Text-Pass Filter: An Efficient Scene Text Detector

Chuang Yang, Haozhao Ma, Xu Han, Yuan Yuan, *Senior Member, IEEE,* and Qi Wang, *Senior Member, IEEE*

***Abstract*—To pursue an efficient text assembling process, existing methods detect texts via the shrink-mask expansion strategy. However, the shrinking operation loses the visual features of text margins and confuses the foreground and background difference, which brings intrinsic limitations to recognize text features. We follow this issue and design Text-Pass Filter (TPF) for arbitrary-shaped text detection. It segments the whole text directly, which avoids the intrinsic limitations. It is noteworthy that different from previous whole text region-based methods, TPF can separate adhesive texts naturally without complex decoding or post-processing processes, which makes it possible for real-time text detection. Concretely, we find that the band-pass filter allows through components in a specified band of frequencies, called its passband but blocks components with frequencies above or below this band. It provides a natural idea for extracting whole texts separately. By simulating the band-pass filter, TPF constructs a unique feature-filter pair for each text. In the inference stage, every filter extracts the corresponding matched text by passing its pass-feature and blocking other features. Meanwhile, considering the large aspect ratio problem of ribbon-like texts makes it hard to recognize texts wholly, a Reinforcement Ensemble Unit (REU) is designed to enhance the feature consistency of the same text and to enlarge the filter's recognition field to help recognize whole texts. Furthermore, a Foreground Prior Unit (FPU) is introduced to encourage TPF to discriminate the difference between the foreground and background, which improves the feature-filter pair quality. Experiments demonstrate the effectiveness of REU and FPU while showing the TPF's superiority.**

***Index Terms*—Scene text detection, irregular-shaped text, computer vision, real-time detector**

I. Introduction

SCENE text understanding [1], [2], [3], [4] is a hot topic in computer vision, which serves as a fundamental task in many practical applications (such as unmanned systems, bionic robots, and cognitive domain security defense). As an essential research branch of scene text understanding, scene text detection [5], [6], [7] is responsible for extracting the text regions from images. In this paper, we aim for real-time arbitrary-shaped text detection [8], [9], [10] from scenes.

With the rapid development of deep learning and recent advances made in image segmentation [11], [12], text detection [13], [14] achieves remarkable progress. Considering the various and complex shapes of scene texts, extracting the text regions from images via segmentation technology becomes a hot branch in the research of scene text detection.

This work was supported by the National Natural Science Foundation of China under Grant 62501511, 62471394, and U21B2041.

Chuang Yang, Haozhao Ma, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China.

Xu 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, Shaanxi, P. R. China.

E-mail: cyang113@mail.nwpu.edu.cn, haozhaoma@mail.nwpu.edu.cn, hxu04100@gmail.com, y.yuan.ieee@gmail.com, crabwq@gmail.com.

Qi Wang is the corresponding author.

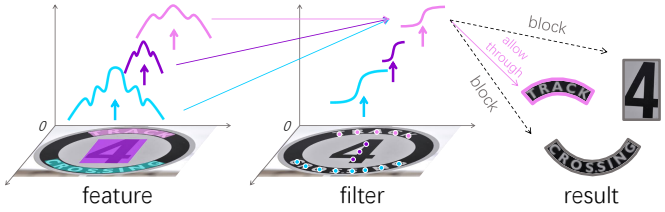


Fig. 1. Motivation of the designed text-pass filter (TPF). It constructs a unique feature-filter pair for each text, where every feature represents a unique text. Each filter can only allow through the corresponding unique text feature and block the others, which helps detect every text separately in an efficient way.

Existing segmentation-based methods can be categorized into whole region-based approaches [15], [4], [16], [17] and shrink-mask-based algorithms [18], [19], [20], [21]. Whole region-based methods aim to segment entire text regions and directly extract the corresponding mask contours as final results. However, to separate overlapping texts, these methods often require complex decoding or post-processing steps to reconstruct text contours, deviating from their original design and introducing additional computational costs. In contrast, shrink-mask-based methods first roughly locate texts using shrink masks and then expand them to rebuild text contours, which streamlines the text assembly process. Nevertheless, the shrinking operation compromises the semantic integrity of the text and confuses the distinction between foreground and background, imposing intrinsic limitations on these methods' ability to recognize text features effectively.

To remedy the problems that exist in the above two kinds of methods simultaneously, we propose an efficient framework in this work for accurate text detection with high inference speed. Specifically, inspired by the band-pass filter in electronics and signal processing, we find the operation mode that allows through components in a specified band of frequencies but blocks components with frequencies above or below this band is suitable for extracting every single text separately and simply. Based on this observation, the text-pass filter (TPF) is designed to extract texts from scenes.

TPF is constructed as an end-to-end convolutional neural network, which ensures straightforward training and inference processes. In the training stage, it encodes a unique feature-filter pair for every single text, where the feature and filter are treated as the passband and the band-pass filter, called pass-feature and text-pass filter, respectively. Different from shrink-mask-based methods, TPF focuses on whole text regions. It helps define a distinct semantic boundary between the text and the background, which enhances the recognition of text features. In the inference stage, TPF first combines all filters as a sieve and inputs the feature map that contains every text feature into it. All filters in the sieve then recognize their unique pass-feature in parallel to extract the corresponding text region.

The efficient filtering process helps TPF separate adhesive texts without extra decoding or post-processing processes that exist in previous whole text region-based frameworks, which provides significant improvements in detection speed.

Meanwhile, different from normal objects, the ribbon-like text shape leads to the large aspect ratio problem. It further results in the feature inconsistency problem for the same text and the filter's limited recognition field problem, which makes filters are difficult to recognize the whole text region accurately. Considering the above problem, we design a Reinforcement Ensemble Unit (REU). It first measures the feature similarities of the same text and the feature differences with the others to enhance the feature consistency. Then, multiple filters of the same text are integrated to generate a strengthened filter according to feature similarities to enlarge the filter's recognition field to encourage the model to recognize whole text regions effectively. Furthermore, considering the feature-filter pair is extracted from the predicted text region, which makes the quality of the filter-feature pair deep relies on the model's ability to discriminate texts from the background, a Foreground Prior Unit (FPU) is introduced to encourage TPF to recognize the difference between the foreground and background. It helps the proposed efficient framework locate text instances more accurately. The contributions of this paper are summarized as follows:

1) Inspired by the band-pass filter (BPF) in electronics and signal processing, a text-pass filter (TPF) is proposed to formulate the text detection problem. It simulates the BPF operation mode to segment the whole region of each text directly and individually, which helps avoid the limitation of shrink-mask-based methods while making TPF enjoy a more efficient pipeline than existing whole region-based methods to separate adhesive texts.
2) A Reinforcement Ensemble Unit (REU) is designed to measure the feature similarities of the same text and the feature differences with the others to enhance text feature consistency. Meanwhile, it integrates multiple filters of the same text according to feature similarities to generate a strengthened filter, which enlarges the filter's recognition field to encourage TPF to recognize whole text regions effectively.
3) A Foreground Prior Unit (FPU) is introduced to discriminate the difference between the foreground and the background. It helps TPF to locate text instances more accurately, which encourages generating feature-filter pairs with high quality and suppressing false positives.

