Balancing Optimization Strategies and Practical Goals: An Efficient Scene Text Detector

Xu Han, Chuang Yang, Junyu Gao, *Member, IEEE,* and Qi Wang, *Senior Member, IEEE*

***Abstract*—Scene text reading is a crucial task for scene understanding. Text detection, as a fundamental task in scene text reading, has recently garnered significant attention. Among various approaches, segmentation-based methods stand out for their flexible pixel-level prediction capabilities. However, two main issues remain. 1) These methods treat all text instances as a pixel set during training, causing the features of large-scale instances to dominate the model optimization process. As a result, the optimization deviates from the instance-level objectives. 2) Segmentation methods filter candidates based on pixel-level class scores, whereas what is needed is an evaluation of whether an instance is text, which also deviates from the original goals. To address these issues, we propose an Instance-Equal Feature Guide Module (IEFGM), a Cross-Level Feature Interaction Module (CLIFM), and a Pixel-Instance Fusion Discriminator (PIFD) to balance optimization strategies with practical goals. The IEFGM introduces instance-level features and positional information, guiding the model to treat instances of different scales equally at the feature level. The CLIFM encourages feature interaction across different levels, enabling the model to recognize text from various perspectives. Unlike existing methods that filter candidates using pixel-level results, the PIFD integrates both instance-level and pixel-level information to identify candidate regions, aligning with the original goals of text detection. A series of ablation studies demonstrates the effectiveness of the proposed modules. Extensive experiments across six datasets from different scenes demonstrate that our method outperforms existing state-of-the-art approaches.**

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

I. Introduction

SCENE text reading helps intelligent devices comprehend deep semantic scene information and accomplish many applications (such as automatic driving, image retrieval, unmanned systems, etc.), significantly improving production efficiency. As its fundamental task, scene text detection is essential for reading, which has attracted increased researchers and has become a hotspot in computer vision.

The advancement of deep learning has greatly impacted the field of scene text detection [1], [2], [3]. However, two main challenges impede progress: irregular shapes and adhesion between adjacent instances. The shrink-based method stands out among various approaches due to its flexible pixel-level predictions, distinct instance separation, and simple post-processing. Based on this, DBNet [4] proposes a differentiable binarization to introduce the binarization process to the network training. DBNet++ [5] proposes an adaptive scale fusion module that enhances the scale robustness of the module. PAN [6], RSMTD [3], LeafText [1], and CT [7] improve the model from a text representation perspective. Although the aforementioned methods use various strategies to improve detection performance, they overlook two key issues: the misalignment of the optimization direction and evaluation criteria with the actual goal. Specifically, as we can see from Fig. 1(a), the loss of the existing segmentation-based method that misjudges a large instance is greater than two small instances, but for the human is converse, which leads to the module not being optimized by the way we envision it and deviate from the ideal state. PixelLink [8] makes every instance enjoy the same weight for the module optimization, which leads to smaller attention on pixels within large samples, resulting in incomplete detection. Additionally, as illustrated in Fig. 2, current segmentation-based methods predominantly use shrink masks for text prediction, which are generated by contracting the text inward. Compared to the text itself, the instance-occupying pixels of the shrink mask are more imbalanced, further exacerbating the neglect of the small-scale sample. Regarding the second issue, the above methods filter

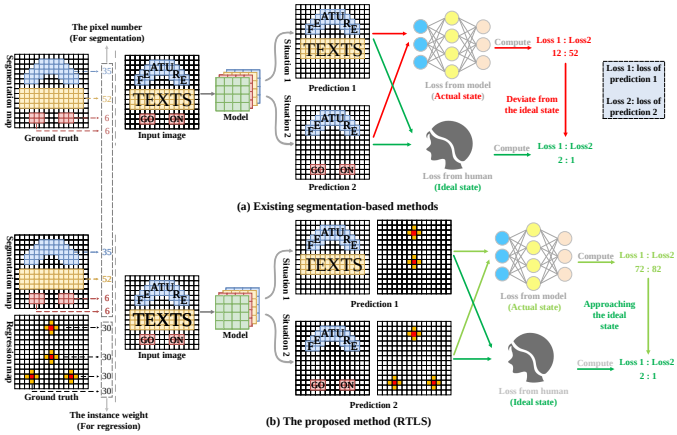


Fig. 1. The loss for the same prediction differs between the segmentation-based model and humans. For the model, the loss is based on the number of correctly predicted pixels. However, for humans, the loss is determined by the number of correctly predicted instances. The differences in pixel counts among different instances are substantial. There is a gap between the optimization goals of models and the objectives of humans.

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).

C. Yang, J. Gao, and Q. Wang are with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (E-mail: omtcyang@gmail.com, gjy3035@gmail.com, crabwq@gmail.com).

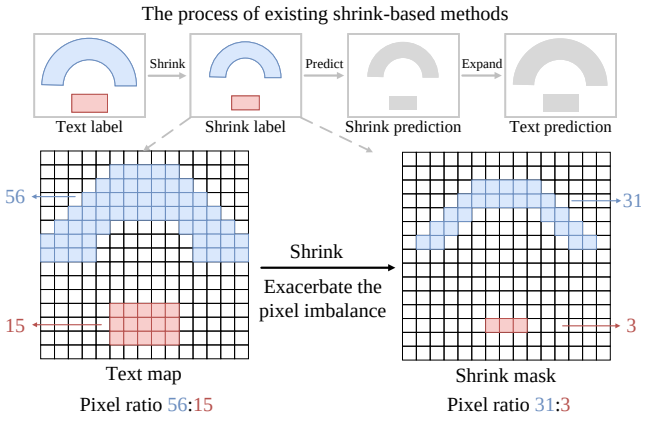


Fig. 2. Most existing segmentation-based methods are based on the shrink mask. For different instances, the proportion of reduced pixels from the text image to the shrink mask varies. Compared to the original text map label, the shrink mask exacerbates the pixel imbalance between instances.

candidates based on pixel predictions, but what is needed is an assessment of whether a region contains text, that deviates from the original intent of scene text detection.

To address the aforementioned issues, we propose an Instance-Equal Feature Guide Module (IEFGM), a Cross-Level Feature Interaction Module (CLIFM), and a Pixel-Instance Fusion Discriminator (PIFD). The IEFGM uses an instance kernel to transform segmentation pixel information into instance-level features, ensuring equal treatment of all instances. As shown in Fig. 1(b), it employs a parallel structure to increase the model's penalty for disregarding small instances at the feature level. This mitigates the segmentation model's tendency to overlook small samples and helps the model focus on instances lost during segmentation. It encourages the model to treat instances of different scales equally, enabling consistent feature modeling for text at various scales. It helps the segmentation-based model optimize in an ideal manner. As an auxiliary branch, it helps balance the optimization direction with the actual goals. Additionally, to further facilitate the interaction between instance hierarchies and pixel features, we propose the CLIFM. It extracts both instance and pixel features for corroboration, aiding the model in better understanding text features at different spatial scales and from various perspectives. However, addressing the issue solely from the feature perspective may not fully resolve the problem of segmentation models' inability to treat instances of different scales equally. Therefore, the PIFD is proposed to align with the assessment criteria and original intent from a result-oriented perspective. Unlike existing segmentation-based methods that filter candidate regions solely based on pixel-level predictions, the PIFD integrates both the probability that a pixel belongs to text and the probability that the region itself is text, providing results analyzed from multiple perspectives. Based on the above modules, we propose an efficient multi-scene text detector that balances optimization strategies with practical goals. The main contributions of this work are summarized as follows:

1) An instance equal feature guide module (IEFGM) is proposed, which extracts instance features and emphasizes equal treatment of instances. It helps the model optimize toward the real goal of instance detection.
2) A cross-level feature interaction module (CLIFM) is proposed, which encourages interaction between pixel-level and instance-level features, helping the model gain a deeper understanding of text from different levels.
3) A pixel-instance fusion discriminator (PIFD) is proposed, which differs from existing methods that solely filter candidate regions based on pixel-level predictions. It combines pixel- and instance-level predictions to comprehensively analyze candidate regions in alignment with the original intent.
4) An efficient and effective text detector named BOSPG (short for Balancing Optimization Strategies and Practical Goals) based on the above modules is proposed. It achieves state-of-the-art (SOTA) performance on multiple public datasets across various scenes.

The remaining parts of the paper are as follows. Section II presents related work on scene text detection. Specific methods and loss functions are introduced in the section III. Section IV showcases the ablation study and comparisons with existing state-of-the-art methods. Finally, the whole paper is summarized in Section V.

