Compensating for the Incomplete with the Complete: An Efficient Scene Text Detector

Xu Han and Qi Wang, *Senior Member, IEEE*

***Abstract*—Scene text reading is an essential component of scene understanding. As its fundamental requirement, text detection has garnered increasing attention. Segmenting the text kernel and extending it to reconstruct text instances is efficient and effective among the various methods. However, the incomplete semantic features of text kernels and the high similarity between kernels and texts make it hard to extract kernels from images accurately. Considering the above, we propose an efficient text detector, termed CIC, which comprises a bidirectional information transfer module (BITM), a dual knowledge integration module (DKIM), and a cross-verification module (CVM). The former generates collaborative information between the predicted text and kernel via the proposed differentiable adaptive gap operator. It forces mutual restraint and collaborative progress between the predictions of text and kernel. Unlike BITM, DKIM designs a knowledge fuse scheme, which helps to locate kernels accurately under the guidance of the complete semantic feature of texts. Intuitively, as the kernel is generated by shrinking the text, the kernel pixel is only presented in the text area. Based on this criterion, the CVM further utilizes text predictions to constrain kernel predictions and reduce false positive predictions. Ablation experiments demonstrate the effectiveness of the proposed BITM, DKIM, and CVM. Extensive experiments show the proposed CIC outperforms existing state-of-the-art (SOTA) methods on five public datasets from different scenes. The code is available at https://github.com/fengmulin/CIC.**

***Index Terms*—Real-time, text detection, multi-scene, semantic segmentation**

I. Introduction

OVER the past years, scene text detection has significantly improved [1], [2], [3], [4], driven by advancements in object detection and semantic segmentation [5]. It has gained increasing attention as it provides crucial information for various intelligent applications, including image recognition, autonomous driving, and image retrieval. As a critical part of text recognition in natural scenes, text detection has developed to current real-time arbitrary-shaped text detection.

Existing scene text detection methods can generally be categorized into regression-based and segmentation-based approaches. The primary advantage of segmentation-based methods lies in their flexible pixel-level prediction, which can effectively represent arbitrary-shaped instances. Intuitively, segmenting text regions and performing contour extraction can effectively detect text. However, as shown in Fig. 1, geographically close texts are sometimes overlapped in scenes, causing their contour properties to be overlooked. It may lead to unclear instance boundaries and even multiple instances being misclassified as a single instance. To alleviate this issue, some methods [6], [7], [8] predict text kernels and expand them to reconstruct instances. As shown in Fig. 1, the text kernel is generated by inward shrinking of the text. Compared to the original text, it retains only the core region, effectively mitigating the issue of instance boundary overlap. However, the aforementioned methods overlook that the text kernel is an artificially defined geometric concept with incomplete semantic features. Furthermore, texts, text kernels, and their gaps share the same local texture information, which complicates the precise separation of kernels from texts. As illustrated in Fig. 2, although texts and some patterns enjoy similar low-level texture features, fortunately, we can distinguish them according to the different high-level semantic features. However, for the text kernel, as the shrinkage destroys its semantic integrity, it not only has analogous local information with some patterns but also has no significantly different high-level semantic feature compared to these patterns, which increases the challenge of differentiating them. For the above reasons, it is significantly more challenging for the model to learn the

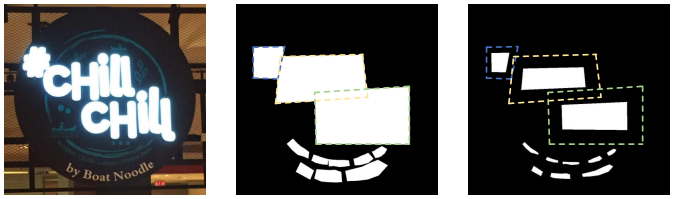


Fig. 1. The visualization of text map and kernel map. Geographically close text edge pixels overlap. Text kernel enjoys incomplete semantic features.

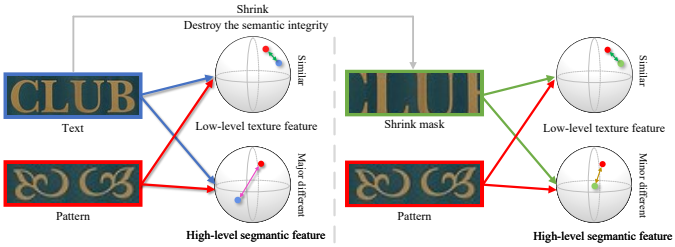


Fig. 2. The difference between the text, kernel, and some patterns on the low-level texture feature and high-level semantic features.

This work was supported by the National Natural Science Foundation of China under Grant U21B2041 and 62471394. (Corresponding author: Qi Wang.)

X. Han is with the School of Computer Science, and with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (E-mail: hxu04100@gmail.com).

Q. Wang is with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (E-mail: crabwq@gmail.com).

features of text kernel regions than text regions.

To consider the above problems, a bidirectional information transfer module (BITM) is proposed, which encourages the model to generate collaborative information between the texts and kernels via the proposed differentiable adaptive gap operator. Unlike previous methods that recognize kernel features in isolation, the collaborative information associates text and kernel features to help optimize the text and kernel prediction branches mutually, which encourages our model to utilize complete text semantic features to guide the learning of complicated, incomplete kernel semantic features. Moreover, BITM emphasizes the difference in features between texts and kernels, which helps the proposed CIC distinguish them effectively and separate kernels from texts accurately. Benefiting that BITM can be removed during the inference phase, CIC achieves real-time performance. Unlike BITM which focuses on collaborative information from the result level, DKIM further establishes contact between the text and kernel from the feature perspective. It steers the model to cognize text kernel accurately with the help of complete text semantic features and promotes text and kernel information fuse, encouraging two types of knowledge to interact and collaborative progress. Furthermore, since the kernel is generated by shrinking the text inwards, the kernel is presented only in areas where text exists. In view of this, a cross-verification module (CVM) is proposed, which constrains kernel prediction through text prediction. It strengthens the intrinsic connection between the two predictions and deepens the understanding of the two types of knowledge for the model, which suppresses the false positive predictions.

Overall, predicting text alone tends to cause instances to stick, and predicting kernels alone is tricky. Based on the above BITM, DKIM, and CVM, we construct an arbitrary-shaped scene text detector, termed CIC, short for **C**ompensating for the **I**ncomplete with the **C**omplete. It combines the strengths of text and kernel on several levels, discarding their weaknesses. The proposed method has the following contribution:

1) A bidirectional information transfer module (BITM) is proposed to generate the collaborative information between the predicted text and kernel. The information helps to locate kernels under the guidance of the complete semantics of texts and to distinguish the differences between them for recognizing kernels from texts accurately. In addition, it is a training-only module that can be removed during inference, reducing computational.
2) A dual knowledge integration module (DKIM) is proposed. Unlike BITM, it facilitates interactions between text and kernel formation from a feature perspective. It utilizes a feature fuse scheme to enhance the model's understanding of text and kernel knowledge, enabling accurate text kernel localization by high-level semantic feature interaction between the text and text kernel.
3) A cross-verification module (CVM) is proposed, which leverages the criterion that no text, no kernel. It suppresses kernel presence in regions without text, furthering the connection between text and kernel predictions and the understanding of the model with respect to both types of knowledge.
4) An effective real-time scene text detector is proposed called CIC, which achieves SOTA performance on multiple benchmarks of text detection based on a lightweight network, including multi-directional text, irregular text, and horizontal text.

The remaining structure is as follows. Some related works about scene text detection are presented in Section II. The proposed BITM, DKIM, and CVM are presented in Section III. In Section IV, ablation experiments are performed to demonstrate the validity of the proposed method. Moreover, we compare the experimental results with other advanced methods. Finally, Section V gives a conclusion of this paper.

II. Related Work

With the continuous growth of deep learning, they dominate the current scene text detection field. Furthermore, these detection approaches can be broadly classified into regression-based methods and segmentation-based methods.