The rest of the paper is organized as follows. Section II introduces the previous works on text detection. Section III describes the overall structure of TPF. The experimental results are discussed in Section IV. Section V concludes the paper.

II. Related Work

Deep learning-based text detection methods [22], [23], [24], [25], [26] have achieved significant progress recently, which can be divided into whole region-based methods and shrink-mask-based methods roughly. They will be introduced next.

A. Whole Text Region-based Methods

Researchers detect whole text regions and achieve superior performance initially [27] based on the detection framework [28], [29], [30], [31]. Zhou *et al.* [32] and He *et al.* [33] followed the idea of Densebox to locate the text center and to regress the offsets between the center and four vertices for reconstructing text boxes. Different from the above methods, Liao *et al.* [34], [35] proposed to regress the offsets between anchor vertices and text box vertices. To extract strong representative features of multi-oriented texts, Liao *et al.* [36] introduced active rotating filters to encode direction information of texts for enhancing rotation-invariant features. Recent research focus has shifted to more challenging irregular-shaped text detection. Some works [37], [38], [39] intuitively segment whole regions of texts to locate them directly based on segmentation methods [11], [12]. However, for separating texts that lie close to each other, Deng *et al.* [37] and Xu *et al.* [38] proposed to encode the text direction information and assigned the pixels within text margins to the corresponding texts according to the information. Except for segmenting all pixels in whole text regions, many researches [17], [15], [40], [4] design effective text representations to predict whole text contours. The works [41], [42], [43] and Wang *et al.* [44] extracted a series of dense contour points via the regression and segmentation strategy to rebuild text contours, respectively. Zhu *et al.* [16] were inspired by Fourier transformation and proposed to transform the text contour into compact signatures. Su *et al.* [4] encoded whole regions of texts into compact vectors via discrete cosine transformation. Although these methods have achieved superior performance for arbitrary-shaped scene text detection, they have to introduce the complex decoder or post-processing for separating adhesive texts.

B. Shrink-Mask-based Methods

To simplify the detection pipeline and speed up the inference speed for pursuing real-time requirements of practical applications, researchers design a shrink-mask expansion strategy to extract texts. Wang *et al.* [45] first predicted multiple shrink-masks with different shrunk scales and whole regions via segmentation methods directly. All pixels within larger shrink-masks then were assigned to smaller ones step by step until whole regions become the aforementioned larger ones. Considering the low efficiency of the stepwise expansion process, Wang *et al.* [46], [47] predicted shrink-masks with one specific scale, and whole regions then were clustered into shrink-masks according to the idea of the clustering algorithm. Different from the above methods that extra predicted whole regions, Liao *et al.* [48], [20] could reconstruct text contours according to shrink-masks merely. They computed an expansion offset via the area and perimeter of shrink-masks and expanded shrink-mask contours by the offset to rebuild text contours. Yang *et al.* [19] observed the strategy that generating shrink-masks via the area and perimeter fail to represent hourglass texts. The authors presented to shrink the text masks according to the text shapes for enhancing the model's fitting ability effectively. Yang *et al.* [21] found the detection results of the method [48] rely on the accuracy of

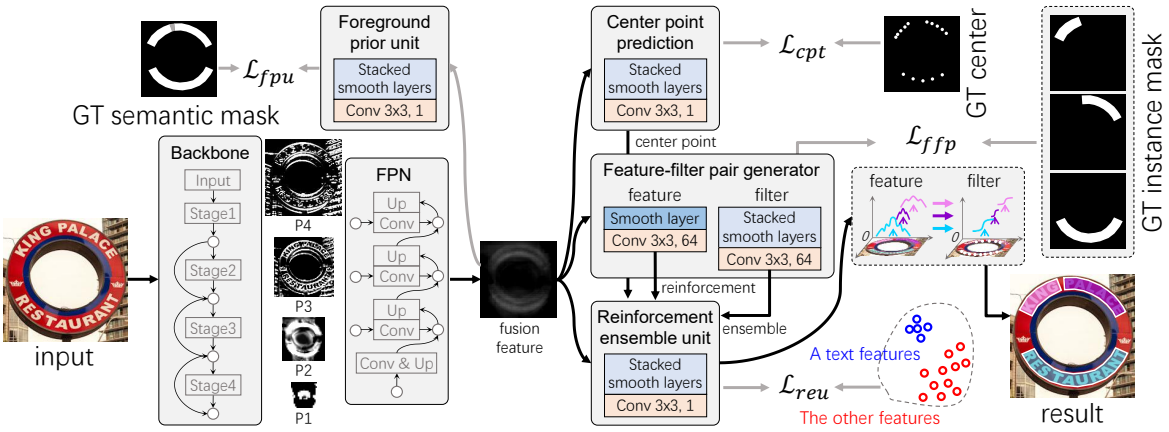


Fig. 2. Overall architecture of the proposed TPF. It consists of a feature extractor, center point prediction header, feature-filter pair generator, Reinforcement Ensemble Unit (REU), and Foreground Prior Unit (FPU). The extractor includes the backbone and FPN and the corresponding output is a concatenated feature map with the size of $\frac{H}{4}, \frac{W}{4}$. The center point prediction header is responsible for locating text instances. REU is designed for encouraging the feature-filter pair generator to produce feature-filter pair with high quality for each text. FPU is introduced to help recognize text center points more accurately. The black flow and gray flow illustrate the forward and backward propagation of the whole training process. Particularly, the gray arrows are the inference-only operators. They bring no extra computational cost for the testing process.