II. Related Works

Recently, deep learning has prompted scene text detection, making significant progress. According to the detection efficiency, these methods can be roughly divided into non-real-time and real-time methods.

A. Non-real-time methods

General object detection methods bring great inspiration for early scene text detection. Two-stage methods (such as Faster-RCNN [9]) obtained progress in accuracy, which is limited in detection speed. To speed up the inference process, many one-stage methods based on single shot multibox detector (SSD) [10] are proposed. TextBoxes [11] modified the convolution kernels and default anchors to adapt varied aspect ratios of the scene texts. On the basis of this, TextBoxes++ [12] detected multi-orientation texts by adding an angle parameter. EAST [13] divided text instances as rotated boxes and quadrangles that design different representations and loss functions. MOST [14] introduced a text feature alignment module that dynamically adjusts receptive fields according to initial detection results, and a position-aware non-maximum suppression technique to merge predictions based on positional information. RFN [15] proposed an improved feature attention network to address issues such as low visual contrast, uneven lighting, and surface corrosion commonly encountered in industrial scenes. These early methods only considered quadrangle text, which can't handle irregular-shaped texts. This issue is a major thrust of current research. To cope with it, PCR [16] proposed a progressive contour regression method. It first generated horizontal text proposals and then evolved the contours to oriented and arbitrary-shaped text proposals. FCENet [17] and ABCNet [18] designed the novel text representations to regress instance contours based on the Fourier signature vector and Bezier

curve, respectively. TextRay [19] converted the conventional Cartesian coordinate contour point representation into geometric encoding in the polar coordinate system. Taking inspiration from morphology, LeafText [1] incorporated leaf vein growth processes into text contour modeling, decomposing a concave object into continuous convex regions. Except for regression-based methods, many connected-component-based methods located characters or parts of text first and then joined them to reconstruct contours. Textsnake [20] represented text utilizing a series of circles and joined them like a snake. DRRG [21] introduced graph convolutional networks (GCN) to model the relationship of text parts. CRAFT [22] proposed character affinity to judge which characters belong to the same instance. SegLink++ [23] regressed the position information of text components, while simultaneously predicting the repulsion and affinity relationships between text components. Finally, it conducted post-processing based on an improved minimum spanning tree algorithm to generate text detection results. PSENet [24] predicted different scale kernels and recovered text contours by a progressive scaling method. ADNet [2] addressed the inconsistency between the contraction distance during shrink mask generation and the expansion distance during text contour restoration by proposing an adaptive dilation network from the perspective of text geometric features. Although these methods could deal with irregular-shaped texts, their low efficiency hinders their application in the real world.

B. Real-time method

To meet the demands of practical applications in the real world, there is an increasing focus on the combined performance of speed and accuracy. PAN [6] and PAN++ [25] established fast post-processing to reconstruct text contour from text kernel regions based on PSE-Net [24], significantly improving detection speed without compromising performance. Although their speed is faster than previous methods, pixel-wise post-processing also occupies some time. CT [7] proposed an efficient text representation that decomposed text instances into text kernels and centripetal shifts. The corresponding post-processing is simple and fast. DBNet [4] proposed an instance-level extension post-processing strategy to save time. In addition, it did not need extra prediction, which is more robust and suitable for other methods. To further improve the detection performance without influencing the inference speed, DBNet proposed a differentiable binarization module, which can be removed during the test stage. Based on it, DBNet++ [5] added an adaptive scale fusion module to help the module recognize various scale texts. CM-Net [26] proposed a more robust and effective representation named concentric mask. RSMTD [3] relied on the extension distance predictions to correct the false kernel predictions. Inspired by the camera zooming process, ZTD [27] introduced a zoom detection module to enhance edge recognition capability and reserve the edge details. HFENet [28] improved the quality of high-frequency information in both scene text and traffic signs. Additionally, a new boundary enhancement module was proposed to strengthen local feature extraction capability. Although the above methods achieve great performance, the accuracy is still unable to meet the demands of practical applications.

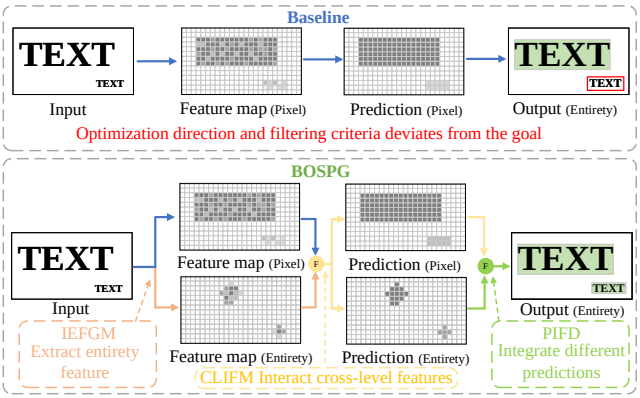


Fig. 3. Differences in structure between the baseline and BOSPG. Existing segmentation methods optimize based on pixel-level predictions, with features of large-scale text dominating the optimization direction of the model, deviating the instance equal goals. The proposed method utilizes IEFGM, CLIFM, and PIFD to alleviate this issue. Specifically, the former advocates for instance equality from a feature perspective, where the CLIFM encourages the cross-level feature mutual. The latter utilizes both pixel and instance predictions to align with the original intention.

III. Methodology

This section illustrates the whole pipeline of the proposed method. Subsequently, we describe the instance equal feature guide module (IEFGM), cross-level feature interaction module (CLIFM), and pixel-instance fusion discriminator (PIFD) in detail. We then present the loss functions used and the corresponding label generation process. Finally, we introduce the post-processing of the BOSPG.

A. Overall Structure

As shown in Fig. 3, existing segmentation methods optimize the model and evaluate candidate regions based on pixel probability predictions. This approach deviates from the original goal of instance detection and results in models that overemphasize low-level texture information. The proposed method addresses this issue at both the feature and output levels. The overall structure of the proposed BOSPG is illustrated in Fig. 4, comprising the feature extraction module, the feature pyramid network (FPN), the instance-equal feature guide module (IEFGM), the cross-level feature interaction module (CLIFM), and the pixel-instance fusion discriminator (PIFD). During the training stage, multi-scale features are extracted through the backbone. Subsequently, the FPN [29] is used to fuse these feature maps to obtain the feature map F containing high-level semantic features and low-level texture features. The shrink mask segmentation head is utilized to predict the probability map of the shrink mask. The IEFGM introduces instance position information at the feature level, encouraging the module to treat text at different scales equally. The CLIFM facilitates interaction between features of different levels. During testing, the PIFD integrates predictions from the instance level and pixel level to make a comprehensive judgment. Finally, the predicted shrink mask is binarized and expanded to generate the final detection results.

The details of the proposed method are described as follows. ResNet [30] with deformable convolution [31], [32] is adopted