A. Regression-Based Methods

Regression-based methods are inspired mainly by traditional object detection algorithms. Liao *et al.* proposed the TextBoxes algorithm [9] to modify the anchor and convolution kernel to detect texts and then proposed the TextBoxes++ [10] to detect multi-directional text instances by adding angle parameters. Based on Faster R-CNN [11], Matas *et al.* [12] proposed the RPN to detect omnidirectional text instances. Zhou *et al.* [13] used FCN [14] to predict the angles, scores, and text boxes. He *et al.* proposed SSTD [15], which employed a text region attention mechanism for text region detection. The above methods are limited when it comes to curved texts. In addition, complex post-processing [16] also limits the development of such methods. To deal with scene text of arbitrary shape, Zhu *et al.* [17] used Fourier Contour Embedding to represent the scene text contours. The reconstruction of instances was achieved by the prediction of the Fourier Signature Vector and the scores of text and text center. TextRay [18] and ABCNet [19] represented the text contour based on the Polar coordinate system and Bezier curves, respectively. PCR [20] proposed a progressive contour regression method to describe text instances, first represented by horizontal proposals, then further evolved into multi-directional text instances, and finally into arbitrary-shaped text. Tang *et al.* [21] used a transformer to group features for excellent performance and did not require post-processing. Existing regression-based methods address arbitrarily shaped text by predicting a series of parameters or regressing and iterating contour points. These methods can effectively represent highly curved instances. However, compared to segmentation-based approaches, their main drawbacks lie in the limited ability to capture local text details and low processing efficiency, making it challenging to meet the real-time inference requirements of certain practical applications.

B. Segmentation-Based Methods

The results of segmentation-based text detection approaches are usually obtained by predicting the probability of each

pixel. Many methods are based on this to detect arbitrary-shaped text. PSE-Net [6] predicted text regions of different scales to separate neighbor instances and then recovered text instances through a progressive expansion algorithm. Pixellink [22] predicted whether each pixel is text or not and predicted the affinities between adjacent pixels to determine whether they belong to the same text. Lyu *et al.* [23] distinguished text boundaries by predicting four different corner regions of the texts and combining them to obtain the results. Xu *et al.* [24] proposed a method named TextField that can localize texts and distinguish text boundaries by predicting text scores and direction vectors. Zhang *et al.* proposed KPN [25] to define different texts as instance-independent feature maps to separate neighboring instances. LeafText [26] proposed a text representation method, that accurately models text of various shapes. Baek *et al.* [27] predicted character regions by synthesizing character-level labels of the dataset and estimated the affinity between characters to obtain results. Tian *et al.* [28] mapped pixels to a high-dimensional space, urging pixels that are part of the same text instance to be closer together and pixels of different instances to be farther. This can effectively separate adjacent text instances. Yu *et al.* [29] introduced CLIP [30] model to the existing scene text detection methods, which enhanced the detection model's domain adaptation ability. Chen proposed a discrete cosine transform network [31], whose backbone is based on the CLIP [30] to extract refined features. Guo *et al.* [32] built a comprehensive multilingual scene text dataset and proposed a multilingual scene text detector that integrated an adaptive spatial feature fusion module to improve the feature pyramid network [33]. Although these methods can effectively cope with arbitrarily shaped text, they are still unsatisfactory in terms of speed due to their complex structure and numerous post-processing steps. The characteristic of real-time text detection approaches is that they ensure the accuracy of the model and also keep a high inference speed. According to some shortcomings of the conventional shrink mask, CM-Net [34] proposed the Concentric Mask in a targeted manner, which has the same center as the text instances and takes advantage of the local features of the text instances. Coupled with the diverse losses, it achieved excellent accuracy while ensuring the detection speed. PAN [35] first predicted the text kernel and text separately and then expanded kernel regions to text regions with the help of the predictions of similarity vectors and text. DB-Net [7] was based on the conventional predictions of the text kernels and added the prediction of the threshold map. Then they used the threshold map to enhance the constraints on the text kernel, and the binarization was put into the network for training, thereby improving the detection accuracy of the model. CT [4] proposed a novel text representation method that splits text instances into text kernel regions and centripetal displacements, and the proposed post-processing method was also faster. ZTD [36] proposed two zoomed view modules to simulate the camera's zoom process. RSMTD [37] designed an efficient text representation for decoupling instances and the shrinkage masks and mitigates errors caused by shrinkage masks that deviated from the truth by predicting dilation values. ADNet [8] also used prediction instead of geometric calculation to adaptively reconstruct text contours. The development of kernel-based methods has been driven by improvements in kernel representation and contour reconstruction techniques. However, these methods overlook the intrinsic relationship between text and text kernels. Unlike the above methods, Wang *et al.* [38] predicted the text area and utilized the text separatrix to differentiate individual instances. TextBPN [39] and TextBPN++ [40] extracted contour points based on the segmentation and updated these points iterly. Segmentation-based methods primarily rely on pixel-level predictions, causing the model to focus on capturing low-level texture features while overlooking high-level semantic features related to text. Compared to regression-based methods, their accuracy has certain limitations.

III. Methodology

The whole framework of the CIC is introduced first in this section. Then, we describe the bidirectional information transfer module (BITM), the dual knowledge integration module (DKIM), and the cross-verification module (CVM) in detail, respectively. In addition, the generation process of the label is defined. Finally, the loss functions are introduced.

A. Overall Framework

The overall architecture of the proposed CIC is shown in Fig. 3, which includes the backbone, text prediction head, kernel prediction head, bidirectional information transfer module (BITM), dual knowledge integration module (DKIM), cross-verification module (CVM), and post-processing. During the training stage, a multi-level feature map is generated through the backbone first. Then, the text and kernel prediction head predicts the text and kernel map based on it. Next, the BITM generates collaborative information via the proposed differentiable adaptive gap operator. It helps locate kernels accurately with the complete semantic features of texts. DKIM enhances the information mutual between the text and kernel. CVM leverages the prediction of the text to guide the prediction of the kernel furthermore, which suppresses false positive predictions. In the inference stage, the BITM, CVM, and text prediction head can be removed, which means only the backbone, kernel prediction head, and expanded post-processing are reserved. Therefore, the proposed CIC achieves real-time performance.

Specifically, the ResNet [41] with deformable convolution [42], [43] and Feature Pyramid Network (FPN) [33] are used as the backbone. It first generates four scale feature maps and then resizes and concatenates them to generate a multi-scale feature map $F_m \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$. Based on it, two smooth layers are utilized to generate text feature maps F_t and kernel feature maps F_k, which can be formulated as follows:

$$F_t = \mathrm{ReLU}_{\mathrm{BN}}(\mathrm{Conv}_{3\times3}(F_m)), \tag{1}$$

$$F_k = \mathrm{ReLU}_{\mathrm{BN}}(\mathrm{Conv}_{3\times3}(F_m)), \tag{2}$$

DKIM fuses the F_t and F_k to obtain fused feature maps F_f The text map prediction head adopts the segmentation structure as follows:

$$H = \mathrm{ReLU}_{\mathrm{BN}}(\mathrm{ConvT}_{2\times2}(F_t)), \tag{3}$$

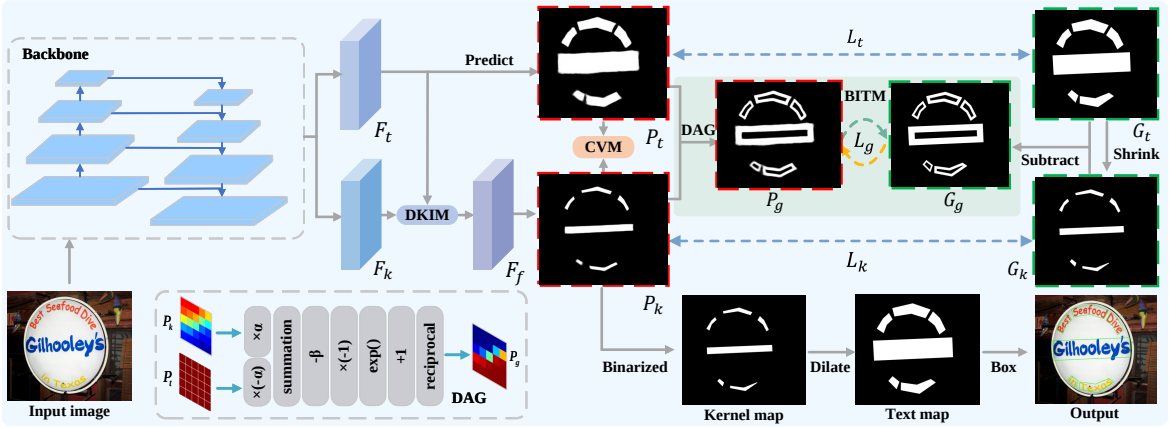
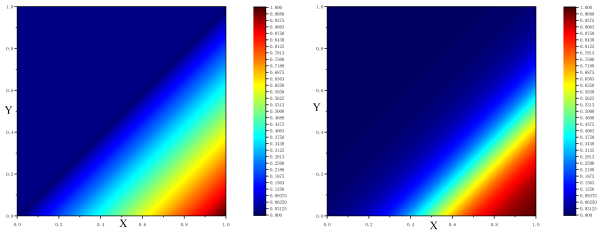


Fig. 3. The overall framework of CIC. In the inference phase, we only need to predict the kernel to post-process to obtain the final detection results. P_k, P_t, and P_g denote the predicted kernel, text, and gap map, respectively. G_k, G_t, and G_g represent the ground truth of the kernel, text, and gap map, respectively.