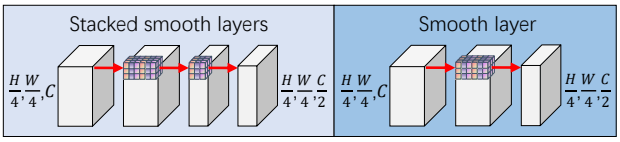


Fig. 3. Structure details of stacked smooth layers and smooth layer. They are composed of 3×3 convolution al layers mainly.

predicted shrink-masks deeply, which leads to sensitive rebuilt text contours. They proposed to encode the shrink-mask and expansion offset separately for pursuing robust reconstructed results. Different from the aforementioned methods, Yang *et al.* [18] constructed an efficient encoder to further improve the detection speed while ensuring the strong representativity of the extracted features. Though the shrink-mask expansion strategy helps separate adhesive texts efficiently, the broken semantic integrity and the feature confusion between the foreground and background bring intrinsic limitations for these methods to recognize text features accurately.

III. Methodology

This section first introduces the overall architecture of the network that is constructed based on the designed text-pass filter (TPF). The structure and operation detail of Reinforcement Ensemble Unit (REU) then is illustrated. Next, Foreground Prior Unit (FPU) is described. In the end, the loss function used for supervising whole network is formulated.

A. Overall Architecture

The overall architecture of the proposed TPF is illustrated in Fig. 2. Following the design of traditional feature extractor [49], [50], TPF first extracts strong representative features by the combination of backbone and feature pyramid network (FPN). It adopts ResNet as the backbone to generate multiple-sized feature maps (including the $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ input image sizes), where smaller feature maps help our model distinguish texts from the background and the larger ones encourage TPF focus on text details. Different from normal objects, considering the large aspect ratio of the text instance, the FPN is constructed behind the backbone to extract a fusion feature with a $\frac{1}{4}$ image size for detecting texts, which helps improve the recall performance of detection results.

The fusion feature is then fed into the center point prediction header, feature-filter pair generator, and reinforcement ensemble unit (REU) for segmenting text masks. Specifically, for the center point prediction header, it consists of stacked smooth layers and one convolution layer with a 1×1 kernel. The stacked smooth layers (the detail can be found in Fig. 3) are composed of two 3×3 convolution layers with C and $\frac{C}{2}$ channels respectively, which are used for smoothing the gap between fusion features and the final output center point map. Inputting the fusion feature into the prediction header, it is smoothed from C channels to $\frac{C}{2}$ channels and a $\frac{1}{4}$ image-sized center point map is segmented for locating texts. The feature-filter pair generator includes two branches for generating a text feature map and constructing the corresponding filter map, respectively. The two branches enjoy a similar structure with the center point prediction header except that the predicted maps are with 64 channels. By combining the feature and filter maps and the predicted center point locations, the text features and filters can be extracted.

Normally, instance masks can be obtained separately by simulating the band-pass filter operation mode that the filters allow through their unique pass-feature and block other features. However, the feature difference problem and the filter local-sensitive problem brought by the large aspect ratio make it hard to segment whole text masks directly. Considering the above problems, the REU takes the sampled text features and filters as input to enhance the feature consistency and filter reliability for encouraging the model to recognize whole text regions (details are described in Section III-B and Fig. 3). The processed text filters are combined as a filter sieve, where all filters allow through their unique pass-features to extract all

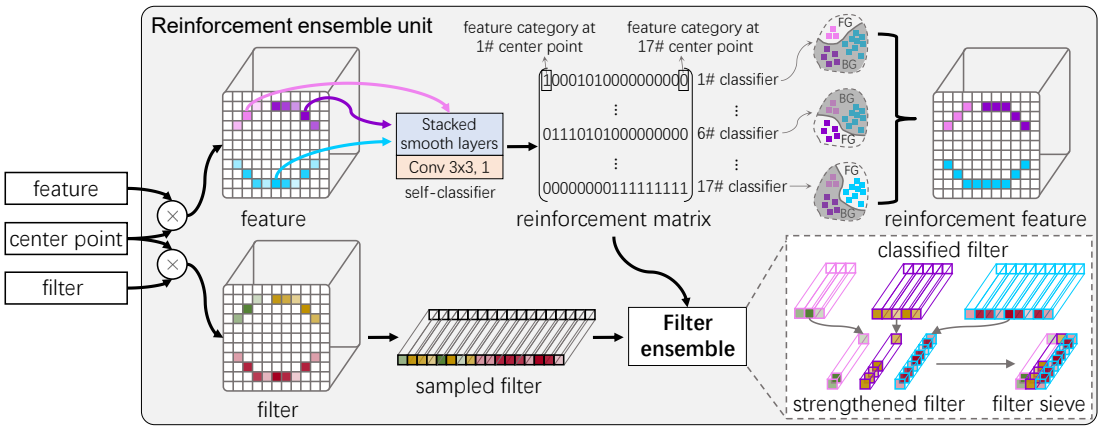


Fig. 4. Structure details of the designed REU. The feature and filter are the outputted feature map and filter map from the feature-filter pair generator. The center point denotes the point coordinates predicted by the center point prediction header. The text features and filters first are sampled from the feature map and filter map according to center point coordinates. The feature consistency and filter recognition ability then are enhanced and strengthened under the guidance of the reinforcement matrix, respectively. In the end, the strengthened filters are combined as a filter sieve by the filter ensemble algorithm (as illustrated in Algorithm 1) for detecting texts efficiently.

text masks in parallel.

B. Reinforcement Ensemble Unit

Considering the large aspect ratio problem of the text results in the feature inconsistency problem and the filter's limited recognition field problem for the same text, which makes it hard for filters to recognize the whole text features effectively, REU is designed for enhancing feature consistency of the same text while enlarging the filter's recognition field.

Algorithm 1 Filter Ensemble

Require: The sampled filters $\mathbf{F}_{fi}$ and reinforcement matrix $\mathbf{M}$;
Ensure: The ensemble filter sieve $\mathbf{F}_{fis}$;

$\mathbf{F}_{fis} \leftarrow [\,]$
$v \leftarrow$ sum($\mathbf{M}$, dim=0)
$v_{sort} \leftarrow$ sort(v, ascending=False)
for u in unique(v_{sort}) **do**
  $idx \leftarrow$ argwhere(v==u)
  $\mathbf{M}_u \leftarrow \mathbf{M}[\mathbf{M}[:, idx]]$
  $\mathbf{M}_c \leftarrow \mathbf{M}_u[0]$
  for r in range(1,$\mathbf{M}_u$.shape[0]) **do**
    $\mathbf{M}_c \leftarrow \mathbf{M}_c \times \mathbf{M}_u[r]$
  end for
  $f_c \leftarrow \mathbf{F}_{fi}$[argwhere($\mathbf{M}_c$==1)] //$f_c$ is classified filter
  $f_{str} \leftarrow$mean(f_c) //f_{str} denotes strengthened filter
  $\mathbf{F}_{fis} \leftarrow f_{str}$
end for
$\mathbf{F}_{fis} \leftarrow$ concate($\mathbf{F}_{fis}$, axis=1) //$\mathbf{F}_{fis}$ is filter sieve

For **enhancing feature consistency** of the same text, as shown in Fig. 4, REU takes center point coordinates and feature map as input. It first extracts sampled text features $\mathbf{F}_{fe} \in \mathbb{R}^{n\times 64}$ from the feature map according to the n center point coordinates, where each feature vector $f_{fe} \in \mathbb{R}^{1\times 64}$ represents a part of text features.