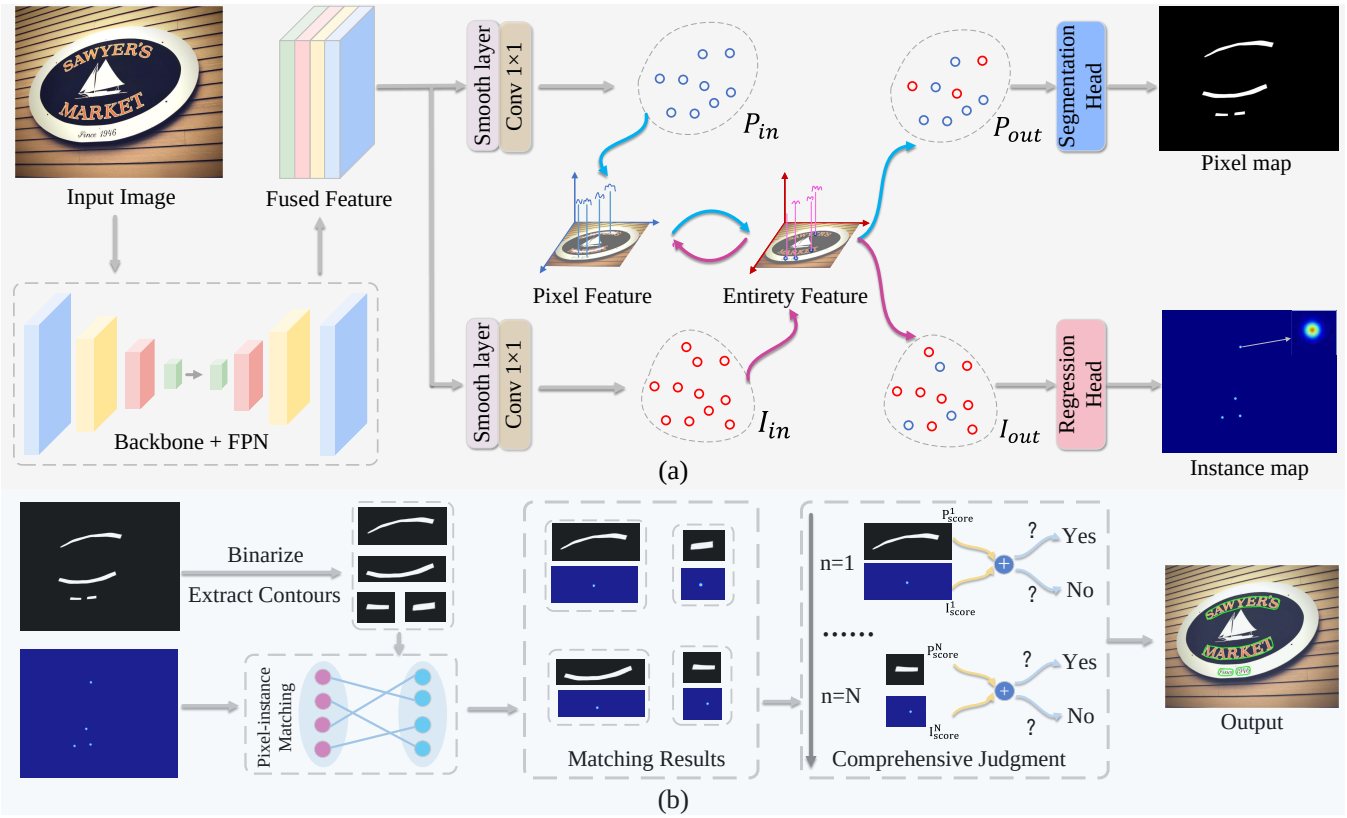


Fig. 4. The whole pipeline of the proposed BOSPG. (a) The network of the proposed method consists of the backbone, the feature pyramid network (FPN), an Instance-Equal Feature Guide Module (IEFGM), and a Cross-Level Feature Interaction Module (CLIFM). (b) The post-processing that Pixel-Instance Fusion Discriminator (PIFD).

as the backbone. Based on it, multi-level feature maps are derived, with sizes of $\frac{H}{4} \times \frac{W}{4} \times C$, $\frac{H}{8} \times \frac{W}{8} \times 2C$, $\frac{H}{16} \times \frac{W}{16} \times 4C$, and $\frac{H}{32} \times \frac{W}{32} \times 8C$, where C represents the channel number of the feature map, and W and H denote the width and height of the image, respectively. The shrink mask segmentation head is constructed as follows:

$$\mathrm{P}_1 = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{3\times3,64}(\mathrm{P}_{\mathrm{out}}))), \tag{1}$$

$$\mathrm{P}_2 = \mathrm{ReLU}(\mathrm{BN}(\mathrm{ConvT}_{2\times2,64}(\mathrm{P}_1))), \tag{2}$$

$$\mathrm{P}_{\mathrm{res}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{ConvT}_{2\times2,1}(\mathrm{P}_2))), \tag{3}$$

where $\mathrm{P}_{\mathrm{out}}$ represents the feature map generated from CLFIM. P_1 and P_2 are the hidden feature maps. BN and ReLU represent batch normalization layer [33] and ReLU activation function. $\mathrm{P}_{\mathrm{res}}$ represents the shrink mask prediction.

B. Instance Equal Feature Guide Module

Existing segmentation-based methods obtain detection results by optimizing them using pixel-wise prediction. Missing a large or small-scale instance is the same for us, that equals missing an object. However, for the segmentation-based module, the error of missing a large sample and a small sample differs greatly, that difference may be several times or dozens of times. In some cases, the segmentation-based method will choose to abandon two small samples and retain one large sample, which is contrary to our original intention of detecting text. To alleviate this problem, we propose an instance equal module feature guide (IEFGM), which encourages the module to treat different scale samples equally. The structure is described as follows:

$$\mathrm{I}_1 = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{3\times3,\frac{\mathrm{C}}{4}}(\mathrm{I}_{\mathrm{out}}))), \tag{4}$$

$$\dot{\mathrm{I}}_{\mathrm{res}} = \mathrm{ReLU}(\mathrm{Conv}_{1\times1,1}(\mathrm{I}_1)), \tag{5}$$

$$\mathrm{I}_{\mathrm{res}} = \mathrm{Up}_{\times4}(\dot{\mathrm{I}}_{\mathrm{res}}), \tag{6}$$

where $\mathrm{I}_{\mathrm{out}}$ and I_1 represent the feature generated from the CLFIM and hidden feature map. $\mathrm{I}_{\mathrm{res}}$ is the instance map prediction. $\mathrm{Up}_{\times4}$ represent the operation of upsamples 4 times.

The Algorithm 1 is used to convert a shrink mask S_m to a point P. We adopt Gaussian convolutions to process different points, which generate a Gaussian-expanded region where the Gaussian distributions will be superimposed for points that are closer. The process can be formulated as follows:

$$G(x_i, y_i) = \frac{1}{2\pi\sigma^2} e^{-\frac{x_i^2+y_i^2}{2\sigma^2}} \times \theta, \mathrm{s.t.}(\mathrm{x_i}, \mathrm{y_i}) \in \mathrm{P}, \tag{7}$$

where $G(x_i, y_i)$ represent the Gaussian kernel superimposed on points P.

Segmentation-based methods treat each pixel equally to optimize the model, granting large-scale text more influence compared to small-scale instances. The above operation disregards the scale of the texts, uniformly transforming them into standardized instances. For the model, the cost of ignoring text at different scales remains the same, which, to some extent,

Algorithm 1: Instance center generation process

Input: Shrink mask S_m
Output: Center point P_c

Let L_p = []
$X_{Max} \leftarrow$ the maximum value on the x-axis in S_m
$X_{Min} \leftarrow$ the minimum value on the x-axis in S_m
$X_{mid} = \frac{X_{max}+X_{min}}{2}$
for pixle p_j in S_m **do**
  if the value on the x-axis of p_j = X_{mid} **then**
    Append (p_j) to L_p.
  end if
  $Y_{mid} \leftarrow$ the mean of points in L_p on the y-axis.
  Center points $P_c \leftarrow (X_{mid}, Y_{mid})$
end for
return P_c

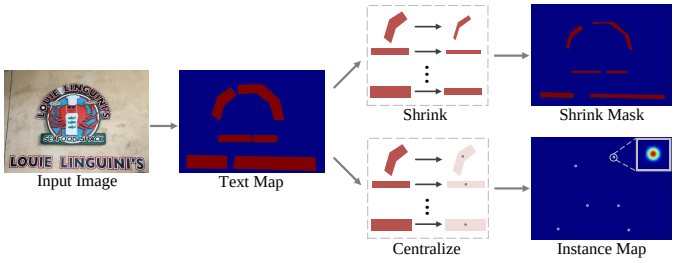


Fig. 5. The visualization of the label generation. The shrink mask and instance map are generated based on the text map.

reflects the segmentation method's tendency to undervalue small-scale text. Specifically, assuming that the area ratio of instances A, B, and C is 100:2:1, the loss incurred by a segmentation-based method for missing instances B and C is only 3% of the loss for ignoring instance A. In contrast, for IEFGM, the loss from missing instances B and C is twice that of ignoring instance A, which better aligns with the goal of text detection.

C. Cross-Level Feature Interaction Module

Instance features and pixel features represent different levels of knowledge, and existing methods lack interaction between these levels. To address this, we propose a cross-level feature interaction module (CLFIM) to facilitate the model's comprehensive understanding of these two distinct levels of features. Specifically, it leverages high-level instance features to generate constraint scores that refine low-level pixel features, while conversely using low-level pixel features to further constrain instance features. The structure of the CLFIM is as follows:

$$\mathrm{I}_{\mathrm{in}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{1\times1}(\mathrm{F}))), \tag{8}$$

$$\mathrm{P}_{\mathrm{in}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{1\times1}(\mathrm{F}))), \tag{9}$$

$$\mathrm{I}_{\mathrm{out}} = \mathcal{C}(\mathrm{I}_{\mathrm{in}}, \mathrm{P}_{\mathrm{in}}), \tag{10}$$

$$\mathrm{P}_{\mathrm{out}} = \mathcal{C}(\mathrm{P}_{\mathrm{in}}, \mathrm{I}_{\mathrm{in}}), \tag{11}$$

where I_{in} and P_{in} represent the input instance and pixel feature maps, respectively. $\mathrm{I}_{\mathrm{out}}$ and $\mathrm{P}_{\mathrm{out}}$ represent the output instance and pixel feature maps, respectively. $\mathcal{C}$ represents the feature match block, which can be formulated as follows:

$$\mathcal{C}(\mathrm{X}, \mathrm{Y}) = \mathrm{Sigmoid}(\mathrm{Conv}_{1\times1}(\mathcal{A}(\mathrm{Y}))) * \mathrm{X}, \tag{12}$$

where $\mathcal{A}(\cdot)$ represent AvgPool operation.