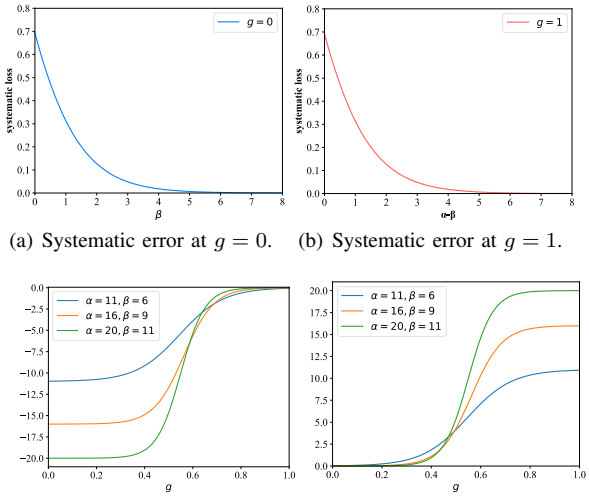


(a) Text−Kernel (b) DAG (Text, Kernel)

Fig. 4. The visualization of direct subtraction and DAG. When the predicted text value and the predicted text kernel value of a pixel take different values, the gap value is calculated by the two methods. The x-axis and y-axis represent the text predictions and the text kernel predictions, respectively.

$$P_t = \text{Sigmoid}(\text{ConvT}_{2\times2}(H)), \tag{4}$$

where H, P_t, and ConvT represent the hidden feature maps, predictions of the text map, and transposed convolution operation, respectively.



(a) Systematic error at $g = 0$. (b) Systematic error at $g = 1$.

(c) Derivative of l_+. (d) Derivative of l_-.

Fig. 5. Illustrations of gradient and systematic loss of DAG, where $g = P_t - P_k$.

B. Bidirectional Information Transfer Module

Many methods [7], [35], [34], [6] predict text kernels and expand them to reconstruct text instances directly. However, these methods ignore that the text kernel is an artificially-defined geometric concept, which leads to incomplete semantic features and makes it hard for models to recognize text kernels accurately. In addition, texts, text kernels, and their gaps enjoy the same local texture information, which causes difficulty in separating kernels from texts precisely. To alleviate the above problems, a bidirectional information transfer module (BITM) is proposed, which helps the proposed model generate collaborative information between the predicted texts and kernels via the proposed differentiable adaptive gap operator. It achieves the information interaction and utilizes complete text semantic features to guide the prediction of kernels to recognize them from texts accurately.

BITM introduces a differentiable adaptive gap (DAG) operator that efficiently fits the gap for different text and text kernel fetching values. The formula is as follows:

$$P_g = \frac{1}{1 + e^{-(\alpha \times P_t - \alpha \times P_k - \beta)}}, \tag{5}$$

where α and β are manually set parameters that will be learned along with the model. The α and β are initialized as 20 and 11, respectively. Note that DB [7] focuses on the kernel predictions only. We mainly explore the relationship between text, gap, and kernel that use it to improve detection performance. The results of real relationships of text and text kernels and calculation by DAG are shown in Fig. 4. For a visual comparison, we replace the negative value of the direct subtraction with 0. Moreover, we explore the systematic error of the DAG as an example of cross-entropy loss as well as its gradient. The DAG is defined as $f(g) = \frac{1}{1+e^{\beta - \alpha g}}$, where $g = P_t - P_k$. Fig. 5(a) and Fig. 5(b) are the systematic errors of DAG for $g = 0$ and $g = 1$, respectively. The loss l_+ for positive samples and the loss l_- for negative samples are

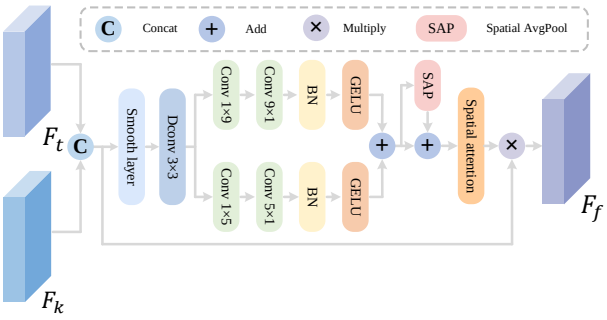


Fig. 6. The overall structure of the proposed DKIM. F_t and F_k represent the input text features and kernel features, respectively, while F_f denotes the output fused features.

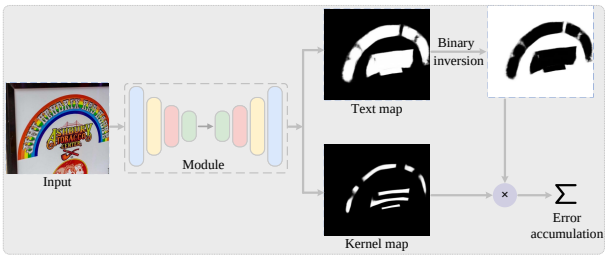


Fig. 7. The overall structure of the proposed CVM. The predicted text map is processed with binary inversion and then multiplied by the kernel prediction to calculate error accumulation.

denoted as:

$$\begin{aligned} l_+ &= -\log \frac{1}{1+e^{\beta-\alpha g}}, \\ l_- &= -\log\left(1-\frac{1}{1+e^{\beta-\alpha g}}\right). \end{aligned} \tag{6}$$

Their differentiation can be calculated as follows:

$$\begin{aligned} \frac{\partial l_+}{\partial x} &= -\alpha f(x) e^{\beta-\alpha g}, \\ \frac{\partial l_-}{\partial x} &= \alpha f(x). \end{aligned} \tag{7}$$

They are shown in Fig. 5(c) and Fig. 5(d). We conclude that the gradient is related to α, and the positive and negative samples are optimized with different scales, thus facilitating the training of the model.

The usefulness of the BITM can also be demonstrated from another perspective. We define the label of the gap as $\dot{P}_g$ and the training target as $P_g = \dot{P}_g$. Thus, it can be expressed as:

$$\frac{1}{1+e^{-(\alpha \times P_t - \alpha \times P_k - \beta)}} = \dot{P}_g. \tag{8}$$

This can be transformed into:

$$P_k = \frac{ln(\frac{1}{\dot{P}_g}-1) + \alpha \times P_t - \beta}{\alpha}. \tag{9}$$

Since we assume that the pseudo-label $\tilde{P}_t$ of the text is equal to the predicted value of the text P_t. The pseudo-label of the kernel $\tilde{P}_k$ can be expressed as:

$$\tilde{P}_k = \frac{ln(\frac{1}{\dot{P}_g}-1) + \alpha \times \tilde{P}_t - \beta}{\alpha}, \tag{10}$$

which is the dual supervision of P_k. Fig. 4 shows the calculation results of the DAG when the text prediction value and the kernel prediction value change.

C. Dual Knowledge Integration Module

Different from BITM, DKIM emphasizes the interaction between text and kernel information at the feature level. Based on text feature maps F_t and kernel feature maps F_k, a fused feature map F_f is generated, which combines text and kernel information simultaneously. This process is shown in Fig. 6, which can be formulated as follows:

$$F = \text{Concat}(F_k, F_t), \tag{11}$$

$$H_1 = \text{Smooth}(\text{DWConv}_{3\times3}(F)), \tag{12}$$

$$H_2 = \sum_s \text{Conv}_{s\times1}(\text{Conv}_{1\times s}(H_1)), s = 5, 9 \tag{13}$$

$$H_3 = H_2 + \text{SAP}(H_2), \tag{14}$$

$$F_f = \text{Att}(H_3) \otimes F, \tag{15}$$

where "SAP" and "Att" represent spatial AvgPool and spatial attention operation. Note that, spatial attention is a lightweight convolution operation.

Similarly, based on the F_f, a segmentation head is performed to obtain the kernel map:

$$H_4 = \text{ReLU}_{\text{BN}}(\text{ConvT}_{2\times2}(F_f)), \tag{16}$$

$$P_k = \text{Sigmoid}(\text{ConvT}_{2\times2}(H_4)), \tag{17}$$

where P_k represents the prediction of the kernel map.