REU then constructs a self-classifier for assigning n feature vectors to different text instances, where the self-classifier is implemented as a combination of a stacked smoothing layer and a convolutional layer featuring a 3×3 kernel with single-channel output, and the number of self-classifier output changed dynamically according to the predicted text mask. For example, given n center points, the classifier sequentially treats each point as a reference and predicts whether all other points belong to the same text instance as this reference (outputting 0 or 1 for each prediction), ultimately generating a rank n reinforcement matrix. The output dimensions of this process are entirely determined by the number of center points. Concretely, in the inference stage, it predicts a reinforcement matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$, where each row of M represents the binary classification results of the self-classifier to all f_{fe}. For rth row of M, the cth column binary value represents that the self-classifier assigns cth f_{fe} and rth f_{fe} into one text instance if the corresponding value is 1. The self-classifier operation forces the feature vectors that belong to the same text instance to enjoy strong similarities with each other. Meanwhile, it strengthens the feature vector differences when they belong to different texts. The above advantages of RUE help filters avoid segmenting the same text as multiple ones and improve the precision performance of detection results effectively.

For **enlarging filter's recognition field**, REU takes the same sampling process as the text features. It first extracts sampled text filters $\mathbf{F}_{fi} \in \mathbb{R}^{n\times 64}$ by the combination of n center point coordinates and filter map, where $\mathbf{F}_{fi}$ consists of n filters $f_{fi} \in \mathbb{R}^{1\times 64}$. All filters are then strengthened and combined as a filter sieve $\mathbf{F}_{fis} \in \mathbb{R}^{m\times 64}$ for segmenting all whole text masks in parallel, where m denotes the number of text instances of the input image. Specifically, as illustrated in Algorithm 1, given the sampled filters $\mathbf{F}_{fi}$ and the reinforcement matrix $\mathbf{M}$ that is predicted according to the sampled text features $\mathbf{F}_{fe}$, the filter ensemble algorithm generates the filter sieve $\mathbf{F}_{fis}$ through the following four steps mainly: (1) performing an add operation along the column direction of the matrix $\mathbf{M}$ to obtain a filter importance sequence $v \in \mathbb{R}^{n\times 1}$; (2) sorting the sequence v in descending order to generate a priority sequence $v_{sort} \in \mathbb{R}^{n\times 1}$; (3)

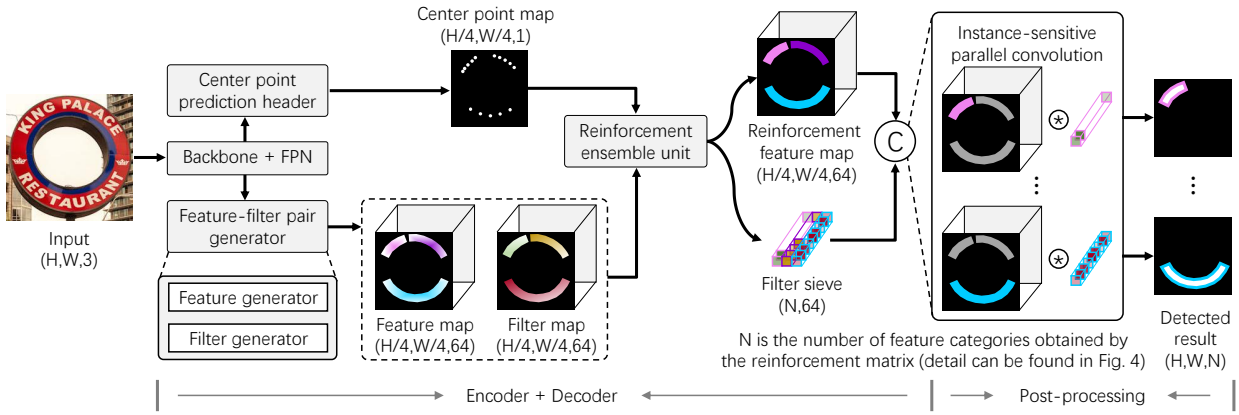


Fig. 5. Visualization of inference process. Structure details of center point prediction header, feature-filter pair generator, REU, and FPU can be found in Fig. 2, Fig. 3, and Fig. 4.

determining the maximum value corresponding column index idx_{max} in v; (4) extracting the rows from $\mathbf{M}$ according to the index idx_{max}; (5) intersecting those rows from M to obtain a strengthened row and extracting each location on the row corresponding filter for collecting all filters that belong to the same text instance at first. Then, merging those filters via the mean value method to generate the final strengthened filter f_{str}, which is responsible for filtering the matched text; (6) repeating (1)–(5) steps to produce m strengthened filters and combined them as a filter sieve $\mathbf{F}_{fis}$. The strengthened filter helps enlarge the recognition field of the sampled filter and the filter sieve accelerates the inference speed significantly.

C. Foreground Prior Unit

As illustrated in Fig. 2 and Fig. 4, TPF extracts feature-filter pair from the feature and filter maps according to the predicted center point coordinates, which leads to the feature-filter pair quality deep relies on the model's foreground recognition ability. To help TPF to locate text instances accurately, FPU is introduced to encourage foreground discrimination from the background under the guidance of prior semantic information.

Considering the center point prediction header, feature-filter pair generator, and REU rely on the fusion feature, FPU takes it as input to strengthen the corresponding semantic feature. FPU first smooths the fusion feature via a shallow smooth layer, which helps fuse high-level and low-level features while ensuring an effective backward propagation of semantic information to the fusion feature. FPU then generates a binary mask through a 3×3 convolutional layer to learn the feature difference between the foreground and background by supervising the predicted mask with GT semantic mask.

D. Inference Process

We have introduced the overall architecture of TPF in Section III-A and the details of the proposed REU and FPU in Section III-B and III-C. To show the efficiency of our method, the inference pipeline is illustrated in this section.