Moreover, BOSPG is a method that achieves strong performance under both real-time and non-real-time conditions. As a component of the BOSPG, CLIFM aims to achieve feature interaction at two hierarchical levels while minimizing additional computational costs. Compared to attention-based interaction methods, its most significant advantages are its simplicity and low computational overhead.

D. Pixel-Instance Fusion Discriminator

Segmentation methods play a significant role in scene text detection with arbitrary shapes, owing to their flexible pixel-level predictions. However, this advantage comes with certain challenges: 1) The model's optimization is primarily guided by pixel classification, which emphasizes low-level texture features while neglecting the holistic features of instances, misaligning the optimization with the actual objective. 2) Candidate regions are selected based on pixel classification scores, causing the evaluation criteria to diverge from the actual goal. The first issue can be mitigated by the aforementioned ISFGM. To tackle the second issue, we propose the Pixel-Instance Fusion Discriminator (PIFD). As shown in Fig. 3, unlike existing segmentation methods that select candidate regions solely based on pixel-level results, the proposed PIFD integrates both pixel and instance scores. This approach balances low-level texture information with high-level semantic information, outperforming existing methods. The specific structure can be represented as follows:

$$\mathrm{I}_{\mathrm{res}} = \mathcal{R}_{\mathcal{H}}(\mathrm{I}_{\mathrm{out}}), \tag{13}$$

$$\mathrm{P}_{\mathrm{res}} = \mathcal{S}_{\mathcal{H}}(\mathrm{P}_{\mathrm{out}}), \tag{14}$$

where $\mathcal{R}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ represent regression head and segmentation head, respectively. $\mathrm{I}_{\mathrm{res}} \in \mathbb{R}^{1\times H\times W}$ and $\mathrm{P}_{\mathrm{res}} \in \mathbb{R}^{1\times H\times W}$ represent the prediction of the instance regression map and pixel segmentation map, respectively. Immediately following, the $\mathrm{P}_{\mathrm{res}}$ is binarized first:

$$\overline{\mathrm{P}}_{\mathrm{res}} = \mathcal{B}(\mathrm{P}_{\mathrm{res}}), \tag{15}$$

$$\mathrm{P}_{\mathrm{ins}} = \mathcal{CE}(\overline{\mathrm{P}}_{\mathrm{res}}), \tag{16}$$

where $\mathcal{CE}$ and $\mathrm{P}_{\mathrm{ins}}$ represent the contour extraction operation and the list of candidate instances, respectively.

$$\mathrm{P}^i_{\mathrm{score}} = \frac{\sum_{j\in M_i} \mathrm{P}^j_{\mathrm{res}}}{|M_i|}, \quad i = 1, 2, 3, \ldots, n \tag{17}$$

$$\mathrm{I}^i_{\mathrm{score}} = \frac{\sum_{j\in M_i} \mathrm{I}^j_{\mathrm{res}}}{\theta}, \quad i = 1, 2, 3, \ldots, n \tag{18}$$

where n and M_i represent the number of instances and the binarized mask of the ith instance. $\mathrm{P}^i_{\text{score}}$ and $\mathrm{I}^i_{\text{score}}$ represent the pixel average score and instance score, respectively.

$$\mathcal{J}^i = \begin{cases} T, & \text{if } \mathrm{I}^i_{\text{score}} + \mathrm{P}^i_{\text{score}} >= \delta \quad i = 1, \ldots, n \\ F, & \text{if } \mathrm{I}^i_{\text{score}} + \mathrm{P}^i_{\text{score}} < \delta \quad i = 1, \ldots, n \end{cases}, \tag{19}$$

where $\mathcal{J}^i$ represent the judge result of the ith instance prediction. $\delta \in [0, 2]$ is a hyperparameter to balance the precision and recall. T and F represent the qualified sample and unqualified sample, respectively.

E. Label generation process

In this section, the label generation of the shrink mask and instance map is introduced, which is shown in Fig. 5. Each instance is labeled with some points. First, these points are converted to a text binary map. Subsequently, the shrink mask region is generated by shrinking the text region with an offset. The shrink offset O is calculated by:

$$O = \frac{S \times (1 - \gamma^2)}{L}, \tag{20}$$

where S and L are the area and perimeter of the text. γ is the shrink ratio, which is set at 0.4.

F. Loss function

In this section, the proposed BOSPG has two tasks that need to be optimized. The total loss $\mathcal{L}$ consists of shrink mask prediction loss $\mathcal{L}_{sm}$ and instance map prediction loss $\mathcal{L}_g$, which can be further described as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{sm} + \lambda_2 \mathcal{L}_g, \tag{21}$$

where λ_1 and λ_2 represent the weight of $\mathcal{L}_{sm}$ and $\mathcal{L}_g$, respectively. The specific details are introduced as follows.

1) Shrink mask loss: The binary cross-entropy (BCE) loss is adopted to optimize the shrink mask prediction. The text area usually takes up only a small portion of the image. To relieve the imbalance between the positive and negative samples, we introduce hard negative mining in BCE loss, which can be formulated as follows:

$$\mathcal{L}_{sm} = \sum_{p \in S} -x^*_p \times log(x_p) - (1 - x^*_p) \times log(1 - x_p), \tag{22}$$

where S represents the selected pixel set. x^*_p and x_p represent the ground truth and prediction of the pixel p.

2) Instance map loss: For the instance map prediction, the Mean Squared Error (MSE) Loss is adopted to optimize the instance map prediction, which can be described as follows:

$$\mathcal{L}_g = \sum_p (y_p - y^*_p)^2, \tag{23}$$

where y^*_p and y_p represent the ground truth and prediction of instance map at the pixel p.

G. Inference

During the inference stage, the predicted shrink mask is binarized first. Some tiny-scale regions are abandoned. Then, the shrink mask is extended a distinct distance $\overline{O}$, which can be formulated as:

$$\overline{O} = \frac{\overline{S} \times \beta}{\overline{L}}, \tag{24}$$

where $\overline{S}$ and $\overline{L}$ represent the area and perimeter of the predicted shrink mask.

IV. Experiments

A. Datasets

MSRA-TD500 [34] 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 [4], we introduce the HUST-TR400 [35] to supplement the training dataset.

ICDAR2015 [55] is composed of 1,000 training images and 500 test images, which enjoy complex backgrounds. Each instance is labeled with four corner points. The resolution of the image is 720 × 1280.

CTW1500 [56] mainly contains curved texts. It includes 1,000 training images and 500 testing images. Each text instance is labeled at the line level.

ASAYAR-TXT is one of the subset datasets of ASAYAR [57], which is collected from Moroccan highways. This dataset mainly includes text images and each image in it has word-level and line-level annotations. It contains 1,100 images and 275 testing images.

Total-Text [58] consists of various texts, including horizontal, tilted, and irregular instances. It contains 1,255 training images and 300 testing images and the text instances are annotated at the word level.

ICDAR2017MLT [59] is a multilingual dataset comprising 7,200 images in the training set, 1,800 images in the validation set, and 9,000 images in the test set.