D. Cross-verification Module

As the text kernel is generated by shrinking the text instance, where there is no text there is no text kernel. Based on it, a cross-verification module (CVM) is proposed, which suppresses the presence of kernels in areas where text is non-existent, helping the model further comprehend the text and kernel features. Specifically, as shown in Fig. 7, the prediction of the text map is transformed into its complement by subtraction, aiming to obtain areas without kernels theoretically. Secondly, the prediction of the text kernel is multiplied by inversion text protection, which represents the accumulation of theoretical errors within the model. This process can be formulated as follows:

$$\tilde{P}_t = 1 - P_t, \tag{18}$$

$$\mathcal{L}_{cv} = \sqrt{\frac{\sum_i \tilde{P}_t^i \times P_k^i}{S}} \times \eta \tag{19}$$

where $\mathcal{L}_{cv}$ and S represent the cross verification loss and number of pixels.

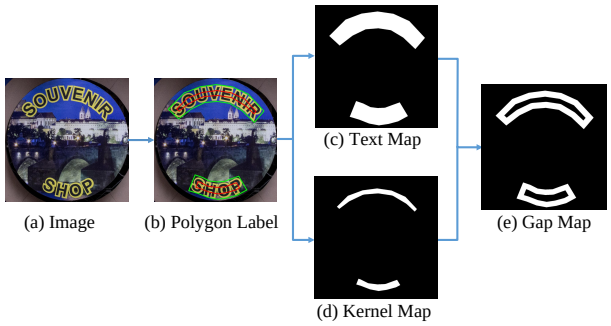


Fig. 8. Label generation procession. The green ones are text labels, and the red ones are kernel labels. The gap label is obtained by subtracting the text kernel label from the text label.

E. Label Generation

Inspired by PSENet [6], we further refine the proposed label generation method. Each text region is described by n points for a given text image. This number is different for datasets. For example, for the ICDAR2015, this number is 4, and for the Total-Text, this number is not fixed. As shown in Fig. 8, the text kernels are generated from the text according to the Vatti clipping algorithm [44]. According to the perimeter and area of the instance, the amount of shrinkage of the text is calculated as follows:

$$D_i = \frac{A_i \times (1 - r^2)}{P_i}, \tag{20}$$

where r is the shrinkage factor, set to 0.4 by default. D_i is the distance that the ith instance shrinks inward. A_i, P_i are the area and the perimeter of the ith instance, respectively. As shown in Fig. 8, for the gap labels, we subtract the kernel labels from the text labels to generate them. The gap label g_i can be described as:

$$G_g^i = \begin{cases} 1, & \text{if } i \in \bigcup\limits_j \left(G_t^j - G_k^j\right) \\ 0, & \text{otherwise} \end{cases}, \tag{21}$$

where G_t, and G_k are the text regions and the text kernel regions, respectively.

F. Optimization

The losses of the three subtasks make up the total loss $\mathcal{L}$ of CIC, which are text segmentation loss $\mathcal{L}_t$, kernel segmentation loss $\mathcal{L}_k$, and gap prediction loss $\mathcal{L}_g$. It can be described as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_k + \lambda_2 \mathcal{L}_t + \lambda_3 \mathcal{L}_g + \lambda_4 \mathcal{L}_{cv}, \tag{22}$$

where λ_1, λ_2, λ_3, and λ_4 are hyper-parameters that are used to balance the loss weights of multitasking, respectively.

The BCE loss $\mathcal{L}_{bce}$ is applied to both $\mathcal{L}_k$, $\mathcal{L}_t$, and $\mathcal{L}_g$. Inspired by [7], to alleviate the imbalance between positive and negative samples, we adopt hard negative mining:

$$\mathcal{L}_{bce} = \sum_{p \in S} -p_{gt} * log(p_{pre}) - (1 - p_{gt}) * log(1 - p_{pre}), \tag{23}$$

where p_{pre} and p_{gt} are the prediction and ground-truth, respectively. Therefore, $\mathcal{L}_t$, $\mathcal{L}_k$, and $\mathcal{L}_g$ can be defined as:

$$\mathcal{L}_t = \mathcal{L}_{bce}(P_t, G_t), \tag{24}$$

$$\mathcal{L}_k = \mathcal{L}_{bce}(P_k, G_k), \tag{25}$$

$$\mathcal{L}_g = \mathcal{L}_{bce}(P_g, G_g), \tag{26}$$

where P_t and G_t represent the ground truth and prediction of the text. P_k and G_k are the ground truth and prediction of the text kernel. P_g and G_g are the predicted gap (computed by the Eq. 5) and groud-truth (computed by the Eq. 21).

G. Inference Stage

The text instances can be reconstructed by only predicting the text kernel without the need to predict the text, which dramatically improves the model's inference speed. The post-processing part consists of three main processes: (1) Using a fixed value as the threshold, binarize the prediction map of the text kernel area to obtain a binarized image. (2) Obtain the connection area for the binarized image through morphological processing. (3) Expand the connected area by a specific offset D^* to obtain the final result. The specific calculation formula of D^* is as follows:

$$D^*{}_i = \frac{A_i^* \times (1 - (r^*)^2)}{P_i^*}, \tag{27}$$

where D_i^* represents the offset that the ith shrunk polygon needs to expand. A_i^* ,P_i^* represent the perimeter of the ith shrunk polygon. r^* represents the scale factor in the experiment, which is set to 1.5.

IV. Experiments

A. Datasets

MSRA-TD500 [45] is a multi-orientation dataset, which contains English and Chinese. Each instance is labeled at line level. It includes 300 images for training and 200 images for testing. Following the previous works [6], [7], [35], [37], we introduce the 400 images from HUST-TR400 [46] to supplement the training dataset.

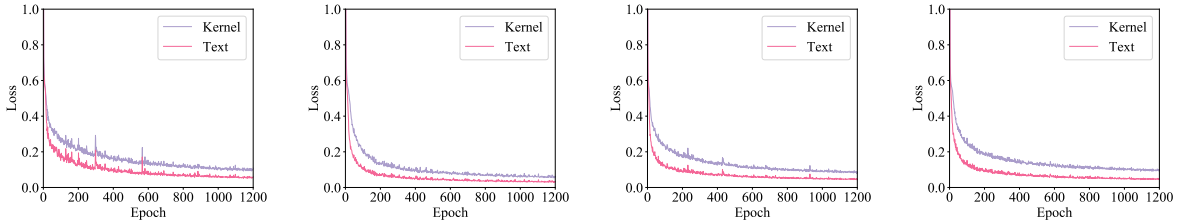
CTW1500 [47] dataset is a scene text dataset with challenges for long curved text detection. It contains 1,500 pictures, 1,000 for training, and 500 for testing. Unlike other datasets (such as ICDAR2015, MSRA-TD500), each instance is annotated with 14 points to represent arbitrary-shaped texts.

ICDAR2015 [48] dataset is usually used for omnidirectional scene text detection. Its training set includes 1,000 images, and the testing set contains 500. Each text instance is described in it with four points.

Total-Text [49] contains all kinds of texts, such as horizontal, multi-directional, and curved texts. It includes 1,255 pictures for training and 300 for testing.

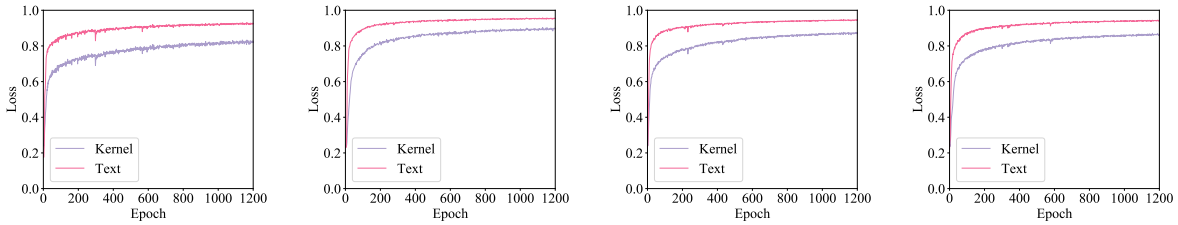
SynthText150k [19] is a synthesized text dataset that contains 150,000 images from different scenes. It is utilized for pre-training to improve the robustness of the model.

ASAYAR-TXT is a subset dataset of ASAYAR [50], whose images are from Moroccan highways. Each instance has word-level and line-level annotations. It includes 1,100 images for training and 275 images for testing.



(a) ICDAR2015 (b) MSRA-TD500 (c) Total-Text (d) CTW1500

Fig. 9. The loss of kernel and text map on the ICDAR2015, MSRA-TD500, Total-Text, and CTW1500, respectively.