Concretely, it is shown in Fig. 5, given an input image, the binary mask map of center point, text feature map, and text filter map can be obtained from center point prediction header and feature filter pair generator at first. To remedy the scene text large aspect ratio problem, the three maps then are fed into REU to enhance the consistency of feature map and enlarge the recognition field of filter map according to reinforcement matrix $\mathbf{M}$ (the process can be referred in Section III-B). Following the motivation of TPF, the reinforcement feature map and filter sieve generated from REU are treated as the text pass-feature and text pass-filter, respectively. In the end, every whole region of text instance can be obtained via the instance-sensitive parallel convolution-based post-processing. Specifically, since each text pass-filter is sensitive to the corresponding text pass-feature only, all filters can recognize their pass-features in parallel to detect all texts separately without the interference of the adhesion problem. It ensures efficient post-processing and makes TPF runs faster than existing whole region-based text detection methods.

E. Label Generation

As illustrated in Fig. 2, the proposed network includes four prediction branches (FPU, center point prediction header, feature-filter pair generator, and REU). This section describes the corresponding label generation process in detail.

For **feature-filter pair generator**, it is designed for generating feature-filter pair for each text instance. To effectively supervise this branch in the training process, we extract all text instance masks from a binary mask one by one and combines them to obtain the corresponding label $\mathbf{F}_{ins} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times m}$, where m denotes the number of text instances of the input image. H and W are the height and width, respectively.

For **center point prediction header**, this branch is responsible for locating texts. The outputted center point coordinates are used for obtaining the feature-filter pair of the text instance from the predicted feature and filter maps from the feature-filter pair generator in the inference process. Given a $\mathbf{F}_{ins}$, we sample center points from every text instance mask by the sampling process in Fig. 6. Concretely, for a specific text instance, the process first computes its height h and width w along the y-axis and x-axis directions simultaneously. Then, the x-axis direction is determined as the sampling direction if $w > h$ else performing the sampling process along the

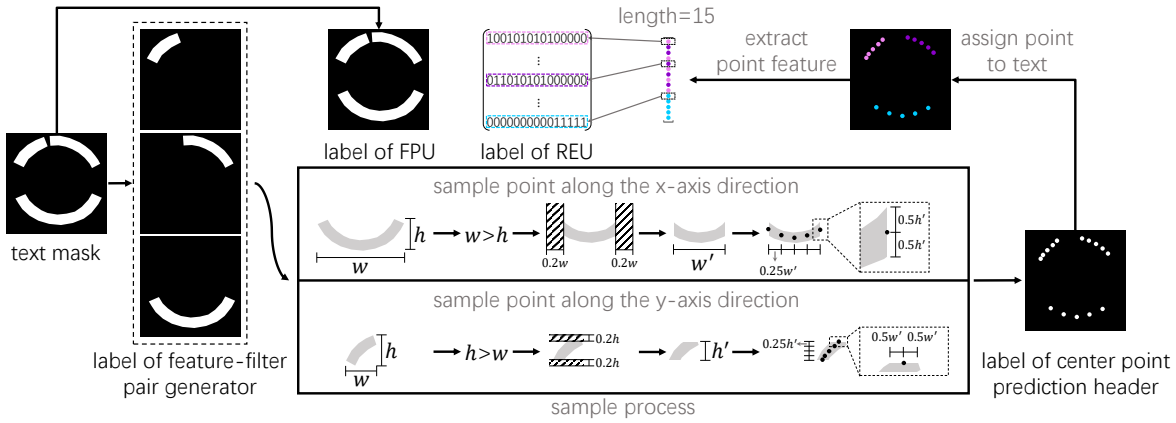


Fig. 6. Visualization of label generation process. For the four branches of center point prediction header, feature-filter pair generator, REU, and FPU, the corresponding label are shown in detail.

y-axis direction. Next, the middle part of the text instance is extracted as the valid sampling area. In the end, n center points are sampled from the valid area along the determined axis direction through equidistant sampling process. The sampled points of all texts are integrated into one mask for generating the final center point label.

For **Reinforcement Ensemble Unit**, we introduce this structure for enhancing the feature consistency of the same text while strengthening the filter's recognition ability (as illustrated in Section III-B and Fig. 4). The label of this branch can be obtained by combining the labels of feature-filter pair generator and center point prediction header. Specifically, all center points are first assigned to different text instances according to $\mathbf{F}_{ins}$, and then a matrix label $\mathbf{M} \in \mathbb{R}^{n\times n}$ is generated based on the point coordinates and point assignment situations. $\mathbf{M}$ is a binary matrix, r th row of $\mathbf{M}$ represents the classification results of r th self-classifier to all points and c th column of $\mathbf{M}$ denotes the classification results of all self-classifiers to c th points ($0 < r < n, 0 < c < n$). For r th row of $\mathbf{M}$, the value of c column being '1' if the c th point and r th point belong to the same text instance else '0'.

For **Foreground Prior Unit**, it is proposed to guide our method to distinguish text instances from the background more accurately. The text binary mask is adopted as the label of FPU. As shown in Fig. 6, all texts are drawn in the mask. The foreground and background are labeled as '1' and '0'.

F. Loss Function

The proposed TPF generates feature-filter pair for each text instance in the training process and extracts all text masks separately by the pass-feature recognition of the filter in the inference process. To predict the feature-filter pair efficiently, FPU is introduced to help our model recognize text instances, which encourages TPF to locate text center points more accurately. Meanwhile, REU is constructed to enhance the filter recognition ability and feature consistency, which improves the quality of the text mask generated by the feature-filter pair while simplifying post-processing.

For supervising the four branches that exist in TPF effectively, we formulate the loss function $\mathcal{L}$ as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{fpu} + \beta\mathcal{L}_{cpt} + \mu\mathcal{L}_{ffp} + \lambda\mathcal{L}_{reu}, \tag{1}$$

where $\mathcal{L}_{fpu}$, $\mathcal{L}_{cpt}$, $\mathcal{L}_{ffp}$, and $\mathcal{L}_{reu}$ are loss functions for supervising FPU, center point prediction header, feature-filter pair generator, and REU, respectively. α, β, μ, and λ are the corresponding importance weights of them.