B. Evaluation Metrics

To ensure a fair comparison, we keep the same as the previous methods to adopt precision (P), recall (R), F-measure (F), and FPS to evaluate the performance and efficiency, which can be formulated as follows:

$$P = \frac{TP}{TP + FP} \times 100\%, \tag{25}$$

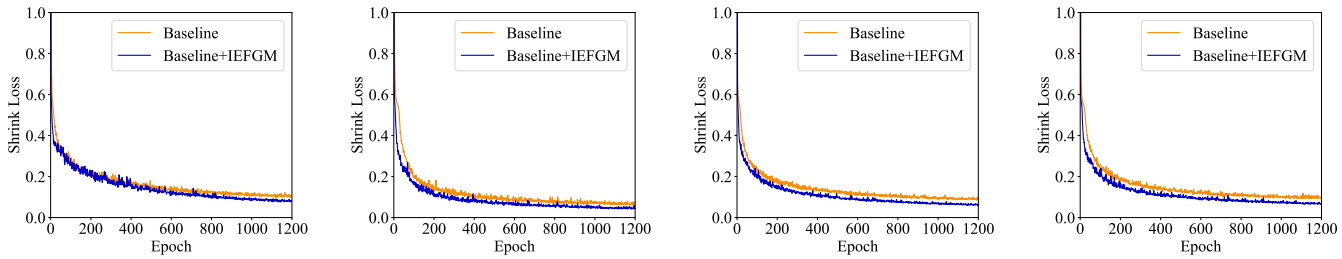
$$R = \frac{TP}{TP + FN} \times 100\%, \tag{26}$$

$$F = \frac{2 \times P \times R}{P + R} \times 100\%. \tag{27}$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively.

TABLE I
THE COMPARISON OF DETECTION RESULTS UNDER DIFFERENT CONDITION SETTINGS ON THE ICDAR2015 AND MSRA-TD500.

ID	Methods	IEFGM	CLIFM	PIFD	ICDAR2015			MSRA-TD500		
					Precision	Recall	F-measure	Precision	Recall	F-measure
1	baseline+	×	×	×	84.4	81.2	82.8	86.8	80.2	83.4
2	baseline+	✓	×	×	86.3	83.2	84.7	87.0	81.4	84.1
3	baseline+	✓	✓	×	87.1	84.1	85.6	89.4	82.0	85.6
4	baseline+	✓	×	✓	88.8	83.1	85.8	86.2	84.5	85.3
5	baseline+	✓	✓	✓	89.7	83.1	86.3	89.8	82.8	86.1



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

Fig. 6. Shrink mask loss under different schemes on the ICDAR2015, MSRA-TD500, Total-Text, and CTW1500, respectively.

TABLE II
THE PERFORMANCE OF BOSPG UNDER DIFFERENT INSTANCE KERNEL SIZES ON THE ICDAR 2015 AND MSRA-TD500 DATASETS.

	η	ICDAR2015			MSRA-TD500		
		P	R	F	P	R	F
BOSPG with	11	89.4	82.4	85.6	88.4	83.7	86.0
BOSPG with	13	89.0	83.2	86.0	89.8	82.8	86.1
BOSPG with	15	89.7	83.1	86.3	90.4	82.1	86.0
BOSPG with	17	88.1	84.1	86.0	87.8	84.4	86.1
BOSPG with	19	89.0	82.8	85.8	89.5	82.1	85.7

C. Implementation Details

ResNet [30] with deformable convolution [31], [32] is adopted as the backbone. Feature Pyramid Network (FPN) is used to fuse different scale feature maps. The training batch size is set to 16. We adopt two training strategies: (1) The model is directly trained on the public datasets. (2) Followed by the CBNet [53] and TextBPN++ [60], the model is pre-trained on ICDAR2017MLT [59] for 300 epochs and then is finetuned on the real dataset. For the ResNet18, the optimizer and initial learning rate are set to Adam [61] and 0.001, respectively. For the ResNet50, the optimizer and initial learning rate are set to Adam and 0.0001, respectively. We follow the "poly" strategy to adjust the learning rate during the training stage. For the CTW1500 dataset, when employing ResNet18 as the backbone network, the total number of parameters is 12.78M, with a computational complexity of 42.97 GFLOPs. When using ResNet50 as the backbone, the total parameter count increases to 29.08M, accompanied by a computational cost of 81.97 GFLOPs. The inference speed of the model in real applications can be further improved by utilizing TensorRT or following PAN [6] to fuse the convolution and batch normalization (BN) [33] layers. The data augmentation includes random rotation, cropping, and flipping. During the test stage, the short side of the input image from the same dataset is resized to the same size while maintaining the aspect ratio. All the speed is tested on a single GTX 1080Ti GPU and an i7-6800K CPU.

D. Ablation Study

To demonstrate the effectiveness of the proposed module, we conduct ablation studies on the ICDAR2015 and MSRA-TD500 datasets. Note that, all ablation experiments are not pre-trained to avoid the influence of additional data, and the backbone is set to ResNet18.

1) The effectiveness of IEFGM: Most segmentation-based text detection methods are optimized by pixel-level predictions, which can easily lead to model optimization being dominated by large-scale text features. As shown in Table I, the proposed IEFGM improves detection performance significantly on the ICDAR2015 dataset. Specifically, it achieves 1.9% and 0.7% improvement in F-measure on the ICDAR2015 and MSRA-TD500, respectively. In addition, as we can see from Fig. 6, compared to the baseline, the introduction of the IEFGM accelerates the drop of shrink mask loss on the ICDAR2015, Total-Text, CTW1500, and MSRA-TD500 datasets. IEFGM can assist the model in extracting consistent features at different scales of the shrink mask and alleviate the optimization direction of the model from being dominated by large-scale text features, which helps the module converge to a better solution.

2) The effectiveness of CLIFM: As shown in Table I, equipping the IEFGM with the CLIFM improves the precision, recall, and F-measure, by 0.8%, 0.9%, and 0.9%, respectively, on the ICDAR2015 dataset. For the MSRA-TD500 dataset, the improvements are 2.4%, 0.6%, and 1.5% in precision, recall, and F-measure, respectively. The experiments demonstrate the effectiveness of the proposed CLIFM. The above experiment confirms that the interaction between pixel-level and instance-level features benefits the model.

3) The effectiveness of PIFD: Relying only on pixel-level predictions to filter instance-level results is prone to errors.

TABLE III
COMPARISON WITH EXISTING STATE-OF-THE-ART (SOTA) REAL-TIME METHODS ON THE CTW1500,TOTAL-TEXT, AND MSRA-TD500 DATASETS, RESPECTIVELY. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL, AND F-MEASURE, RESPECTIVELY. THE BEST AND SUBOPTIMAL RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY.

Methods	Venue	Backbone	CTW1500				TotalText				MSRA-TD500			
			P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
DBNet [4]	AAAI'20	ResNet18	84.8	77.5	81.0	55	88.3	77.9	82.8	50	90.4	76.3	82.8	62
CT [7]	NeurIPS'21	ResNet18	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++ [25]	TPAMI'22	ResNet18	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 [26]	TMM'22	ResNet18	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 [28]	TITS'23	ResNet18	85.1	81.2	83.1	32.2	85.7	81.7	83.7	22.0	89.7	81.1	85.2	40.9
FS [36]	TIP'23	ResNet18	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 [3]	TMM'23	ResNet18	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++ [5]	TPAMI'23	ResNet18	86.7	81.3	83.9	40	87.4	79.6	83.3	48	87.9	82.5	85.1	55
ZTD [27]	TNNLS'24	ResNet18	88.4	80.2	84.1	76.9	90.1	82.3	86.0	75.2	91.6	82.4	86.8	59.2
BOSPG	**Ours**	ResNet18	86.9	86.9	**86.9**	48.3	89.6	87.8	**88.7**	44.8	93.5	88.8	**91.1**	53.6

TABLE IV
COMPARISON WITH EXISTING STATE-OF-THE-ART (SOTA) APPROACHES ON THE CTW1500,TOTAL-TEXT, AND MSRA-TD500 DATASETS. "-" REPRESENTS THE CORRESPONDING RESULTS THAT ARE NOT REPORTED IN THE PAPER. THE BEST AND SUBOPTIMAL RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY.

