Adaptive Shrink-Mask for Text Detection

Chuang Yang, Mulin Chen, Yuan Yuan, Qi Wang*, Xuelong Li
The School of Computer Science, and the School of Artificial Intelligence,
Optics and Electronics (iOPEN),
Northwestern Polytechnical University, Xian 710072, Shaanxi, P. R. China.

Abstract

Existing real-time text detectors reconstruct text contours by shrink-masks directly, which simplifies the framework and can make the model run fast. However, the strong dependence on predicted shrink-masks leads to unstable detection results. Moreover, the discrimination of shrink-masks is a pixelwise prediction task. Supervising the network by shrink-masks only will lose much semantic context, which leads to the false detection of shrink-masks. To address these problems, we construct an efficient text detection network, Adaptive Shrink-Mask for Text Detection (ASMTD), which improves the accuracy during training and reduces the complexity of the inference process. At first, the Adaptive Shrink-Mask (ASM) is proposed to represent texts by shrink-masks and independent adaptive offsets. It weakens the coupling of texts to shrink-masks, which improves the robustness of detection results. Then, the Super-pixel Window (SPW) is designed to supervise the network. It utilizes the surroundings of each pixel to improve the reliability of predicted shrink-masks and does not appear during testing. In the end, a lightweight feature merging branch is constructed to reduce the computational cost. 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.

1. Introduction

Text detection, a key technology to provide text location information for many scene text recognition (STR) [3, 18, 24] 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, 36, 39] focuses on achieving high detection accuracy. However, since the complex framework leads to low detection speed and high memory requirement, non-real-time methods are hard to deploy in the device. Though the latter [15, 29] can run fast with a lightweight network and simple post-processing, the limited detection accuracy restricts their applicability. Therefore, how to design an efficient framework to detect arbitrary-shaped text instances with high detection accuracy and speed is still explored.

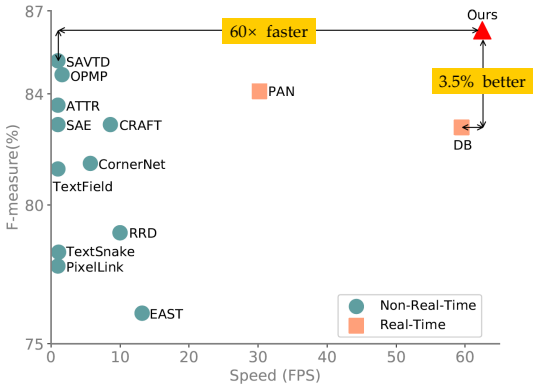


Figure 1. Performance of detection accuracy and speed on MSRA-TD500 dataset. Our method outperforms the best non-real-time method SAVTD [7] in detection accuracy and runs faster than the best real-time method DB [15].

Considering the aforementioned problems, we construct a novel real-time text detection network based on adaptive shrink-mask (ASM) and super-pixel window (SPW), namely ASMTD, which can achieve comparable detection accuracy with non-real-time methods and detection speed with real-time methods. Specifically, the proposed ASM represents text instances by shrink-masks and adaptive offsets. In the inference stage, text contours can be rebuilt by extending the shrink-mask contours outward by adaptive offsets directly, which simplifies the post-processing and makes it work that run faster and deploy easier. At the same time, adaptive offsets can evaluate the distance between shrink-mask contours and text contours accurately even when the predicted shrink-masks deviate from the corresponding ground-truth, which brings significant improvement for the accuracy of rebuilt text contours. Moreover, SPW is proposed to improve the accuracy of pixelwise prediction tasks. It encourages the network to provide rich semantic context for the discrimination of shrink-masks and the prediction of adaptive offsets. Importantly, the SPW can

be removed from the inference process, which does not introduce any temporal cost. Additionally, a lightweight feature merging branch is constructed to save computational cost to further speed up the inference process. The experiment results demonstrate our method is superior to existing state-of-the-art (SOTA) text detection methods (as shown in Fig. 1). The contributions of this work are summarized as follows:

1. An Adaptive Shrink-Mask is proposed to rebuild text contour by the shrink-mask and adaptive offset, which ensures a simple framework and can reconstruct text contour accurately even when the predicted shrink-mask deviates from the corresponding ground-truth.

2. Super-pixel Window is proposed to encourage the network to provide rich semantic context for pixelwise prediction tasks, which helps to recognize the shrink-mask and predict the adaptive offset and brings no extra computational cost to the inference process.