For $\mathcal{L}_{cpt}$ and $\mathcal{L}_{reu}$, they are responsible for evaluating the accuracy of the predicted center points and reinforcement matrix. As illustrated in Section III-E and Fig. 6, the label of the center point and reinforcement matrix enjoy the characteristics of sparsity, discontinuity, and imbalance between positive and negative samples. Cross entropy (CE) loss is proposed for binary classification tasks initially, but it performs badly when handling the sample imbalance. Considering the deficiency of CE loss, focal loss $\mathcal{L}_{fl}$ [51] is designed based on CE loss. It forces the model to focus on the positive samples by reducing the weights of negative samples, which is suitable for supervising the center point prediction header and REU:

$$\begin{aligned} \mathcal{L}_{fl}(p_t) &= -(1-p_t)^{\gamma} log(p_t), \\ p_t &= \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases}. \end{aligned} \tag{2}$$

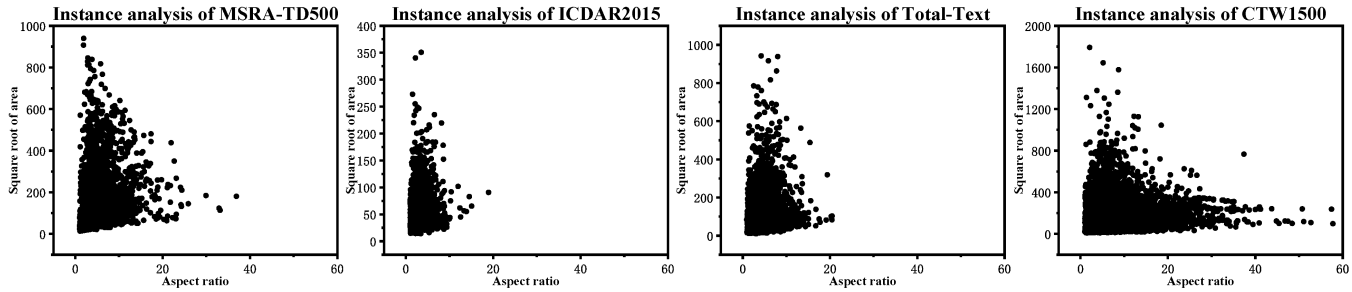
Considering the superiority of $\mathcal{L}_{fl}$, we formulate $\mathcal{L}_{cpt}$ and $\mathcal{L}_{reu}$ through it as follows:

$$\begin{aligned} \mathcal{L}_{cpt} &= \mathcal{L}_{fl}(p_{cpt}), \\ \mathcal{L}_{reu} &= \mathcal{L}_{fl}(p_{reu}), \end{aligned} \tag{3}$$

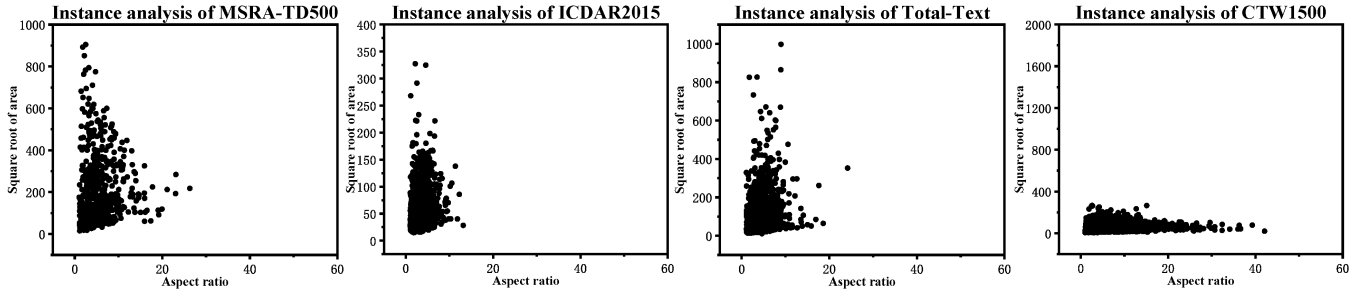
where p_{cpt} and p_{reu} are the predicted probability values of the center point and reinforcement matrix element.

Since the label of FPU and feature-filter pair generator are continuous, we construct the corresponding loss function $\mathcal{L}_{fpu}$ and $\mathcal{L}_{ffp}$ by dice loss [52]:

$$\begin{aligned} \mathcal{L}_{dl}(\mathbf{P}, \widehat{\mathbf{P}}) &= 1 - \frac{2 \times |\mathbf{P} \cap \widehat{\mathbf{P}}| + \varepsilon}{|\mathbf{P}| + |\widehat{\mathbf{P}}| + \varepsilon}, \\ \mathcal{L}_{fpu} &= \mathcal{L}_{dl}(\mathbf{P}_{fpu}, \widehat{\mathbf{P}}_{fpu}), \\ \mathcal{L}_{ffp} &= \mathcal{L}_{dl}(\mathbf{P}_{ffp}, \widehat{\mathbf{P}}_{ffp}), \end{aligned} \tag{4}$$



(a) Instance geometry characteristics analysis of training dataset



(b) Instance geometry characteristics analysis of testing dataset

Fig. 7. Analysis of geometry characteristics of text instances on multiple public datasets (MSRA-TD500, CTW1500, Total-Text, and ICDAR2015). (a) and (b) are the training and testing dataset, respectively.

where $\mathbf{P}_{fpu}$ and $\mathbf{P}_{ffp}$ are the predicted foreground mask and reinforcement matrix. $\widehat{\mathbf{P}}_{fpu}$ and $\widehat{\mathbf{P}}_{ffp}$ are the corresponding label (generation process can be referred in Fig. 6).

The center point prediction header, feature-filter pair generator, and REU participate in the process of segmenting text masks directly. Different from them, FPU is introduced as an assistant module to help the above three branches recognize texts more accurately. Considering the different importance of the four branches, we set α, β, μ, and λ as 0.7, 1, 1, and 1, respectively in the following experiments of this paper.

IV. Experiments

TPF is designed for detecting arbitrary-shaped scene texts efficiently. In this section, the effectiveness of the proposed approach and its enhancement modules first are verified in the ablation study. The superior comprehensive performance of TPF then is shown by comparing it with existing methods.

A. Datasets

SynthText [53] provides 800 thousand scene text images and 8 million text instances to help pre-train detection networks for improving the model's generalization ability, where each image is synthesized manually by placing multiple randomly generated texts in a natural scene image.

MSRA-TD500 [54] consists of line-level multi-oriented texts, which include English and Chinese simultaneously. It has 300 training and 200 testing images that are sampled from indoors. Considering the fewer training samples, we follow previous works to introduce 400 images from HUST-TR400 [55] to build the training dataset.

ICDAR2015 [56] is proposed in the Incidental Scene Text 2015 challenge of the Robust Reading Competition. It assigns 1000 images and 500 images for training and testing the model, respectively. The multi-scaled text instances and the complex image background bring challenges for text detection.