Methods	Venue	Back.	CTW1500				TotalText				MSRA-TD500			
			P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
OPMP [37]	TMM'21	ResNet50	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	ResNet50	87.6	83.4	85.5	-	89.3	82.5	85.8	-	-	-	-	-
DText [38]	PR'22	ResNet50	86.9	82.7	84.7	-	90.5	82.7	86.4	-	87.9	83.1	85.4	-
I3CL [39]	IJCV'22	ResNet50	88.4	84.6	86.5	-	89.8	84.2	86.9	-	-	-	-	-
NASK [40]	TCSVT'22	ResNet50	83.4	80.1	81.7	12.1	85.6	83.2	84.4	8.4	-	-	-	-
LEMNet [41]	TMM'22	ResNet50	86.6	83.8	85.2	-	89.9	85.4	87.6	-	85.6	84.8	85.2	-
HFENet [28]	TITS'23	ResNet50	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 [42]	TMM'23	ResNet50	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++ [5]	TPAMI'23	ResNet50	87.9	82.8	85.3	26	88.9	83.2	86.0	28	91.5	83.3	87.2	29
RP-Text [43]	TMM'23	ResNet18	87.8	81.6	84.7	-	89.4	82.8	86.0	-	88.4	84.6	86.5	-
KPN [44]	TNNLS'23	ResNet50	84.4	84.2	84.3	16.3	88.7	85.6	87.1	15.0	-	-	-	-
FS [36]	TIP'23	ResNet50	85.3	82.5	83.9	25.1	88.7	79.9	84.1	24.3	89.3	81.6	85.3	25.4
DPText-DETR [45]	AAAI'23	ResNet50	91.7	86.2	88.8	-	91.8	86.4	89.0	-	-	-	-	-
MorphText [46]	TMM'23	ResNet50	90.0	83.3	86.5	-	90.6	5.2	87.8	-	90.7	83.5	87.0	-
ASSTD [47]	TMM'23	VGG16	89.8	83.3	86.4	-	89.4	85.8	87.6	-	90.5	83.8	87.0	-
LeafText [1]	TMM'23	ResNet18	87.1	83.9	85.5	-	90.8	84.0	87.3	-	92.1	83.8	87.8	-
ADNet [2]	TMM'23	ResNet50	88.2	83.1	85.6	-	90.6	84.4	87.4	-	92.0	83.2	87.4	-
SRFormer [48]	AAAI'24	ResNet50	89.4	89.8	**89.6**	-	91.5	87.9	89.7	-	-	-	-	-
TTDNet [49]	TITS'24	ResNet50	87.4	82.2	84.7	4.6	-	-	-	-	90.4	83.9	87.0	-
STD [50]	TMM'24	ResNet50	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 [51]	TCSVT'24	ResNet50	88.7	86.3	87.5	-	91.2	85.0	88.0	-	93.4	88.2	90.7	-
FEPE [52]	TMM'24	ResNet50	88.8	83.5	86.0	22	91.3	81.9	86.4	32	90.5	85.4	88.0	32
CBNet [53]	IJCV'24	ResNet18	89.0	81.9	86.0	-	90.1	82.5	86.1	-	91.1	84.8	87.8	-
CT-Net [54]	TCSVT'24	ResNet50	88.5	83.8	86.1	11.2	90.8	85.0	87.8	10.1	90.8	84.4	87.5	11.6
BOSPG	**Ours**	ResNet50	87.4	86.8	87.1	21.6	89.5	88.8	**89.1**	16.0	94.9	92.3	**93.6**	18.0

To address this problem, we propose a PIFD to encourage the module to combine different levels of prediction results for screening. As we can see from Table I, when equipping the IEFGM, the PIFD improves the F-measure by 1.1% and 1.2% on the ICDAR2015 and MSRA-TD500 datasets. Moreover, based on the IEFGM and CFIFM, the PIFD brings 0.7% and 0.5% improvements on the ICDAR2015 and MSRA-TD500. The experiments demonstrate the effectiveness of the PIFD.

4) The effectiveness of Gauss kernel size: We conduct experiments with the proposed BOSPG on the ICDAR2015 and MSRA-TD500 datasets under different Gauss kernel sizes. As shown in Table II, the proposed BOSPG achieves the best performance at 86.3% and 86.1% in F-measure on the ICDAR2015 and MSRA-TD500, respectively. These experiments show the appropriate setting and we set the size of the Gauss kernel to 15 in the following experiments.

E. Comparisons with Previous Methods

We evaluate the proposed BOSPG with existing state-of-the-art (SOTA) methods across six public datasets to demonstrate its superiority and versatility. The first four are collected from natural scenes, with one for validating multi-directional text, two for validating irregular text, and one for validating long text. The images in the last two datasets are from traffic and industrial scenes.

1) Evalution on CTW1500: To confirm the robustness of the proposed method for handling curved text at the line level, experiments are performed on the CTW1500 dataset. Existing state-of-the-art methods are divided into real-time and non-real-time categories. As indicated in Table III, BOSPG-ResNet18 significantly outperforms existing real-time methods. The proposed BOSPG achieves an F-measure of 86.9%, which exceeds the existing advanced methods FS [36], RSMTD [3], and DBNet++ [5] by 5.9%, 3.0%, and 3.0%,

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

Methods	Back.	P	R	F	FPS
DB [4]	ResNet18	86.8	78.4	82.3	48
PAN++ [25]	ResNet18	85.9	80.4	83.1	28.2
CM-Net [26]	ResNet18	86.7	81.3	83.9	34.5
ZTD [27]	ResNet18	87.5	79.0	83.0	48.3
DB++ [5]	ResNet18	90.1	77.2	83.1	44
BOSPG	ResNet18	89.7	84.3	**86.9**	47.2
DText [38]	ResNet50	88.5	85.6	87.0	-
LEMNet [41]	ResNet50	88.3	85.9	87.1	-
DB [4]	ResNet50	91.8	83.2	87.3	12
FS [36]	ResNet50	89.8	82.7	86.1	12.1
FCENet [17]	ResNet50	90.1	82.6	86.2	
LeafText [1]	ResNet50	88.9	82.9	86.1	-
DB++ [5]	ResNet50	90.9	83.9	87.3	10
VTD [62]	ResNet50	88.5	85.8	87.1	-
CBNet [53]	ResNet50	91.0	85.4	**88.1**	-
RP-Text [43]	ResNet18	89.6	82.4	85.9	-
KPN [44]	ResNet50	88.3	84.8	87.4	6.3
ADNet [2]	ResNet50	92.5	83.7	87.9	-
TTDNet [49]	ResNet50	90.0	85.6	87.7	-
SMNet [63]	ResNet50	89.7	85.5	87.6	-
BOSPG	ResNet50	88.5	87.6	**88.1**	6.7

TABLE VI
THE COMPARISON WITH EXISTING STATE-OF-THE-ART METHODS ON THE MPSC DATASET. THE BEST PERFORMANCE IS LABELED IN **BOLD**. $*$ REPRESENTS THE DETECTION RESULTS ARE COLLECTED FROM [15].

Methods	Precision	Recall	F-measure
EAST* [13]	83.6	73.5	78.2
MASK R-CNN* [64]	85.3	73.5	82.2
RRPN* [65]	82.0	78.9	80.4
PSENet* [24]	85.4	78.4	81.8
PAN* [6]	87.1	81.6	84.2
BDN* [66]	86.6	77.5	81.8
ContourNet* [67]	87.8	81.0	84.3
RRPN++* [68]	86.7	83.9	85.3
FCENet* [17]	87.1	81.6	84.3
RFN [15]	89.3	83.3	86.2
BOSPG-ResNet50	89.7	87.4	**88.5**

respectively. In addition, it even outperforms almost all non-real-time methods, such as CT-Net [54], DBNet++ [5], and KPN [44]. As we can see from Tab IV, even though it is lower than transformer-based SRFormer [48] and DPText-DETR [45], it still surpasses the existing SOTA methods MorphText [46], RP-Text [43], and TextDCT [42] by 0.6%, 2.4%, and 2.0% in F-measure. Additionally, compared to the transformer-based models, our method enjoys a lower computational cost and achieves competitive performance even when using the lightweight ResNet-18 as the backbone, striking a balance between performance and inference speed.

2) Evalution on Total-Text: Total-Text contains both horizontal text, multi-directional text, and curved text. To verify the shape robustness of the proposed BOSPG, we conduct experiments on it. As exhibited in Table III, the BOSPG using ResNet-18 achieves precision, recall, and F-measure of 89.6%, 87.8%, and 88.7%, respectively. Compared with the previous SOTA methods DBNet++ [5], RSMTD [3], and ZTD [27], it improves by 5.4%, 2.6%, and 2.7% in F-measure. It is also superior to most existing SOTA non-real-time methods. As shown in Table IV, it achieves the best performance among various real-time methods. Moreover, with the help of the

TABLE VII
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
Texboxes++ [57]	66	52	58
CTPN [57]	80	95	86
CTPN+Baseline [57]	83	97	89
HFENet [28]	96.3	97.2	96.8
BOSPG	98.6	98.2	**98.4**

Fig. 7. Visualization of some prediction results, including shrink mask, instance map, and post-processing output.