3. An 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.

2. Related Work

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

2.1. Non-Real-Time Text Detection Methods.

Non-real-time text detection methods are composed of complicated network and post-processing and focus on achieving high detection accuracy. Liao *et al*. [16] proposed rotation-sensitive features for detecting oriented text instances. Zhou *et al*. [38] proposed dense prediction for abandoning the anchor mechanism. Law and Deng [14] modeled 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*. [28] represented text contours by multiple contour points, highly curved text contours could not be fitted accurately. Some works [2, 21, 22] decomposed text instances into a number of character-level boxes and connected them to rebuild text contours. Zhu *et al*. [39] converted text instance contours from point sequences into Fourier signature vectors. Liu *et al*. [19] represented text contours by Bezier-Curve. Zhang *et al*. [35] detected text instances by an effective pyramid lengthwise and sidewise residual sequence model. Zhang *et al*. [34] and Wang *et al*. [30] predicted quadrilateral bounding boxes at first and then 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*. [31] predicted pixel classes and directions to detect text from the background and distinguish different text instances, respectively. Although these methods can fit arbitrary-shaped text contours accurately, the low detection speed and high memory requirement hinder providing support for STR-related [1, 11, 12] applications.

2.2. 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*. [29] and Liao *et al*. [15] rebuilt text contours by extending shrink-mask regions. Specifically, Wang *et al*. [29] segmented shrink-masks and text masks simultaneously. In the inference stage, the authors extended shrink-masks to text masks through pixel-level post-processing, which was time-consuming. Since Liao *et al*. [15] could compute the distances between shrink-masks and text masks according to shrink-masks through the algorithm proposed in [27], this method only needed to segment shrink-masks and 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.

3. Methodology

In this section, the overall architecture of ASMTD is introduced firstly. Then, the adaptive shrink-mask (ASM) 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.

3.1. Overall Architecture

The overall architecture of ASMTD is shown in Fig. 2, which consists of feature extracting stem, feature merging branch, output layer, and adaptive 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 the shrink-mask, adaptive offset, and SPW. The SPW is a training only operator, which helps to provide rich context information for the prediction tasks and brings no extra computational cost to the inference process. For shrink-mask and adaptive offset, the pixel value denotes the probability of whether it belongs to the text instance 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 adaptive offsets through

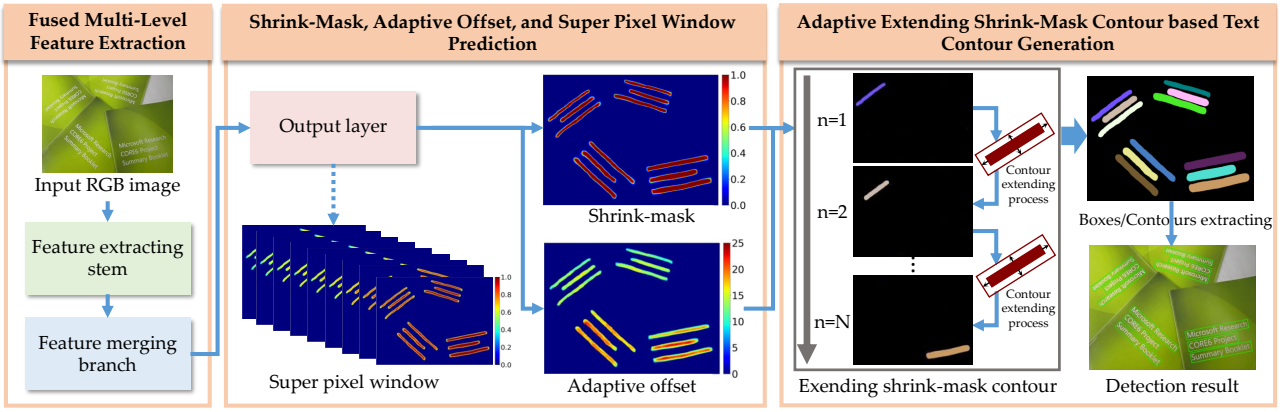


Figure 2. Overall architecture of ASMTD. Dashed arrow is the training only operator. "n=1, n=2, ..., n=N" indicates the 1th, 2th, ..., Nth shrink-mask in the image, respectively.