Total-Text [57] includes 1255 training images and 300 testing images. Different from the line-live text instances, the word-level texts in this dataset can demonstrate the model's superior ability to separate adhesive texts effectively.

CTW1500 is constructed by 1500 scene images, where 1000 images are used for training models and 500 images are responsible for testing models. The line-level curved text instances of this dataset can evaluate the model's ability for detecting irregular-shaped scene texts integrally.

We will demonstrate the superiority of our TPF on the above public scene text detection datasets next. For well analyzing the model performance to texts with different scales, aspect ratios, and shapes, we visualize the text instance information in Fig. 7. It can be found that the text instances of the training and testing data enjoy a similar scale and aspect ratio distribution for MSRA-TD500, Total-Text, and ICDAR2015 datasets, which is beneficial for evaluating model performance without interference. For the CTW1500 dataset, the training sample scale is larger than the testing sample scale a lot. The performance of the model that is trained and tested on samples with different scales can be explored on this dataset. Moreover, the line-level datasets (MSRA-TD500 and CTW1500) enjoy a larger aspect ratio than word-level datasets (Total-Text and ICDAR2015). The model's ability for dealing with samples with different aspect ratios can be analyzed by the cross-evaluation experiments on these datasets.

B. Implementation Details

The network architecture of the designed TPF is shown in Fig. 2, which consists of the backbone, FPN, and four

TABLE I
PERFORMANCE ANALYSIS OF THE MODELS WITH DIFFERENT SETTINGS ON THE MSRA-TD500 BENCHMARK. THE "SHORT SIDE: 736" DENOTES THE TESTING IMAGE IS RESIZED TO THE 736 PROPORTIONALLY ALONG ITS SHORT SIDE. THE MODEL IS NOT PRE-TRAINED ON ANY PUBLIC DATASETS.

Image size for testing (short side: 736)							
#	Methods	REU	FPU	Precision(%) ↑	Recall(%) ↑	F-measure(%) ↑	FPS↑
1	baseline	✗	✗	87.9	79.2	83.3	33.6
2	baseline+	✔	✗	89.7 (**1.8**↑)	80.7 (**1.5**↑)	85.0 (**1.7**↑)	36.2 (**2.6**↑)
3	baseline+	✔	✔	89.9 (**0.2**↑)	82.8 (**2.1**↑)	86.2 (**1.2**↑)	37.7 (**1.5**↑)

TABLE II
COMPUTATIONAL EFFICIENCY ANALYSIS OF THE MODELS WITH DIFFERENT SETTINGS ON THE MSRA-TD500 BENCHMARK. "ED" AND "POST" INDICATES THE TWO SPECIFIC SUB-PIPELINES (ENCODER+DECODER AND POST-PROCESSING) OF TPF IN THE INFERENCE PROCESS (DETAILS CAN BE REFERRED IN FIG. 5). "OVERALL GAIN" DENOTES THE OVERALL TIME COST GAINS BROUGHT BY "ED" AND "POST". THE MODEL IS NOT PRE-TRAINED ON ANY PUBLIC DATASETS.

Methods	Params(M)↓	GFLOPs↓	Time cost(ms)↓		
			ED	Post	Overall gain
baseline	13.0	102.7	21.5	8.3	–
baseline+REU	13.4 (**0.4**↑)	119.6 (**16.9**↑)	23.5 (**2.0**↑)	4.1 (**4.2**↓)	**2.2**↓
baseline+REU+FPU	13.4 (0.0–)	119.6 (0.0–)	23.5 (0.0–)	3.0 (**1.1**↓)	**1.1**↓

branches. For comparing our approach with existing methods to show the comprehensive superiorities of TPF in a fair comparison environment, we choose ResNet-18 and ResNet-50 as our backbone. The structure of FPN can be referred to [50] except for the channel of output fusion feature is set to 512. Meanwhile, in the experimental platform and setup, we adopt 4 Nvidia 1080Ti GPUs for pretraining and finetuning our model and 1 Nvidia 1080Ti GPU to evaluate the model performance in the inference process, the choice of the mainstream GPU ensures a fair hardware condition for performance comparisons. Notably, to adapt to the versions of different dependency libraries, we choose a higher version of Pytorch 1.9 to code the TPF framework and the training and testing processes.

In the data pre-processing stage, the image is resized to the specific size proportionally along its short side. Normally, the short sides of the images in MSRA-TD500 and ICDAR2015 datasets are resized to 736. For Total-Text and CTW1500 datasets, they are re-scaled to 640 in this paper. To further explore the model's ability for dealing with different-scaled input, those image sizes will be adjusted to 512 in comparison experiments. Except for re-scaling input images, data augmentation is extra adopted to improve model generalization in the training stage. Concretely, the augmentation consists of the following four steps mainly:(1) random scaling slightly for increasing the text instance scales; (2) random horizontal flipping for providing inverted training samples; (3) random rotating for generating multi-oriented data with diverse angles; (4) random cropping and padding for ensuring a uniform image size in the same batch.

In the training stage, the CNN layers of backbone, FPN, and four branches have to be initialized first. For the backbone, we load the ResNet that is pre-trained on ImageNet [58] into our model to initialize the corresponding layers. For the other parts of TPF, we initialize them via normal distribution. The training stage includes the two sub-stages of pre-training and fine-tuning. The proposed TPF is pre-trained on SynthText [53] by 1 epoch and is fine-tuned on other official datasets (such as MSRA-TD500, CTW1500, and so on) by 1200 epochs with a batch size of 16. In the backward process of the training stage, Adam optimizer is chosen to propagate the gradient. The initial learning rate is set as 0.001 and is decayed via the 'PolyLr' strategy.

TABLE III
PERFORMANCE ANALYSIS OF THE MODELS WITH DIFFERENT WEIGHTED FPU LOSSES ON THE MSRA-TD500 DATASET. 'α' DENOTES THE WEIGHT OF FPU LOSS. † MEANS THE MODEL IS NOT PRE-TRAINED ON THE SYNTHTEXT DATASET.