ResNet50, our method exceeds the existing advanced methods CT-Net [54], ADNet [2], and LeafText [1] by 1.3%, 1.7%, and 1.8%, respectively. For the transform-based models, the proposed method performs between DPText-DETR [45] and SRFormer [48] in terms of performance and has a significant advantage in inference speed. The above experiments strongly demonstrate that our method can effectively handle arbitrary-shaped scene texts.

3) Evalution on MSRA-TD500: MSRA-TD500 is a multi-oriented line-level labeled dataset. The existing methods are divided into real-time and non-real-time according to the inference speed. As listed in Table III, When adopting the ResNet18 as the backbone, the proposed BOSPG achieves 93.5%, 88.8%, and 91.1% in precision, recall, and F-measure, respectively. It significantly outperforms existing real-time methods and also outperforms most methods using ResNet50 as the backbone. Compared to the existing (SOTA) real-time method ZTD [27], it improves 4.3% in F-measure. As shown in Tab IV, for the ResNet50 as the backbone, the proposed method achieves precision, recall, and F-measure of 94.9%, 92.3%, and 93.6%, respectively. Compared to the existing SOTA methods TPPAN [51], BOSPG improves 1.6%, 4.1%, and 2.9% in precision, recall, and F-measure, respectively. The above experiment effectively demonstrates the superiority of the proposed BOSPG.

4) Evalution on ICDAR2015: The images in the ICDAR2015 dataset enjoy complex and blurry backgrounds. As same as the MSRA-TD500, existing methods are classified into real-time and non-real-time methods. According to Table V, utilizing ResNet18 as the backbone, our BOSPG significantly outperforms existing real-time methods, achieving an F-measure of 86.9%. It exceeds the performance of prior SOTA methods ZTD [27], DBNet++ [5], and CMNet [26] by 3.9%, 3.8%, and 3.0%, respectively. In addition, its performance

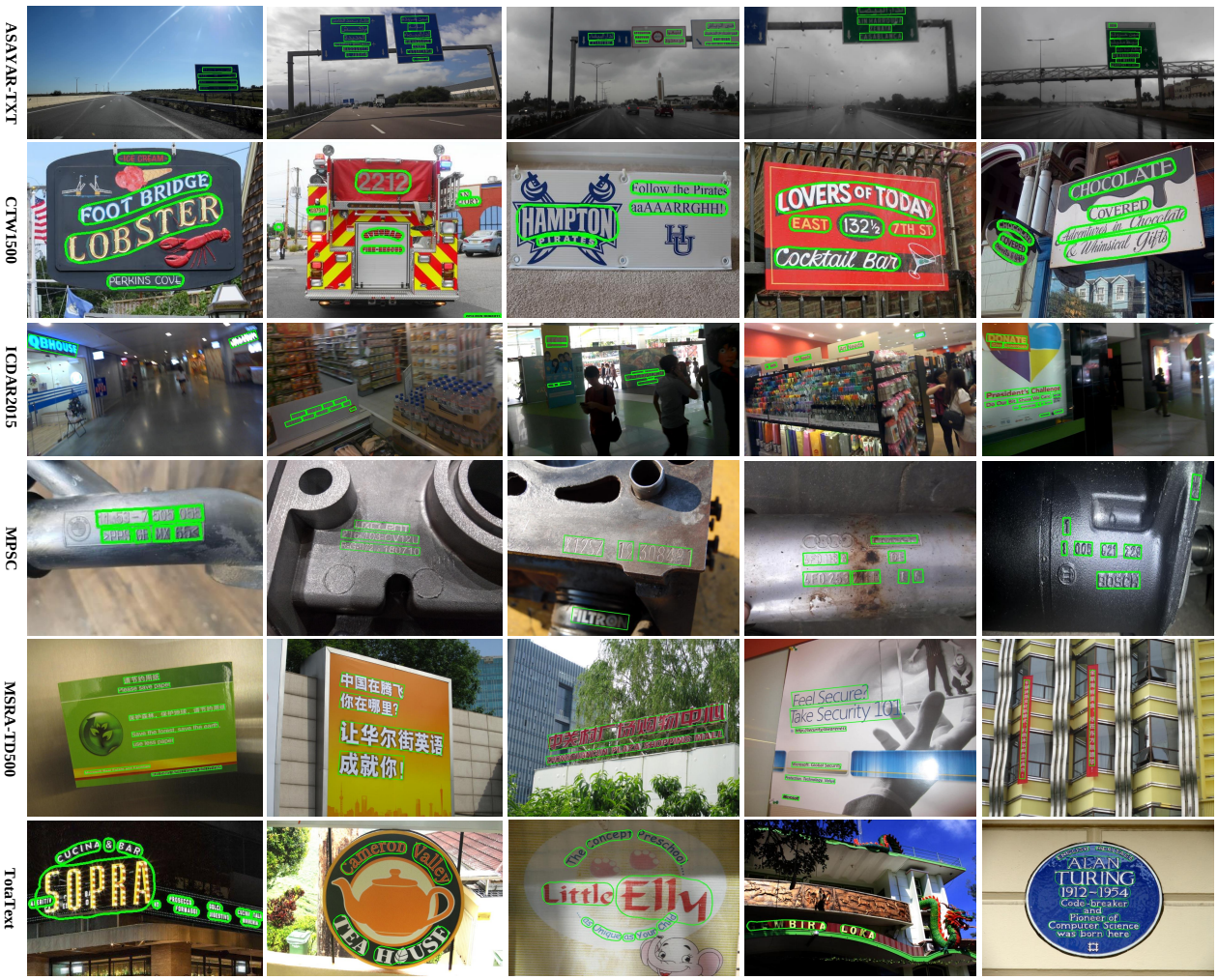


Fig. 8. The visualization results of some natural scene images are displayed. From top to bottom, the images are from ASAYAR-TXT, CTW1500, ICDAR2015, MPSC, MSRA-TD500, and TotalText, respectively.

is better than some non-real-time methods, such as LeafText [1], RP-Text [43], and KPN [44]. With the ResNet50 as the backbone, the proposed BOSPG achieves 88.5%, 87.6%, and 88.1% in terms of precision, recall, and F-measure, respectively. It surpasses most previous methods, such as ADNet [2], FS [36], KPN [44], and DBNet++ [5], SMNet [63].

5) Evalution on MPSC: Text in the industrial scene enjoys low contrast and corroded surfaces. To validate the effectiveness of the proposed BOSPG for detecting industrial text, experiments are conducted on the MPSC dataset. In the inference stage, the short side of the input images is resized to 800 pixels. As shown in Table VI, for existing state-of-the-art methods, ContourNet [67], RRPN++ [68], and RFN [15] achieve 84.3%, 85.3%, and 86.2% in F-measure. Benefiting that the proposed IEFGM encourages the module to extract instance-level features, the proposed BOSPG achieves 89.7%, 87.4%, and 88.5% in precision, recall, and F-measure when adopting ResNet50 as the backbone, which is superior to the existing SOTA method RFN [15].

6) Evalution on ASAYAR-TXT: The images in ASAYAR-TXT are from the traffic scene. As we can see from Table VII, the existing SOTA method HFENet achieves 96.3%, 97.2%, and 96.8% in precision, recall, and F-measure, respectively. Compared to it, the proposed method improves by 2.3%, 1.0%, and 1.6% in precision, recall, and F-measure, which achieves the best performance. The experiments demonstrate the effectiveness of the BOSPG in detecting traffic texts.