(a) ICDAR2015 (b) MSRA-TD500 (c) Total-Text (d) CTW1500

Fig. 10. The IOU of kernel and text map on the ICDAR2015, MSRA-TD500, Total-Text, and CTW1500, respectively.

COCO-Text [51] is a large-scale dataset, which contains 43,686 training images, 10,000 validation images, and 10,000 test images.

B. Evaluation Metrics

To achieve a fair comparison between different methods, we use precision, recall, F-measure, and FPS to evaluate the results, which are commonly used. In the comparative analysis of the results, we use F-measure and FPS to represent the performance and speed of the model, respectively. TP, FP, and FN describe the predicted quantity of true positives, false positives, and false negatives, respectively. The precision (P), recall (R), and F-measure (F) can be expressed as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, \tag{28}$$

$$F = \frac{2 \times P \times R}{P + R}. \tag{29}$$

C. Implementation Details

ResNet-18 [41] with deformable convolution [42] [43] is used as the model backbone. We pre-train on the SynthText150k for 20 epochs, respectively. Then the model is trained for 1,200 epochs on other datasets to finetune the model. The batch size of the entire training is selected as 16. For the ResNet18, SGD is applied for gradient updating and the learning rate is initialized to 0.007. The weight decay and momentum during training are set to 0.0001 and 0.9, respectively. For the ResNet50, AdamW [52] is adopted and the learning rate is initialized to 0.0001. The weight decay is set to 0.0001. For the CTW1500 dataset, using ResNet18 as the backbone results in 12.23M parameters and a computational complexity of 46.22 GFLOPs. Using ResNet50 as the backbone increases the parameter count to 26.91M, with a computational cost of 94.71 GFLOPs. The learning rate is adjusted by the "poly" [53] method. For data augmentation, random cropping and slight rotation are used. We follow DBNet [7] to validate the detection results. In the multi-task training, the parameters λ_1, λ_2, λ_3, and λ_4 are set to 6, 3, 1, and 10, respectively. The inference speed is tested on a single GTX 1080Ti with an i7-6800K CPU.

TABLE I
THE COMPARISON OF DETECTION RESULTS UNDER DIFFERENT CONDITION SETTINGS ON THE MSRA-TD500 DATASET, WHERE "EXT." REPRESENTS USING EXTRA DATA TO PRE-TRAIN.

BITM	DKIM	CVM	Ext.	Precision	Recall	F-measure
×	×	×	×	82.3	76.6	79.4
✓	×	×	×	85.8	79.9	82.7
×	✓	×	×	84.8	78.7	81.6
×	×	✓	×	86.1	80.1	83.0
✓	×	✓	×	87.8	79.0	83.2
✓	✓	×	×	86.3	81.3	83.7
✓	✓	✓	×	86.7	81.6	84.1
✓	✓	✓	✓	91.6	86.3	88.8

TABLE II
THE DETECTION RESULTS ON THE MSRA-TD500 DATASET WITH DIFFERENT α AND β.

	α	β	Precision	Recall	F-measure
CIC with	11	6	84.5	79.4	81.8
CIC with	16	9	84.7	79.0	81.8
CIC with	20	11	85.8	79.9	**82.7**

D. Ablation Study

The ablation study is conducted on the MSRA-TD500 dataset to show the validity of the proposed methods.

The comparison of the difficulty of learning texts and text kernels. We employ BCE loss for ablation experiments to verify the learning difficulty of text regions and text kernel regions. The weights of kernel loss and text loss are set to 1:1.

TABLE III
THE DETECTION RESULTS ON THE MSRA-TD500 DATASET UNDER DIFFERENT η.

	η	Precision	Recall	F-measure
CIC with	1	84.8	80.8	82.7
CIC with	10	86.1	80.1	**83.0**
CIC with	100	84.0	81.4	82.7

TABLE IV
THE DETECTION RESULTS ON THE MSRA-TD500 DATASET UNDER TWO DIFFERENT SCHEMES.

Concat	Add	Precision	Recall	F-measure
✓		86.7	81.6	**83.7**
	✓	86.6	80.2	**83.3**

As shown in Fig. 9 and Fig. 10, the loss of the text kernel is continuously much higher than that of the text region, which also reflects that the text kernel region is more challenging to recognize than the text region for the model. For the IOU, the text kernel is continuously much lower than that of the text region. This also verifies the feasibility of text is easier to learn than kernel. The Loss and IOU in Fig. 9 and Fig. 10 are obtained by separately predicting the text region and the kernel region within the same experiment.

The influence of BITM. As shown in Table I, compared to the baseline, the BITM improves the precision, recall, and F-measure by 3.5%, 3.3%, and 3.3%, respectively. It generates collaborative information to help the model locate kernels under the guidance of the complete semantic features of texts and to distinguish the differences between them for recognizing kernels from texts accurately. Extensive experimental results powerfully demonstrate the importance of the collaborative progress between the kernel and text and the superiority of the proposed BITM.

The effectiveness of DKIM. As shown in Table I, the baseline with DKIM outperforms the baseline by 2.5%, 2.1%, and 2.2% in precision, recall, and F-measure, respectively. In addition, on the basis of equipping BITM, it brings 1.0% in terms of F-measure. The above experiments demonstrate the fusion of text and kernel features helps the model detect text kernel accurately.

Importances of CVM. It is found in Table I, that the CVM brings 3.8%, 3.5%, and 3.6% improvements compared to the baseline in precision, recall, and F-measure, respectively. When equipping BITM and CKFM, it brings 0.4% improvements in terms of F-measure. The above experiments verify the superiority of the proposed CVM.

Impacts of different settings for BITM. Table II displays the results of our approach without SWM for different values of α and β. As shown in Fig. 5, when they take too small a value, they cause excessive systematic errors in DAG at $g = 0$ and $g = 1$. When α and β set too large values, DAG in $g \in (0, 1)$ fitting error is large. After experimental verification, the best results can be achieved when α and β are set as 20 and 11, respectively. These values are also set in the following experiments for accurate detection results.

Influence of different settings for DKIM. We design "Add" and "Concat" schemes to perform experiments. As listed in Table III, the latter outperforms the former 0.4% in F-measure and is adopted in the following experiments.

Impacts of different settings for CVM. As we can see from Table IV, η has little effect on the results of the experiment. When the η is set to 10, the proposed model achieves the best performance, 86.1%, 80.1%, and 83.0% in precision, recall, and F-measure, respectively. When the η is changed to 10, the performance is dropped by 0.3% in terms of F-measure.

E. Comparisons with Previous Methods

The comparison with current advanced methods is built on five public datasets from different scenes. ICDAR2015 and MSRA-TD500 are used to detect multi-directional text and long text. CTW1500 and Total-Text datasets are used for curved text detection. In addition, the ASAYAR-TXT dataset is used for traffic text detection.

Long multi-directional text detection. MSRA-TD500 has line-level annotations and has both Chinese and English. We follow DBNet [7] to resize the short side of the input image to 736. As can be seen in Table V, the proposed method not only obtains the best performance but also enjoys a high inference speed. Compared to the state-of-the-art DBNet++ [56], our approach improves the F-measure by 3.7% with ResNet18 and ResNet50, respectively. Additionally, the CIC outperforms MixNet [63], TPPAN [3], and CT-Net [71]. The proposed method even outperforms the majority of current approaches that use large networks when the backbone is ResNet-18. This proves that BITM has an excellent performance advantage when facing long texts, which can also be verified on the CTW1500 dataset.

Curved text detection. CTW1500 and Total-Text contain large quantities of curved text. The proposed approach is validated on them to demonstrate its robustness for arbitrary-shaped text. For TotalText and CTW1500, the short side of the input image is set to 800 and 640, respectively. Our approach achieves excellent performance and speed in Table V. Compared with DBNet++-ResNet18 [56], CIC-ResNet18 improves the F-measure by 2.8% on CTW1500. It outperforms the existing SOTA real-time methods significantly, whose improvement is more than 2%. For the ResNet50, it achieves 88.9%, 86.2%, and 87.5% in terms of precision, recall, and F-measure, respectively. Although it is slower than SRFormer [67], it is still superior to most existing methods. Compared to DBNet++, the F-measure improvement is 1.8%. Similar to the detection results of MSRA-TD500, the proposed CIC outperforms the majority of approaches using large networks when the backbone adopts ResNet-18. Our approach still achieves excellent performance in the real-time methods on Total-Text. The proposed method achieves 88.8%, 84.6%, and 86.6% in precision, recall, and F-measure when adopting ResNet18, which outperforms both existing SOTA real-time methods. Overall, even our method adopts ResNet18 as the backbone, which is also superior to existing SOTA methods that adopt ResNet50. When equipped with ResNet50, although it is slighter lower than TPPAN [3], it exceeds the most existing methods LeafText [26] and ADNet [8] by 0.5% and 0.4% in F-measure.