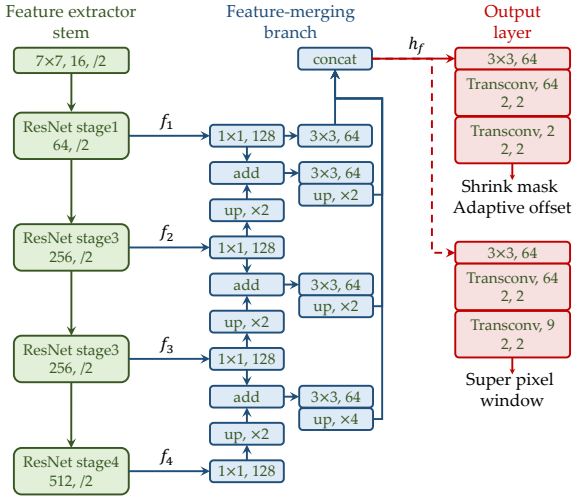


Figure 3. Details of feature extracting network. Red dashed arrow is the training only operator.

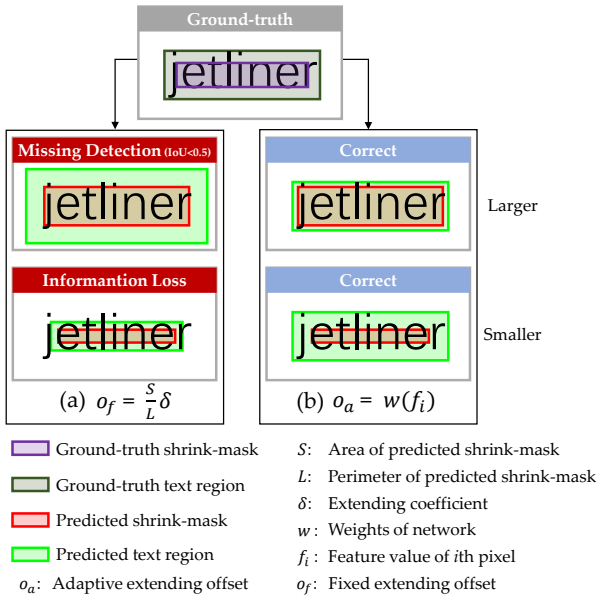


Figure 4. Illustration of essential differences between traditional fixed extending strategy and adaptive extending strategy.

post-processing one by one. Benefiting from the advantages of adaptive offset, our method can rebuild text contours even when the predicted shrink-mask deviates from the ground-truth, which brings significant improvement for detection accuracy.

The details of feature extracting stem, feature merging branch, and output layer are illustrated in Fig. 3. ResNet [10] 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) [17] and construct a lightweight feature merging branch, which is used for concatenating the multi-level feature maps to generate a fused feature map. The output layer is responsible for predicting the shrink-mask, adaptive offset, and SPW.

3.2. Adaptive Shrink-Mask

To avoid missing detection and improve the information integrity of detected text instances, we propose an adaptive shrink-mask (ASM) to represent text instances. ASM based model can rebuild text contours by extending shrink-mask contours outward by adaptive offsets accurately, even when the predicted shrink-mask deviates from the ground-truth.

As we can see from Fig. 4 (a), existing real-time methods (such as [15]) rebuild text contour through extending the predicted shrink-mask contour outward by a fixed offset o_f:

$$o_f = \frac{S_s}{L_s}\delta_t, \tag{1}$$

where S_s and L_s are the area and perimeter of the predicted

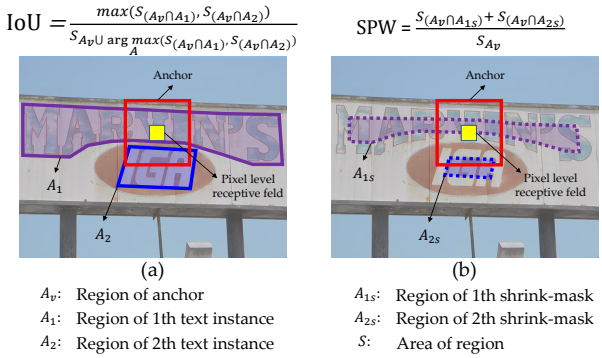


Figure 5. Illustration of essential differences between Intersection of Union (IoU) and Super-pixel Window (SPW).

shrink-mask, respectively. δ_t is the extending coefficient.

