorance of information from surrounding pixels) existing in previous works and introduces RSM and SPW to help RSMTD work well in most cases of detecting arbitrary-shape texts, it fails for some challenging samples. In this section, we show some difficult text instances to further explore the limitation of our algorithm. As depicted in Fig. 13, there are three typical failure cases: (1) large spacing between different words of long text instance (in Fig. 13 (a)); (2) dark light and low color contrast (in Fig. 13 (b)); (3) text occlusion (in Fig. 13 (c)). For case (1), the proposed method lacks the ability to recognize the affinities between different words, which leads to failed detection of them. Though some works (such as [26]) have considered the problem of large spacing, they bring much computational cost and complicated post-processing. Meanwhile, for case (2), the weak representative features make it difficult to distinguish between the text and the background. Furthermore, for case (3), both the segmentation-based methods and regression-based method are hard to represent them, and there is a lack of enough research so far. Therefore, how to find effective and efficient solutions to these problems will be our future work.

V. CONCLUSION

In this paper, we propose an effective and efficient framework to detect arbitrary-shaped texts. We firstly propose Reinforcement Shrink-Mask (RSM) to fit text instances by shrink-masks and reinforcement offsets to decouple texts and shrink-masks, which improves the reliability of rebuilt text contours effectively. Then, we introduce the Super-pixel Window (SP-W) to enhance the pixelwise prediction tasks, which brings significant improvements to the accuracy of the predicted shrink-masks. Importantly, SPW can be removed from the inference process and brings no extra computational cost. In the end, a lightweight feature merging branch is constructed to fuse multi-level feature maps, which further facilitates the detection speed. Extensive experiments demonstrate the effectiveness of ASM and SPW. Comparison experiments show that the proposed RSMTD has superior comprehensive performance compared with existing SOTA methods.

REFERENCES

[1] M. Xue, P. Shivakumara, C. Zhang, Y. Xiao, T. Lu, U. Pal, D. Lopresti, and Z. Yang, "Arbitrarily-oriented text detection in low light natural scene images," *IEEE Transactions on Multimedia*, vol. 23, pp. 2706–2720, 2020.

[2] Y. Wang, H. Xie, Z. Zha, Y. Tian, Z. Fu, and Y. Zhang, "R-net: A relationship network for efficient and accurate scene text detection," *IEEE Transactions on Multimedia*, vol. 23, pp. 1316–1329, 2020.

[3] P. Dai, Y. Li, H. Zhang, J. Li, and X. Cao, "Accurate scene text detection via scale-aware data augmentation and shape similarity constraint," *IEEE Transactions on Multimedia*, 2021.

[4] L. Wu, Y. Xu, J. Hou, C. P. Chen, and C.-L. Liu, "A two-level rectification attention network for scene text recognition," *IEEE Transactions on Multimedia*, 2022.

[5] D. Peng, L. Jin, W. Ma, C. Xie, H. Zhang, S. Zhu, and J. Li, "Recognition of handwritten chinese text by segmentation: A segment-annotation-free approach," *IEEE Transactions on Multimedia*, 2022.

[6] M. Li, B. Fu, Z. Zhang, and Y. Qiao, "Character-aware sampling and rectification for scene text recognition," *IEEE Transactions on Multimedia*, 2021.

[7] W. Zhang, Y. Qiu, M. Liao, R. Zhang, X. Wei, and X. Bai, "Scene text detection with scribble line," in *ICDAR*. Springer, 2021, pp. 79–94.

[8] W. Feng, F. Yin, X. Zhang, and C. Liu, "Semantic-aware video text detection," in *CVPR*, 2021, pp. 1695–1705.

[9] S. Zhang, X. Zhu, J. Hou, C. Liu, C. Yang, H. Wang, and X. Yin, "Deep relational reasoning graph network for arbitrary shape text detection," in *CVPR*, 2020, pp. 9696–9705.

[10] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Y. Zhibo, T. Lu, and C. Shen, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[11] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[12] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization." in *AAAI*, 2020, pp. 11 474–11 481.

[13] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *CVPR*, 2018, pp. 5909–5918.

[14] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *CVPR*, 2017, pp. 5551–5560.

[15] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018, pp. 734–750.

[16] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *ACMMM*, 2020, pp. 111–119.

[17] C. Ma, L. Sun, Z. Zhong, and Q. Huo, "Relatext: exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks," *Pattern Recognition*, vol. 111, p. 107684, 2021.

[18] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *ECCV*, 2018, pp. 20–36.

[19] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *CVPR*, 2019, pp. 9365–9374.

[20] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *AAAI*, 2018, pp. 6773–6780.

[21] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *CVPR*, 2021, pp. 3123–3131.

[22] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "Abcnet: Real-time scene text spotting with adaptive bezier-curve network," in *CVPR*, 2020, pp. 9809–9818.

[23] S. Zhang, Y. Liu, L. Jin, Z. Wei, and C. Shen, "Opmp: An omnidirectional pyramid mask proposal network for arbitrary-shape scene text detection," *IEEE Transactions on Multimedia*, vol. 23, pp. 454–467, 2020.

[24] 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 *CVPR*, 2019, pp. 10 552–10 561.

[25] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *CVPR*, 2020, pp. 11 753–11 762.

[26] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *CVPR*, 2019, pp. 4234–4243.

[27] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: Learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.

[28] K. Karthick, K. Ravindrakumar, R. Francis, and S. Ilankannan, "Steps involved in text recognition and recent research in ocr; a study," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 2277–3878, 2019.

[29] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *ICCV*, 2019, pp. 4715–4723.

[30] R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A scalable handwritten text recognition system," in *ICDAR*, 2019, pp. 17–24.

[31] 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 *ICCV*, 2019, pp. 8440–8449.

[32] R. Vatti, "A generic solution to polygon clipping," *Communications of the ACM*, vol. 35, no. 7, pp. 56–63, 1992.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[35] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV*, 2016, pp. 565–571.

[36] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016, pp. 761–769.

[37] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *CVPR*, 2016, pp. 2315–2324.

[38] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *CVPR*, 2012, pp. 1083–1090.

[39] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[40] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *ICDAR*, vol. 1, 2017, pp. 935–942.

[41] Y. Liu, L. Jin, S. Zhang, and S. Zhang, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.

[42] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[46] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *ECCV*. Springer, 2020, pp. 706–722.

Chuang Yang received the B.E. degree in automation and the M.E. degree in control engineering from Civil Aviation University of China, Tianjin, China, in 2017 and 2020 respectively. He is currently working toward the Ph.D. degree in the School of Computer Science and School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and machine learning.

Mulin Chen received the B.E. degree in software engineering and the Ph.D. degree in computer application technology from Northwestern Polytechnical University, Xi'an, China, in 2014 and 2019 respectively. He is currently a researcher with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, X-i'an, China. His current research interests include computer vision and machine learning.

Yuan Yuan (M'05-SM'09) is currently a Full Professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. She has authored or co-authored over 150 papers, including about 100 in reputable journals, such as the IEEE TRANSACTIONS AND PATTERN RECOGNITION, as well as the conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.

Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, pattern recognition and remote sensing.