TABLE V
COMPARISON WITH EXISTING STATE-OF-THE-ART (SOTA) APPROACHES ON THE CTW1500,TOTAL-TEXT, AND MSRA-TD500 DATASETS. "**BLUE**", AND "**RED**" REPRESENT THE OPTIMAL PERFORMANCE FOR THE REAL-TIME AND NO-REAL-TIME METHODS, RESPECTIVELY. "-" REPRESENTS THE CORRESPONDING RESULTS THAT ARE NOT REPORTED IN THE PAPER. "RT" AND "NRT" REPRESENT REAL-TIME AND NO-REAL-TIME METHODS, RESPECTIVELY.

Type	Methods	Venue	Back.	CTW1500				TotalText				MSRA-TD500			
				P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
RT	DBNet [7]	AAAI'20	Res18	84.8	77.5	81.0	55	88.3	77.9	82.8	50	90.4	76.3	82.8	62
	CT [4]	NeurIPS'21	Res18	88.3	79.9	83.9	40.8	90.5	82.5	86.3	40.0	90.0	82.5	86.1	34.8
	PAN++ [54]	TPAMI'22	Res18	87.1	81.1	84.0	36.0	89.9	81.0	85.3	38.3	85.3	84.0	84.7	32.5
	CM-Net [34]	TMM'22	Res18	86.0	82.2	84.1	50.3	88.5	81.4	84.8	49.8	89.9	80.6	85.0	41.7
	HFENet [55]	TITS'23	Res18	85.1	81.2	83.1	32.2	85.7	81.7	83.7	22.0	89.7	81.1	85.2	40.9
	ZTD [36]	TNNLS'23	Res18	88.4	80.2	84.1	76.9	90.1	82.3	86.0	75.2	91.6	82.4	86.8	59.2
	FS [38]	TIP'23	Res18	84.6	77.7	81.0	35.2	85.8	77.0	81.1	33.5	90.0	80.4	84.9	35.5
	RSMTD [37]	TMM'23	Res18	87.8	80.3	83.9	72.1	88.5	83.8	86.1	70.9	89.8	83.1	86.3	62.5
	DBNet++ [56]	TPAMI'23	Res18	86.7	81.3	83.9	40	87.4	79.6	83.3	48	87.9	82.5	85.1	55
	CIC	**Ours**	Res18	88.2	85.0	**86.5**	50.1	88.8	84.6	**86.6**	40.0	91.6	86.3	**88.8**	40.9
NRT	OPMP [57]	TMM'21	Res50	85.1	80.8	82.9	1.4	87.6	82.7	85.1	1.4	86.0	83.4	84.7	1.6
	FCE [17]	CVPR'21	Res50	87.6	83.4	85.5	-	89.3	82.5	85.8	-	-	-	-	-
	DText [58]	PR'22	Res50	86.9	82.7	84.7	-	90.5	82.7	86.4	-	87.9	83.1	85.4	-
	I3CL [59]	IJCV'22	Res50	88.4	84.6	86.5	-	89.8	84.2	86.9	-	-	-	-	-
	NASK [1]	TCSVT'22	Res50	83.4	80.1	81.7	12.1	85.6	83.2	84.4	8.4	-	-	-	-
	LEMNet [60]	TMM'22	Res50	86.6	83.8	85.2	-	89.9	85.4	87.6	-	85.6	84.8	85.2	-
	HFENet [55]	TITS'23	Res50	88.1	83.4	85.7	18.1	89.0	84.0	86.4	12.2	92.8	84.0	88.2	21.4
	TextDCT [61]	TMM'23	Res50	85.0	85.3	85.1	17.2	87.2	82.7	84.9	15.1	88.9	86.8	87.5	17.2
	DBNet++ [56]	TPAMI'23	Res50	87.9	82.8	85.3	26	88.9	83.2	86.0	28	91.5	83.3	87.2	29
	RP-Text [62]	TMM'23	Res18	87.8	81.6	84.7	-	89.4	82.8	86.0	-	88.4	84.6	86.5	-
	KPN [25]	TNNLS'23	Res50	84.4	84.2	84.3	16.3	88.7	85.6	87.1	15.0	-	-	-	-
	FS [38]	TIP'23	Res50	85.3	82.5	83.9	25.1	88.7	79.9	84.1	24.3	89.3	81.6	85.3	25.4
	MixNet [63]	arXiv'23	FSNet_M	-	-	-	-	93.0	88.1	90.5	15.2	90.7	88.1	89.4	-
	MorphText [64]	TMM'23	Res50	90.0	83.3	86.5	-	90.6	5.2	87.8	-	90.7	83.5	87.0	-
	ASSTD [65]	TMM'23	VGG16	89.8	83.3	86.4	-	89.4	85.8	87.6	-	90.5	83.8	87.0	-
	LeafText [26]	TMM'23	Res18	87.1	83.9	85.5	-	90.8	84.0	87.3	-	92.1	83.8	87.8	-
	ADNet [8]	TMM'23	Res50	88.2	83.1	85.6	-	90.6	84.4	87.4	-	92.0	83.2	87.4	-
	STD [66]	TMM'24	Res50	88.5	84.9	86.7	12.1	90.7	83.9	87.2	12.1	92.8	86.9	89.8	13.4
	TPPAN [3]	TCSVT'24	Res50	88.7	86.3	87.5	-	91.2	85.0	88.0	-	93.4	88.2	90.7	-
	SRFormer [67]	AAAI'24	Res50	89.4	89.8	**89.6**	-	91.5	87.9	**89.7**	-	-	-	-	-
	FEPE [68]	TMM'24	Res50	88.8	83.5	86.0	22	91.3	81.9	86.4	32	90.5	85.4	88.0	32
	CBNet [69]	IJCV'24	Res18	89.0	81.9	86.0	-	90.1	82.5	86.1	-	91.1	84.8	87.8	-
	TTDNet [70]	TITS'24	Res50	-	-	-	-	87.4	82.2	84.7	-	90.4	83.9	87.0	-
	CT-Net [71]	TCSVT'24	Res50	88.5	83.8	86.1	11.2	90.8	85.0	87.8	10.1	90.8	84.4	87.5	11.6
	CIC	**Ours**	Res50	88.9	86.2	87.5	22.9	90.2	85.5	87.8	16.7	95.1	87.1	**90.9**	17.1

TABLE VI
THE COMPARISON WITH SOTA METHODS ON THE ICDAR2015 DATASET. THE BEST PERFORMANCE IS LABELED IN **BOLD**.

Methods	Backbone	P	R	F	FPS
DB [7]	ResNet18	86.8	78.4	82.3	48
PAN++ [54]	ResNet18	85.9	80.4	83.1	28.2
CM-Net [34]	ResNet18	86.7	81.3	83.9	34.5
ZTD [36]	ResNet18	87.5	79.0	83.0	48.3
DB++ [56]	ResNet18	90.1	77.2	83.1	44
CIC	ResNet18	89.0	83.7	**86.3**	30.8
DText [58]	ResNet50	88.5	85.6	87.0	-
LEMNet [60]	ResNet50	88.3	85.9	87.1	-
FS [38]	ResNet50	89.8	82.7	86.1	12.1
KPN [25]	ResNet50	88.3	84.8	87.4	6.3
LeafText [26]	ResNet50	88.9	82.9	86.1	-
DB++ [56]	ResNet50	90.9	83.9	87.3	10
ADNet [8]	ResNet50	92.5	83.7	**87.9**	-
VTD [72]	ResNet50	88.5	85.8	87.1	-
TTDNet [72]	ResNet50	90.0	85.6	87.7	-
CIC	ResNet50	91.8	83.5	87.5	5.7

Omnidirectional text detection. ICDAR2015 is a multi-directional text dataset containing many small and low-resolution texts. When using ResNet18 and ResNet50 as the backbone, the short side of the image is resized to 736 and 1152, respectively. As shown in Table VI, compared with the existing SOTA method DBNet++ [56], the proposed method is superior to it in performance when adopting ResNet18 or

TABLE VII
THE COMPARISON WITH EXISTING STATE-OF-THE-ART METHODS ON THE COCO-TEXT TEXT SET. THE BEST PERFORMANCE IS LABELED IN **BOLD**.