As shown in Eq. (1), 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. 4 (a)) and influences model detection accuracy. Considering these problems, we propose to extend the shrink-mask contour by an adaptive offset o_a :

$$o_a = w\left(f_i\right), \tag{2}$$

where $w(\cdot)$ denotes the weights of the output layer in Fig. 3. f_i means the ith pixel value of the fused feature map (h_f in Fig. 3). Different from o_f, o_a is independent of the predicted shrink-mask, which means the distance between 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_a, our method can avoid missing detection and improve text information integrity when rebuilding text contour based on incorrect shrink-mask (Fig. 4 (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 ASMTD enjoys a simpler network and post-processing than non-real-time methods (such as [36, 39]), which facilitates our model run times faster.

3.3. 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. 5). To provide rich semantic context for the pixelwise prediction tasks, we propose the Super-pixel Window (SPW) to train our model.

The existing way to obtain rich semantic context is to optimize the network under the supervision of the intersection

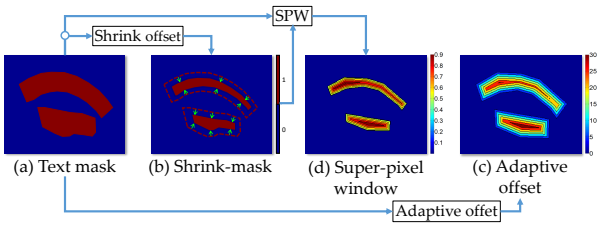


Figure 6. Illustration of the label generation process.

of union (IoU). As shown in Fig. 5, the IoU is defined as:

$$\text{IoU} = \frac{max\left(S_{(A_v \cap A_1)}, S_{(A_v \cap A_2)}\right)}{S_{A_v \cup \underset{A}{argmax}\left(S_{(A_v \cap A_1)}, S_{(A_v \cap A_2)}\right)}}, \tag{3}$$

where A_v indicates the anchor region. 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 for the training process when meeting some very long text instances (such as A_1 in Fig. 5 (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 propose SPW:

$$\text{SPW} = \frac{S_{(A_v \cap A_{1s})} + S_{(A_v \cap A_{2s})}}{S_{A_v}}, \tag{4}$$

where A_{1s} and A_{2s} are the regions of shrink-masks.

Different from IoU, as shown in Fig. 5 (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 adaptive 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 high detection speed.

3.4. Label Generation

The label generation processes for shrink-mask map, adaptive offset map, and SPW map are visualized in Fig. 6.

For shrink-mask (as shown in Fig. 6 (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 [27]:

$$o_s = \frac{S_t}{L_t}\left(1 - \delta_s^2\right), \tag{5}$$

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 adaptive offset o_a (as shown in Fig. 6 (c)), which is defined as the minimum distance between the pixel in the range of text instance and text contour:

$$o_a = min\left\{\|p - p_m\|_2^2\right\},\ m = 1, 2, ..., M, \tag{6}$$

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. 6 (d)), it treats shrink-masks as valid regions and the corresponding pixel values of the map are computed by the Eq. (4).

3.5. Optimization

The proposed ASMTD can be regarded as a multi-task network aiming at shrink-mask segmentation, adaptive offset prediction, and SPW prediction. For shrink-mask segmentation, the classic dice loss [23] is used for evaluating the difference between the segmentation binary masks and the corresponding labels, which is formulated as:

$$L_{sm} = 1 - \frac{2 \times |Y \cap \widehat{Y}| + 1}{|Y| + |\widehat{Y}| + 1}, \tag{7}$$

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) [25] is adopted when calculating L_{sm}.

For adaptive offset and SPW prediction, the ratio loss L_{ratio} is adopted to compute their gradients:

$$L_{ratio}(P,\ \widehat{P}) = \log \frac{max\left(P,\ \widehat{P}\right)}{min\left(P,\ \widehat{P}\right)}, \tag{8}$$

where P and $\widehat{P}$ are the ground-truth and prediction, respectively. Therefore, L_{ratio} based adaptive offset loss L_{o_a} and SPW loss L_{SPW} can be defined as:

$$L_{o_a} = L_{ratio}(o_a,\ \widehat{o_a}), \tag{9}$$

$$L_{SPW} = L_{ratio}(SPW,\ \widehat{SPW}), \tag{10}$$

where o_a and $\widehat{o_a}$ are the ground-truth (computed by the Eq. (6)) and predicted o_a. SPW and $\widehat{SPW}$ are the ground-truth (computed by the Eq. (4)) and predicted SPW.

The final loss function L used for training the proposed network is given by:

$$L = \lambda_1 L_{sm} + \lambda_2 L_{o_a} + \lambda_3 L_{SPW}, \tag{11}$$

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.