F. Visual Analysis

To further highlight the superiority of the proposed BOSPG, we illustrate the predictions of the shrink mask and instance map and the corresponding outputs in Fig. 7. In addition, to verify the generality of the proposed BOSPG, the detection results from the ASAYAR-TXT, CTW1500, ICDAR2015, MPSC, MSRA-TD500, and TotalText are visualized in Fig. 8, which contains curved texts and multi-oriented texts on natural, industrial, and traffic scenes. As shown in Fig. 9, under low lighting, the proposed model is able to detect text that is difficult for humans to identify. Specifically, the two targets in the upper-left and upper-right corners are challenging to detect with the naked eye. The proposed method successfully identifies them. After contrast enhancement, it is clear that text exists in these regions. This demonstrates the superiority of the proposed method under low lighting conditions. As for the

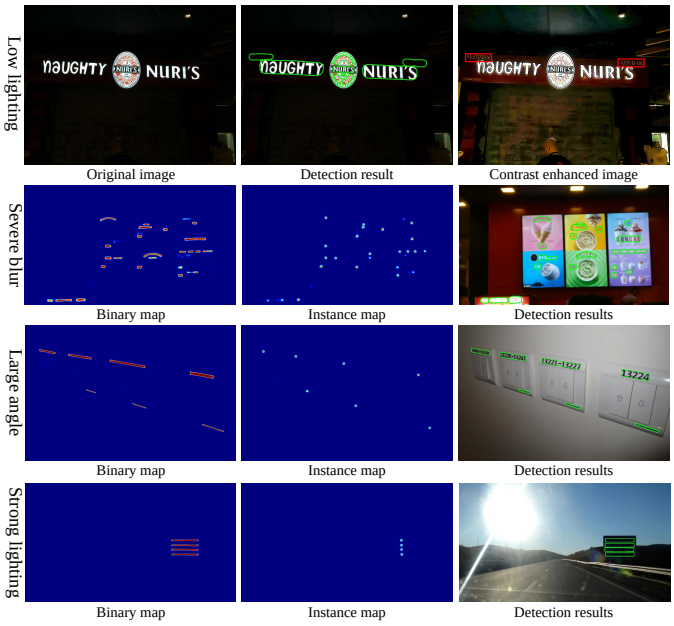


Fig. 9. Visualization under extreme scenarios, including low lighting, severe blur, large angle, and strong lighting.

TABLE VIII
THE CROSS-DATASET VALIDATION RESULTS ARE PERFORMED IN WORD-LEVEL AND LINE-LEVEL ANNOTATED DATASETS, WHERE IC15, TOTAL, CTW, AND TD500 REPRESENT ICDAR2015, TOTAL-TEXT, CTW1500, AND MSRA-TD500, RESPECTIVELY.

Train	Test	Method	P	R	F
IC15	Total	TextField [69]	61.5	65.2	63.3
		CM-Net [26]	75.8	64.5	69.7
		ZTD [27]	78.5	64.1	70.6
		BOSPG	85.0	74.9	**79.6**
Total	IC15	TextField [69]	77.1	66.0	71.1
		CM-Net [26]	76.5	68.1	72.1
		ZTD [27]	79.8	69.3	74.2
		BOSPG	84.0	73.3	**78.3**
TD500	CTW	TextField [69]	75.3	70.0	72.6
		CM-Net [26]	77.2	69.7	72.8
		ZTD [27]	84.1	73.4	78.4
		RSMTD [3]	82.7	74.3	78.3
CTW	TD500	BOSPG	83.8	74.5	**78.9**
		TextField [69]	85.3	75.8	80.3
		CM-Net [26]	85.8	77.1	81.2
		ZTD [27]	86.8	77.9	82.1
		RSMTD [3]	82.5	77.8	80.1
		BOSPG	89.3	88.5	**88.9**

blurred text, the BOSPG can detect some mildly blurred targets but struggles to effectively detect highly blurred ones. For images captured at large angles, our method detects most of the texts, with one target remaining undetected due to blurring and angle issues. Under strong lighting, the proposed BOSPG performs well in detecting scene text. These results validate the effectiveness and superiority of the proposed approach in extreme scenarios.

G. Cross Validation

To verify the generalization performance and data robustness of the BOSPG, we follow the ZTD [27] to conduct cross-dataset experiments. We use ICDAR2015, Total-Text, MSRA-TD500, and CTW1500 to perform experiments, which can be divided into word-level and line-level according to the annotation format. Initially, the model is trained on the ICDAR2015 and is utilized to conduct testing on the Total-Text. As listed in Table VIII, the proposed BOSPG achieves precision, recall, and F-measure of 85.0%, 74.9%, and 79.6%, which significantly outperforms previous methods, such as TextFields [69] (16.3%), CM-Net [26] (9.9%), and ZTD [27] (9.0%). After swapping the training and testing datasets, the proposed BOSPG achieves an F-measure of 78.3%, which is a competitive result and exceeds TextFields, CM-Net, and ZTD by 7.2%, 6.2%, and 4.1%, respectively. For line-level datasets, we train on the MSRA-TD500 and test on the CTW1500. The BOSPG achieves 83.8%, 74.5%, and 78.9% in precision, recall, and F-measure, which is superior to TextFields, CM-Net, RSMTD, and ZTD. Finally, the BOSPG is trained on the CTW1500 and is tested on the MSRA-TD500, which achieves 88.9% in F-measure. It not only outperforms TextField, CM-Net, and ZTD, but also is superior to some advanced methods directly using the MSRA-TD500 dataset for training, such as FS [36], DText [38], and LEMNet [41]. Overall, training on a dataset containing curved text and testing on another dataset yields better results compared to swapping them. This is because the multi-directional text dataset does not include samples of curved text. The above experiments and analysis show excellent generalization performance of the BOSPG.

V. CONCLUSION

In this paper, we propose BOSPG, an efficient and effective multi-scene text detector. It consists of an Instance Equal Feature Guide Module (IEFGM), a Cross-Level Feature Interaction Module (CLIFM), and a Pixel-Instance Fusion Discriminator (PIFD). The IEFGM encourages the module to treat samples of different scales equally at the feature level and incorporates both instance features and positional information to distinguish text from the background. The CLIFM facilitates the interaction between pixel-level and instance-level features, enabling the model to understand text properties from various perspectives. Unlike existing segmentation-based methods that rely solely on pixel-class scores to filter candidate instances, the PIFD combines pixel-level and instance-level predictions to classify samples as positive or negative. Extensive experiments demonstrate that BOSPG achieves SOTA performance across multiple public benchmarks while effectively balancing speed and accuracy. Ablation studies validate the effectiveness of the proposed modules. Future work will focus on developing a more efficient and adaptable text detector to handle diverse scenes.

REFERENCES

[1] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Text growing on leaf," *IEEE Transactions on Multimedia*, vol. 25, pp. 9029–9043, 2023.
[2] 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.
[3] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Reinforcement shrink-mask for text detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 6458–6470, 2023.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[11] 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.

[12] 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.

[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] M. He, M. Liao, Z. Yang, H. Zhong, J. Tang, W. Cheng, C. Yao, Y. Wang, and X. Bai, "Most: A multi-oriented scene text detector with localization refinement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8813–8822.

[15] T. Guan, C. Gu, C. Lu, J. Tu, Q. Feng, K. Wu, and X. Guan, "Industrial scene text detection with refined feature-attentive network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6073–6085, 2022.

[16] 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.

[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] 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.

[19] 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.

[20] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European conference on computer vision*, 2018, pp. 20–36.

[21] S. Zhang, X. Zhu, J. Hou, C. Liu, C. Yang, H. Wang, and X. Yin, "Deep relational reasoning graph network for arbitrary shape text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9699–9708.

[22] 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.

[23] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, and X. Bai, "Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping," *Pattern recognition*, vol. 96, p. 106954, 2019.

[24] 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.

[25] 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.

[26] 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.

[27] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Zoom text detector," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 15 745–15 757, 2024.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] M. Ye, J. Zhang, S. Zhao, J. Liu, B. Du, and D. Tao, "Dptext-detr: Towards better scene text detection with dynamic points in transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3241–3249.

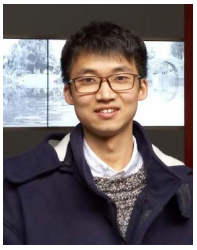
[46] C. Xu, W. Jia, R. Wang, X. Luo, and X. He, "Morphtext: Deep morphology regularized accurate arbitrary-shape scene text detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 4199–4212, 2023.

[47] 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.
[48] 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.
[49] 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.
[50] 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.
[51] 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.
[52] 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.
[53] 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.
[54] 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.
[55] 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.
[56] 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.
[57] 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.
[58] 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.
[59] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon *et al.*, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *2017 14th IAPR International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 2017, pp. 1454–1459.
[60] 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.
[61] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[62] 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.
[63] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Real-time text detection with similar mask in traffic, industrial, and natural scenes," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2024.
[64] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
[65] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.
[66] Y. Liu, T. He, H. Chen, X. Wang, C. Luo, S. Zhang, C. Shen, and L. Jin, "Exploring the capacity of an orderless box discretization network for multi-orientation scene text detection," *International Journal of Computer Vision*, vol. 129, pp. 1972–1992, 2021.
[67] Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 753–11 762.
[68] J. Ma, "Rrpn++: Guidance towards more accurate scene text detection," *arXiv preprint arXiv:2009.13118*, 2020.
[69] 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.

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.

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



Junyu Gao received the B.E. degree and the Ph.D. degree in computer science and technology from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2015 and 2021 respectively. He is currently an associate professor with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

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/).