Methods	Backbone	Precision	Recall	F-measure
Texboxes++ [10]	VGG-16	60.9	56.7	58.7
Lyu *et al.* [73]	VGG-16	72.5	52.9	61.1
MaskSpotter [74]	Res50	66.8	58.3	62.3
Boundary [75]	Res50	59.0	67.7	63.0
Feng *et al.* [76]	Res50	66.8	59.4	62.9
FC^{2}RN [77]	Res50	68.5	58.2	63.0
CBN [69]	Res50	66.8	62.1	64.4
CIC	Res18	69.1	61.1	**64.9**

TABLE VIII
THE COMPARISON WITH EXISTING STATE-OF-THE-ART METHODS ON THE ASAYAR-TXT. THE BEST PERFORMANCE IS LABELED IN **BOLD**.

Methods	Precision	Recall	F-measure	FPS
Texboxes++ [10]	66	52	58	-
CTPN [78]	80	95	86	-
CTPN+Baseline [78]	83	97	89	-
HFENet [55]	96.3	97.2	96.8	-
CIC-Res18	97.9	97.5	**97.7**	30.5

Fig. 11. The visualization of the proposed CIC on Totaltext, CTW1500, ICDAR2015, MSRA-TD500, and ASAYAR-TXT, respectively.

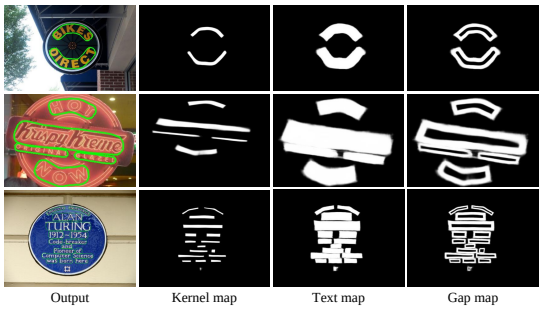


Fig. 12. The visualization of the output, kernel map, text map, and gap map.

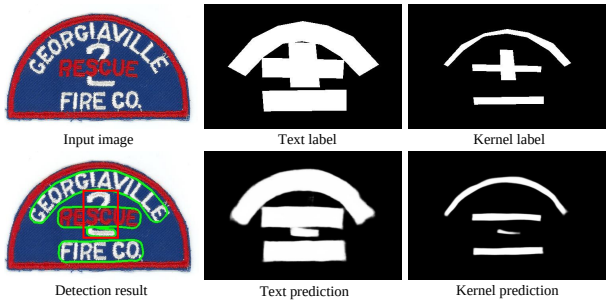


Fig. 13. The visualization of two highly overlapping instances.

ResNet50 as the backbone. Although the proposed method is slighter lower than ADNet, it is still superior to most existing methods. Compared with VTD [72] and LeafText [26], our approach improves the F-measure by 0.4% and 1.4%. The proposed method has sufficient advantages in terms of performance and speed in comparison to real-time methods. In addition, our CIC achieves the best performance among real-time methods. For the COCO-Text dataset, the proposed method achieves precision, recall, and F-measure of 69.1%, 64.1%, and 64.9%, respectively, outperforming existing state-of-the-art methods.

Traffic text detection. ASAYAR-TXT is a traffic text dataset, which includes different weather images. Following the HFENet [55], the short size of the image is resized to 736 during the inference stage. As we can see from Table VIII, compared to the existing SOTA method HFENet, our CIC outperforms by 1.6%, 0.3%, and 0.9% in precision, recall, and F-measure, respectively. The above experiments verify the superiority of detecting traffic texts.

F. Visualization

To exhibit the generality of the proposed CIC, the detection results from different datasets (TotalText, CTW1500, ICDAR2015, MSRA-TD500, and ASAYAR-TXT) are visualized in Fig. 11, which contains curved texts and multi-oriented texts. To further show the superiority of the proposed CIC, we illustrate the predictions of the kernel map, text map, gap map, and output in Fig. 12. Additionally, we present an image with two highly overlapping instances in Fig. 13. In this case, the generated kernel labels fail to provide sufficient information for the model to distinguish highly overlapping instances, which is a limitation of the segmentation method. Notably, such situations are rare in real-world scenarios. As shown in Fig. 14, the proposed method demonstrates a clear advantage in separating text instances compared to existing methods (TextPMs [79], DBNet++ [56], and KPN [25]). The

Fig. 14. The visual comparison with existing state-of-the-art methods. The incorrect results are labeled in red.

TABLE IX
THE CROSS-DATASET VALIDATION RESULTS ARE SHOWN ON LINE-LEVEL AND WORD-LEVEL ANNOTATED DATASETS.

Type	Train	Test	Method	P	R	F
Word	IC15	Total	TextField [24]	61.5	65.2	63.3
			CM-Net [34]	75.8	64.5	69.7
			ZTD [36]	78.5	64.1	70.6
			CIC(ours)	76.5	72.7	**74.6**
	Total	IC15	TextField [24]	77.1	66.0	71.1
			CM-Net [34]	76.5	68.1	72.1
			ZTD [36]	79.8	69.3	74.2
			CIC(ours)	78.9	77.5	**78.2**
Line	MSRA	CTW	TextField [24]	75.3	70.0	72.6
			CM-Net [34]	77.2	69.7	72.8
			ZTD [36]	84.1	73.4	78.4
			CIC(ours)	85.5	75.4	**80.2**
	CTW	MSRA	TextField [24]	85.3	75.8	80.3
			CM-Net [34]	85.8	77.1	81.2
			ZTD [36]	86.8	77.9	82.1
			CIC(ours)	85.6	84.0	**84.8**

above visualization demonstrates further the superiority of the proposed CIC.

G. Generalizability Validation of CIC

Inspired by CM-Net [34], we perform four groups of cross-validation experiments to validate the robustness of our approach and compare the detection results with CM-Net in Table IX. For the line-level annotated dataset, we choose MSRA-TD500 and CTW1500, the former containing mainly multi-directional text and the latter containing curved text. We train on CTW1500 to test the MSRA-TD500 dataset and achieve an F-measure of 84.8%, a 3.6% improvement compared to CM-Net. This result is still superior to some existing SOTA methods, which are trained directly on the MSRA-TD500, such as DBNet [7], PAN [35], and PAN++ [54]. After swapping the positions of the two datasets, the proposed approach achieves an F-measure of 80.2%, which is 7.4% higher than CM-Net. It demonstrates the excellent robustness of our model in extracting line-level features and the feasibility of using it on other line-level labeled data. For the word-level annotated dataset, we choose ICDAR2015 and Total-Text, the former containing multi-directional text and the latter including curved text. We have done similar experiments on the above datasets, and the results are also competitive. This shows that the proposed model has acceptable generalization for word-level data as well.

In summary, the proposed CIC has strong enough generalization in the face of different scenario texts. We train the model on curved texts and test on multi-directional texts and still achieve competitive results, indicating that our approach is sufficiently robust to text shapes.

V. CONCLUSION

We propose an effective scene text detector to deal with arbitrary-shaped texts. A bidirectional information transfer module (BITM) is designed to generate the collaborative information between the predicted text and kernel via the proposed differentiable adaptive gap operator. With the help of complete semantic features of texts, this information helps to locate kernels and recognize kernels from texts accurately. Moreover, we propose a dual knowledge integration module (DKIM), which encourages text and kernel information to fuse at the feature perspective. In addition, a cross-verification module (CVM) is proposed, which leverages text prediction to suppress the false positive prediction of the kernel. Our approach achieves SOTA results on multiple public benchmarks from different scenes, especially line-level labeled texts. In addition, benefiting that BITM and CVM can be removed in the inference phase, the proposed approach achieves real-time inference speed while ensuring accurate results. Nevertheless, our approach still has some limitations. For example, it cannot handle the case where two text instances overlap, an issue that we will address in future work.

REFERENCES

[1] M. Cao, C. Zhang, D. Yang, and Y. Zou, "All you need is a second look: Towards arbitrary-shaped text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 758–767, 2022.

[2] Y. Cai, C. Liu, P. Cheng, D. Du, L. Zhang, W. Wang, and Q. Ye, "Scale-residual learning network for scene text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2725–2738, 2021.

[3] J. Xu, A. Lin, J. Li, and G. Lu, "Text position-aware pixel aggregation network with adaptive gaussian threshold: Detecting text in the wild," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 1, pp. 286–298, 2024.

[4] T. Sheng, J. Chen, and Z. Lian, "Centripetaltext: An efficient text instance representation for scene text detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 335–346, 2021.