4. Experiments

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

4.1. Datasets

SynthText [8] 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 ASMTD. **MSRA-TD500** [33] 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 [32] as training data. **Total-Text** [4] 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** [20] is a dataset for testing the model performance on line-level curved text instances, which has 1000 training images and 500 testing images.

4.2. Implementation Details

The feature extracting stem is pre-trained on ImageNet [6] and the whole network is pre-trined on SynthText [8]. In the fine-tuning stage, Adam [13] is employed to train the model. The initial learning rate is set to 0.001 and is adjusted by the 'poly' strategy used in [37]. The training batch size is set to 16. We initialize the weights of feature merging branch and output layers with the strategy in [9]. For all datasets, the blurred text regions labeled as DO NOT CARE are ignored during the training. 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).

4.3. Ablation Study

The ablation study is conducted to show the effectiveness of the proposed Adaptive Shrink-Mask (ASM) and Superpixel Window (SPW).

Adaptive Shrink-Mask. As shown in Tab. 1 #2, since the proposed ASM can avoid missing detection effectively, which brings 1.3% improvement in terms of F-measure on MSRA-TD500. At the same time, we evaluate our method on Total-Text benchmark with different settings. Specifically, in Tab. 2, ASM achieves 1.2% and 1.1% gain in F-measure when IoU are set as 50% and 75% respectively. Particularly, for the ASM based model, the corresponding

#	Image scale for testing (S : 736)						
		ASM	SPW	Precision (%)	Recall (%)	F-measure (%)	FPS
1	baseline			87.2	81.6	84.3	64.2
2	baseline+	✓		88.9	82.5	85.6	62.5
3	baseline+	✓	✓	89.8	83.1	86.3	62.5

Table 1. Detection results with different settings on MSRA-TD500. "S: 736" means that the shorter side of each testing image is resized to be 736 pixels. "baseline" means the framework equipped with traditional shrink-mask only. "ASM" indiates adaptive shrink-mask. "SPW" means super-pixel window.

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

Table 2. Detection results with different settings on Total-Text. "ASM" indiates adaptive 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%.

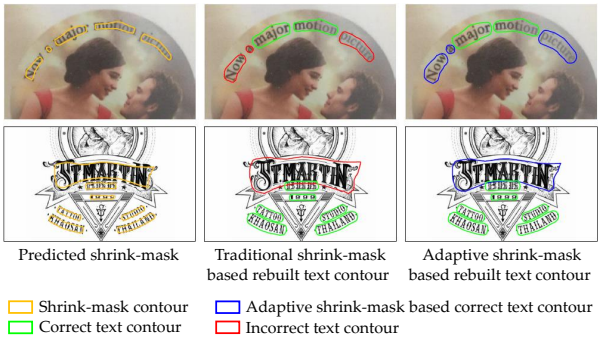


Figure 7. Qualitative comparisons with traditional shrink-masks based rebuilt text contours and adaptive shrink-masks based rebuilt text contours.

F-measure with 75% IoU achieves 85.5%, which outperforms the F-measure of baseline with 50% IoU by 0.6%, which shows the ASM can improve the integrity of rebuilt text contours. Moreover, we show some examples in Fig. 7. Traditional shrink-masks based rebuilt text contours (Fig. 7 red colored geometries) are smaller than ground-truth, which leads to text information loss. Adaptive shrink-masks based rebuilt text contours (Fig. 7 blue colored geometries) avoid these problems effectively. In Fig. 7 second row, adaptive shrink-masks based rebuilt text contours are accurate, even when the shrink-masks are larger than ground-truth. These experiments demonstrate the superiority of the proposed ASM for detecting text instances.

Super-pixel Window. Benefiting from the rich semantic context brought by SPW, our method achieves significant performance gain in terms of F-measure. As shown in Tab. 1 #3, the ASM and SPW bring 2% improvement in F-measure. Moreover, since the SPW does not participate in the reconstructing process of text contours, the model equipped with SPW brings no extra computational cost to the inference process (as shown in Tab. 1 #2–#3). The experiment results prove the SPW not only can improve the detection accuracy but also ensures high detection speed.

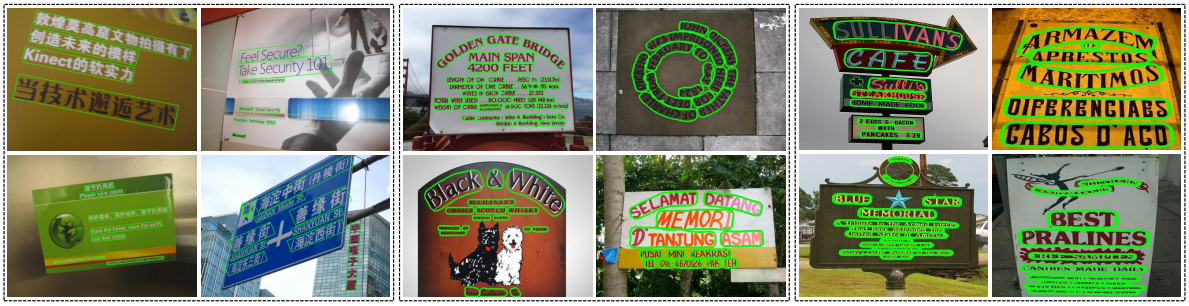
Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [38]	87.3	67.4	76.1	**13.2**
	PixelLink [5]	83.0	73.2	77.8	-
	TextSnake [21]	83.2	73.9	78.3	1.1
	RRD [16]	87.0	73.0	79.0	10
	TextField [31]	87.4	75.9	81.3	-
	CornerNet [14]	87.6	76.2	81.5	5.7
	CRAFT [2]	88.2	78.2	82.9	8.6
	SAE [26]	84.2	81.7	82.9	-
	OPMP [35]	86.0	83.4	84.7	1.6
	SAVTD [7]	89.2	81.5	**85.2**	-
RT	DB [15]	90.4	76.3	82.8	59.5*
	PAN [29]	84.4	83.8	84.1	30.2
	Ours	89.8	83.1	**86.3**	**62.5**

Table 3. 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. "*" means the result measured in our environment.

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

In this section, we compare our method with the related state-of-the-art (SOTA) methods on MSRA-TD500, Total-Text, and CTW1500 datasets to show the superiority of the ASMTD in both detection accuracy and speed.

MSRA-TD500: Long Straight Text Benchmark. The results on MSRA-TD500 are shown in Tab. 3. ASMTD achieves the F-measure of 86.3% at an astonishing detection speed (62.5 FPS). For real-time text detection methods, our method outperforms DB [15] by 3.5% about F-measure because text contours can be rebuilt accurately even when the predicted shrink-masks deviate from ground-truth. Moreover, the lightweight feature merging branch facilitates ASMTD to run 2x times faster than PAN [29]. Compared with OPMP [35] and SAVTD [7], since our method is equipped with a simpler framework and SPW, which helps to surpass them by 1.6%–1.1% in F-measure and runs 40x times faster than them at least. The detection results demonstrate the superiority of our method for detecting long straight text instances. We also illustrate some



(a) MSRA-TD500 (b) Total-Text (c) CTW1500

Figure 8. Illustration of some qualitative detection results of the proposed ASMTD.

Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [38]	50.0	36.2	42.0	-
	TextSnake [21]	82.7	74.5	78.4	-
	TextField [31]	81.2	79.9	80.6	-
	TextRay [28]	83.5	77.9	80.6	-
	OPMP [35]	85.2	80.3	82.7	3.7
	LOMO [34]	87.6	79.3	83.3	-
	FCENet [39]	87.4	79.8	83.4	-
	CRAFT [2]	87.6	79.9	83.6	-
	ABCNet [19]	87.9	81.3	84.5	-
	ReLaText [22]	86.0	83.3	84.8	**10.6**
	ContourNet [30]	86.9	83.9	85.4	3.8
	DRRG [36]	86.5	84.9	**85.7**	-
RT	DB [15]	88.3	77.9	82.8	47.7*
	PAN [29]	89.3	81.0	85.0	39.6
	Ours	88.5	83.8	**86.1**	**70.9**

Table 4. Comparison with related methods on Total-Text. "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. "*" means the result measured in our environment.

Type	Methods	P (%)	R (%)	F (%)	FPS
NRT	EAST [38]	78.7	49.1	60.4	**21.2**
	TextSnake [21]	67.9	85.3	75.6	1.1
	TextField [31]	83.0	79.8	81.4	-
	TextRay [28]	82.8	80.4	81.6	-
	OPMP [35]	85.1	80.8	82.9	1.4
	LOMO [34]	85.7	76.5	80.8	-
	FCENet [39]	85.7	80.7	83.1	-
	CRAFT [2]	86.0	81.1	83.5	-
	ABCNet [19]	81.4	78.5	81.6	-
	ReLaText [22]	84.8	83.1	84.0	-
	ContourNet [30]	83.7	84.1	83.9	4.5
	DRRG [36]	85.9	83.0	**84.4**	-
RT	DB [15]	84.8	77.5	81.0	53.1*
	PAN [29]	86.4	81.2	83.7	39.8
	Ours	87.8	80.3	**83.9**	**72.1**

Table 5. 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. "*" means the result measured in our environment.

qualitative results in Fig. 8 (a).