[5] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.

[6] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9336–9345.

[7] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[8] Y. Qu, H. Xie, S. Fang, Y. Wang, and Y. Zhang, "Adnet: Rethinking the shrunk polygon-based approach in scene text detection," *IEEE Transactions on Multimedia*, pp. 1–14, 2022.

[9] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[10] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE transactions on image processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

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

[13] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[15] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3047–3055.

[16] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. C-20, no. 5, pp. 562–569, 1971.

[17] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3123–3131.

[18] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. Association for Computing Machinery, 2020, p. 111–119.

[19] 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 *proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 9809–9818.

[20] P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 7393–7402.

[21] J. Tang, W. Zhang, H. Liu, M. Yang, B. Jiang, G. Hu, and X. Bai, "Few could be better than all: Feature sampling and grouping for scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4563–4572.

[22] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[23] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[24] 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.

[25] S. Zhang, X. Zhu, J. Hou, C. Yang, and X. Yin, "Kernel proposal network for arbitrary shape text detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[26] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Text growing on leaf," *IEEE Transactions on Multimedia*, vol. 25, pp. 9029–9043, 2023.

[27] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365–9374.

[28] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4234–4243.

[29] W. Yu, Y. Liu, W. Hua, D. Jiang, B. Ren, and X. Bai, "Turning a clip model into a scene text detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 6978–6988.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[31] Z. Chen, "Arbitrary shape text detection with discrete cosine transform and clip for urban scene perception in its," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2025.

[32] H. Guo, T. Wang, J. Yun, and J. Zhao, "Multilingual natural scene text detection via global feature fusion," *Applied Intelligence*, vol. 55, no. 1, pp. 1–16, 2025.

[33] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[34] C. Yang, M. Chen, Z. Xiong, Y. Yuan, and Q. Wang, "Cm-net: Concentric mask based arbitrary-shaped text detection," *IEEE Transactions on Image Processing*, pp. 2864–2877, 2022.

[35] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8440–8449.

[36] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Zoom text detector," *IEEE Trans. Neural Networks and Learning Systems*, 2023, early Access, doi: 10.1109/TNNLS.2023.3289327.

[37] ——, "Reinforcement shrink-mask for text detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 6458–6470, 2023.

[38] F. Wang, X. Xu, Y. Chen, and X. Li, "Fuzzy semantics for arbitrary-shaped scene text detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 1–12, 2023.

[39] S. Zhang, X. Zhu, C. Yang, H. Wang, and X. Yin, "Adaptive boundary proposal network for arbitrary shape text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 1305–1314.

[40] S.-X. Zhang, C. Yang, X. Zhu, and X.-C. Yin, "Arbitrary shape text detection via boundary transformer," *IEEE Transactions on Multimedia*, vol. 26, pp. 1747–1760, 2024.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[43] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 9308–9316.

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

[45] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1083–1090.

[46] 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.
[47] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.
[48] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition*. IEEE, 2015, pp. 1156–1160.
[49] C. Ch'ng and C. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *2017 14th IAPR international conference on document analysis and recognition*, vol. 1. IEEE, 2017, pp. 935–942.
[50] M. Akallouch, K. S. Boujemaa, A. Bouhoute, K. Fardousse, and I. Berrada, "Asayar: A dataset for arabic-latin scene text localization in highway traffic panels," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3026–3036, 2022.
[51] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *arXiv preprint arXiv:1601.07140*, 2016.
[52] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.
[53] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision*, 2018, pp. 325–341.
[54] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Z. Yang, T. Lu, and C. Shen, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5349–5367, 2022.
[55] M. Liang, X. Zhu, H. Zhou, J. Qin, and X.-C. Yin, "Hfenet: Hybrid feature enhancement network for detecting texts in scenes and traffic panels," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 200–14 212, 2023.
[56] 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*, vol. 45, no. 1, pp. 919–931, 2023.
[57] 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, 2021.
[58] Y. Cai, Y. Liu, C. Shen, L. Jin, Y. Li, and D. Ergu, "Arbitrarily shaped scene text detection with dynamic convolution," *Pattern Recognition*, vol. 127, p. 108608, 2022.
[59] B. Du, J. Ye, J. Zhang, J. Liu, and D. Tao, "I3cl: intra-and inter-instance collaborative learning for arbitrary-shaped scene text detection," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1961–1977, 2022.
[60] M. Xing, H. Xie, Q. Tan, S. Fang, Y. Wang, Z. Zha, and Y. Zhang, "Boundary-aware arbitrary-shaped scene text detector with learnable embedding network," *IEEE Transactions on Multimedia*, vol. 24, pp. 3129–3143, 2022.
[61] Y. Su, Z. Shao, Y. Zhou, F. Meng, H. Zhu, B. Liu, and R. Yao, "Textdct: Arbitrary-shaped text detection via discrete cosine transform mask," *IEEE Transactions on Multimedia*, vol. 25, pp. 5030–5042, 2023.
[62] Q. Wang, B. Fu, M. Li, J. He, X. Peng, and Y. Qiao, "Region-aware arbitrary-shaped text detection with progressive fusion," *IEEE Transactions on Multimedia*, vol. 25, pp. 4718–4729, 2023.
[63] Y.-X. Zeng, J.-W. Hsieh, X. Li, and M.-C. Chang, "Mixnet: toward accurate detection of challenging scene text in the wild," *arXiv preprint arXiv:2308.12817*, 2023.
[64] C. Xu, W. Jia, R. Wang, X. Luo, and X. He, "Morphtext: Deep morphology regularized accurate arbitrary-shape scene text detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 4199–4212, 2023.
[65] C. Xu, W. Jia, T. Cui, R. Wang, Y.-f. Zhang, and X. He, "Arbitrary-shape scene text detection via visual-relational rectification and contour approximation," *IEEE Transactions on Multimedia*, vol. 25, pp. 4052–4066, 2023.
[66] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Spotlight text detector: Spotlight on candidate regions like a camera," *IEEE Transactions on Multimedia*, pp. 1–14, 2024.
[67] Q. Bu, S. Park, M. Khang, and Y. Cheng, "Srformer: Text detection transformer with incorporated segmentation and regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 2, 2024, pp. 855–863.
[68] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Focus entirety and perceive environment for arbitrary-shaped text detection," *IEEE Transactions on Multimedia*, vol. 27, pp. 287–299, 2025.
[69] X. Zhao, W. Feng, Z. Zhang, J. Lv, X. Zhu, Z. Lin, J. Hu, and J. Shao, "Cbnet: A plug-and-play network for segmentation-based scene text detection," *International Journal of Computer Vision*, pp. 1–20, 2024.
[70] R. Wang, Y. Zhu, H. Chen, Z. Zhu, X. Zhang, Y. Ding, S. Qian, C. Gao, L. Liu, and N. Sang, "Ttdnet: An end-to-end traffic text detection framework for open driving environments," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2024.
[71] Z. Shao, Y. Su, Y. Zhou, F. Meng, H. Zhu, B. Liu, and R. Yao, "Ct-net: Arbitrary-shaped text detection via contour transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 3, pp. 1815–1826, 2024.
[72] J.-B. Zhang, W. Feng, M.-B. Zhao, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Video text detection with robust feature representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 6, pp. 4407–4420, 2024.
[73] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7553–7563.
[74] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 67–83.
[75] H. Wang, P. Lu, H. Zhang, M. Yang, X. Bai, Y. Xu, M. He, Y. Wang, and W. Liu, "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 160–12 167.
[76] W. Feng, F. Yin, X.-Y. Zhang, W. He, and C.-L. Liu, "Residual dual scale scene text spotting by fusing bottom-up and top-down processing," *International Journal of Computer Vision*, vol. 129, pp. 619–637, 2021.
[77] X. Qin, Y. Zhou, Y. Guo, D. Wu, and W. Wang, "Fc^{2}rn: a fully convolutional corner refinement network for accurate multi-oriented scene text detection," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4350–4354.
[78] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*. Springer, 2016, pp. 56–72.
[79] S.-X. Zhang, X. Zhu, L. Chen, J.-B. Hou, and X.-C. Yin, "Arbitrary shape text detection via segmentation with probability maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2736–2750, 2023.

Xu Han received the B.E. degree in information and computing sciences from Northeast Agricultural University, Harbin, China, in 2021

He is currently pursuing the Ph.D. degree with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN). His research interests include computer vision, pattern recognition and text detection.

Qi Wang (Senior Member, IEEE) 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. For more information, visit the link (https://crabwq.github.io/).