Total-Text: Word-Level Curved Text Benchmark. As we can see from Tab. 4, ASMTD achieves 86.1 in F-measure and 70.9 in FPS, which outperforms previous state-of-the-art (SOTA) methods. Specifically, since the SPW provides rich semantic context for prediction tasks, the ASMTD outperforms ReLaText [22] and DRRG [36] by 2.7% and 0.4% in F-measure. At the same time, ASM based text representation method saves much computational cost for the inference process, which makes our method can run 6.7 times faster than the fastest non-real-time method (ReLaText [22]). The results demonstrate the effectiveness of ASMTD for detecting word-level curved text instances. Moreover, as shown in Fig. 8 (b), though many text instances are close to each other, our method still can distinguish them accurately because of the superiority of ASM based text representation method.

CTW1500: Line-Level Curved Text Benchmark. As shown in Tab. 5, since ASM helps ASMTD to enjoy the same efficiency as existing real-time methods and the feature merging branch reduces the network complexity, our model runs 19 FPS and 32.3 FPS faster than DB and PAN, respectively. Moreover, SPW brings significant improvement for recognizing shrink-masks by providing rich context information and defining the semantic difference with interval region, which makes ASMTD outperform most non-real-time methods in F-measure. Although our method is a little lower (0.5%) than DRRG [36] in F-measure, ASMTD has a least 70x times faster speed (72.1 FPS) than it. The evaluation results on the CTW1500 show the model robustness for detecting long curved text instances. Moreover, some qualitative detection results on CTW1500 are illustrated. As we can see from Fig. 8 (c) first row and second column, even 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.

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

Table 6. Cross-dataset evaluations on line-level datasets.

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 7. Time consumption of ASMTD 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.

Methods	GFLOPs	F-measure (%)		
		MSRA-TD500	Total-Text	CTW1500
PAN [29]	63.88	84.1	85.0	83.7
DB [15]	52.54	82.8	82.8	81.0
Ours	46.96	86.3	86.1	83.9

Table 8. Comparison of computational cost and performance of different real-time detectors. "GFLOPs" indicates floating point of operations.

4.5. Cross Dataset Text Detection

We further verify the generalization ability of the proposed ASMTD by cross evaluation experiments in this section. Since both MSRA-TD500 and CTW1500 are line-level benchmarks, we design two groups experiments on them. Specifically, we train our model on MSRA-TD500 and test it on CTW1500 at first. Then, the proposed ASMTD is trained on CTW1500 and evaluated on MSRA-TD500. As shown in Tab. 6, since the SPW provides rich semantic information for prediction tasks and ASM makes text contours can be reconstructed according to inaccurate shrink-masks, which facilitates our method outperforms the TextSnake [21] (in Tab. 5) by 2.7% in F-measure on CTW1500 and surpasses EAST [38], PixelLink [5], and TextSnake [21] (in Tab. 3) by 1.8% F-measure at least on MSRA-TD500. The experiment results verify the proposed ASMTD enjoys strong generalization and robustness for diverse shaped text instances in different benchmarks.

4.6. Speed Analysis

To verify the superiority of our method in terms of detection speed and computational resources, we show the time consumption of ASMTD and compare the computational cost with related SOTA methods. As illustrated in Tab. 7,

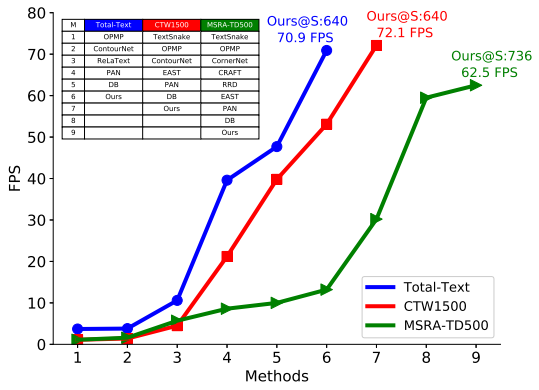


Figure 9. Comparison of detection speed on multiple benchmarks. "S" is the shorter side of each testing image for our method.

since the ASM helps text contours can be rebuilt through extending shrink-mask contours by adaptive offsets directly, which makes the post-processing takes about 14% of the total time consumption. At the same time, because the SPW can be removed from the inference stage and the lightweight feature merging branch saves much computational cost, the head only takes about 36% of the total time consumption (as shown in Fig. 3). The experiment results demonstrate the effectiveness of the proposed ASM and SPW for improving the model detection speed. Additionally, we verify the superiority of ASMTD in detection speed. As shown in Fig. 9, our method outperforms other related SOTA methods a lot. Moreover, we show the computational cost of existing real-time methods in Tab. 8. Because ASMTD 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.

5. Conclusion

In this paper, we propose a novel real-time framework to detect arbitrary-shaped texts with high detection accuracy. We firstly propose the Adaptive Shrink-Mask (ASM) to fit text instances by shrink-masks and adaptive offsets, which improves the accuracy of rebuilt text contours effectively. Then, we introduce the Super-pixel Window (SPW) to provide rich semantic context for the prediction tasks, which brings significant improvements for the reliability of the discrimination of shrink-masks and the prediction of adaptive offsets. 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 concatenate multi-level feature maps, which further facilitates the detection speed. Extensive experiments demonstrate the effectiveness of the AES and SPW. Comparison experiments show that the proposed ASMTD is superior to all state-of-the-art real-time methods.

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4715–4723, 2019. 2

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019. 2, 6, 7

[3] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. Towards the unseen: Iterative text recognition by distilling from errors. In *ICCV*, pages 14950–14959, 2021. 1

[4] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *ICDAR*, volume 1, pages 935–942, 2017. 5

[5] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *AAAI*, pages 6773–6780, 2018. 6, 8

[6] Jia Deng, Wei Dong, Richard Socher, Lijia Li, Kai Li, and Feifei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[7] Wei Feng, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Semantic-aware video text detection. In *CVPR*, pages 1695–1705, 2021. 1, 6

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[11] R Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C Popat. A scalable handwritten text recognition system. In *ICDAR*, pages 17–24, 2019. 2

[12] K Karthick, KB 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*, 8(1):2277–3878, 2019. 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 2, 6

[15] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020. 1, 2, 3, 6, 7, 8

[16] Minghui Liao, Zhen Zhu, Baoguang Shi, Guisong Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *CVPR*, pages 5909–5918, 2018. 2, 6

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[18] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11962–11972, 2020. 1

[19] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, pages 9809–9818, 2020. 2, 7

[20] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 5

[21] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, pages 20–36, 2018. 2, 6, 7, 8

[22] Chixiang Ma, Lei Sun, Zhuoyao Zhong, and Qiang Huo. Relatext: exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks. *Pattern Recognition*, 111:107684, 2021. 2, 7

[23] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571, 2016. 5

[24] Nguyen Nguyen, Thu Nguyen, Vinh Tran, Minh-Triet Tran, Thanh Duc Ngo, Thien Huu Nguyen, and Minh Hoai. Dictionary-guided scene text recognition. In *CVPR*, pages 7383–7392, 2021. 1

[25] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016. 5

[26] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *CVPR*, pages 4234–4243, 2019. 2, 6

[27] R Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 35(7):56–63, 1992. 2, 4

[28] Fangfang Wang, Yifeng Chen, Fei Wu, and Xi Li. Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection. In *ACMMM*, pages 111–119, 2020. 2, 7

[29] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, pages 8440–8449, 2019. 1, 2, 6, 7, 8

[30] Yuxin Wang, Hongtao Xie, Zhengjun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *CVPR*, pages 11753–11762, 2020. 2, 7

[31] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 28(11):5566–5579, 2019. 2, 6, 7

[32] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014. 5

[33] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090, 2012. 5

[34] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *CVPR*, pages 10552–10561, 2019. 2, 7

[35] Sheng Zhang, Yuliang Liu, Lianwen Jin, Zhongrong Wei, and Chunhua Shen. Opmp: An omnidirectional pyramid mask proposal network for arbitrary-shape scene text detection. *IEEE Transactions on Multimedia*, 23:454–467, 2020. 2, 6, 7

[36] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *CVPR*, pages 9696–9705, 2020. 1, 4, 7

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 5

[38] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *CVPR*, pages 5551–5560, 2017. 2, 6, 7, 8

[39] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhuanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. *arXiv preprint arXiv:2104.10442*, 2021. 1, 2, 4, 7