α	Presicion	Recall	F-measure
0.1†	88.0	80.3	84.0
0.2†	87.8	81.2	84.4
0.3†	88.7	81.1	84.7
0.4†	88.8	81.9	85.2
0.5†	90.3	80.7	85.2
0.6†	90.0	80.7	85.1
0.7†	89.7	81.7	85.5
0.8†	85.8	83.4	84.6
0.9†	88.4	81.5	84.8
1.0†	89.2	80.5	84.6

C. Ablation Study and Hyperparameter Tuning

In this section, we demonstrate the effectiveness of REU and FPU for improving detection accuracy and analyze the inference speed gains brought by REU. Meanwhile, we analyze the impacts of different weighted FPU loss ($\mathcal{L}_{fpu}$) (can be referred to III-F) and different numbers of center points on model performance. Furthermore, the model's ability for dealing with different-scaled input images is explored. The details of the corresponding experiments are shown next.

Effectiveness and Efficiency of REU. As described in Section III-B, REU is designed for enhancing feature consistency of the same text while strengthening filter's recognition ability

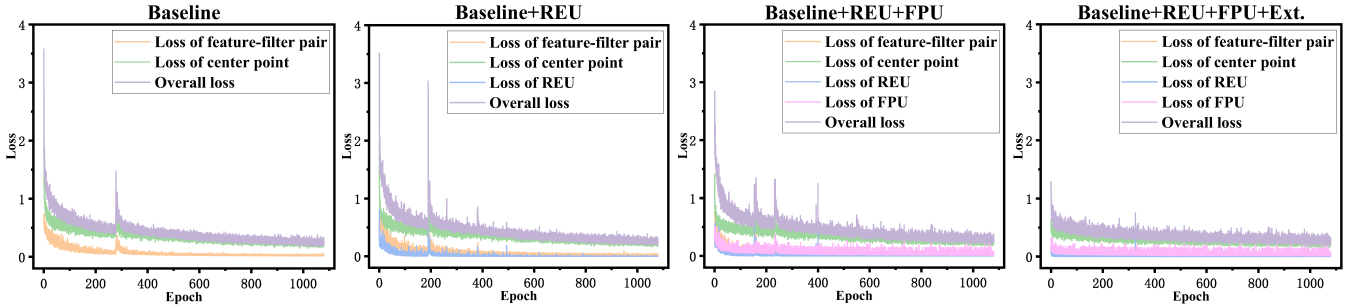


Fig. 8. Visualization of the training processes. We show the training details of the models of 'baseline', 'baseline+REU', 'baseline+REU+FPU', and 'baseline+REU+FPU+Ext.' from left to right, respectively.

to improve the quality of the detection result. To verify the effectiveness of REU, we compare the experimental results of the models with REU and without it in Table I #1–#2. It can be found that the enhanced feature consistency and the strengthened filter brought by REU brings 1.8% improvements in the performance of precision. Meanwhile, we show the training process and improvement by visualizing the predicted binary masks in Fig. 8 and Fig. 9, respectively. Concretely, in the orange circled regions of Fig. 9(b) and Fig. 9(c), it is observed that REU helps to recognize the text instance detail more accurately. Except for the precision, REU helps TPF obtain 1.5% gains in the performance of recall (as shown in Table I #2). We explain the reason in the third row of Fig. 9(b) and Fig. 9(c). The visualization shows that REU can encourage our model to avoid detecting one text instance as multiple parts, which improves the recall rate of detection results effectively. The performance gains in both aspects of precision and recall bring 1.7% improvements in the comprehensive performance of F-measure.The above experimental results demonstrate the effectiveness of the designed REU.

Benefiting from the advantages of ensemble operation of the predicted filters (as illustrated in Fig. 4 and Algorithm 1), REU can speed up the inference process efficiently. It is verified in Table II #1–#2. Though the REU branch brings 16.9 GFLOPs to our model in the inference process, it improves the feature consistency and enlarges the filter recognition field, which helps suppress many low-quality detection results. The enhancement of detection results helps save 50.6% computational costs for the post-processing of TPF. Therefore, the model with REU can bring performance gains in both two aspects of detection accuracy and speed simultaneously.

Effectiveness and Efficiency of FPU. It can be found in Fig. 2 and Fig. 4 that the proposed TPF first locates text instances according to the predicted center points and then extract feature-filter combined with the point coordinates for detecting texts. To improve the reliability of the center point, FPU is introduced to enhance the model's ability to foreground recognition. As shown in Table I, FPU brings 2.1% improvements for the recall rate of detection results, which benefits from the strong distinguishment ability of the foreground from the background brought by FPU. As visualized in the red circled regions of Fig. 9(c) and Fig. 9(d), FPU helps our model recognize hard positive samples. Meanwhile, FPU enhances TPF's ability to suppress false-positive samples (the red boxed regions of Fig. 9(c) and Fig. 9(d)). They are helpful for locating text center points accurately. The above experimental results in Tabla I and Fig. 9 demonstrate the performance gains for predicting center points brought by FPU. Moreover, as we mentioned before, FPU is proposed to help locate text instances accurately and do not participate in the text detection directly. Therefore, it brings no extra computational costs to the inference process (as shown in Table II #2–#3). Benefiting from the advantages of the improvements of text location accuracy, instances that needed to be processed in post-processing are decreased, which makes FPU further save 37% computational costs compared with the model with REU only.

TABLE IV
PERFORMANCE ANALYSIS OF THE MODELS WITH DIFFERENT NUMBERS OF CENTER POINTS ON THE TOTAL-TEXT DATASET. 'N' DENOTES THE CENTER POINT NUMBER. † MEANS THE MODEL IS NOT PRE-TRAINED ON THE SYNTHTEXT DATASET.

N	Precision	Recall	F-measure	Post cost (ms)	FPS
5†	84.9	83.4	84.1	5.1	40.2
10†	87.2	82.8	84.9	7.9	36.1
15†	87.2	84.1	85.6	11.9	31.5

Impacts of Different Weighted FPU Losses. The experimental results in Table I and Fig. 9 show the strong recognition of the foreground brought by FPU can help our model suppress false-positive samples and detect hard-positive samples, which brings gains for detection performance effectively. To further explore the effectiveness of FPU, we evaluate the performance of the models with different weighted FPU Losses. As shown in Table III, we tune the FPU loss weight α from 0.2 to 1.0 for testing model performance, respectively. It can be observed that there is a significant gain when α is set as 0.7. Compared with the model without FPU (Table I #2), the experimental results demonstrate the effectiveness of FPU in improving the model's ability to distinguish the foreground. Furthermore, the performance continues to slow-degrading when α is tuned smaller or bigger than 0.7. Based on this conclusion, α is set as 0.7 in the next experiments unless otherwise noted.

Impacts of Different Numbers of Center Points. For the center point prediction header, we sample a specific number of points from each text region as the training label (as shown in Fig. 6). To verify the impacts of different numbers of center

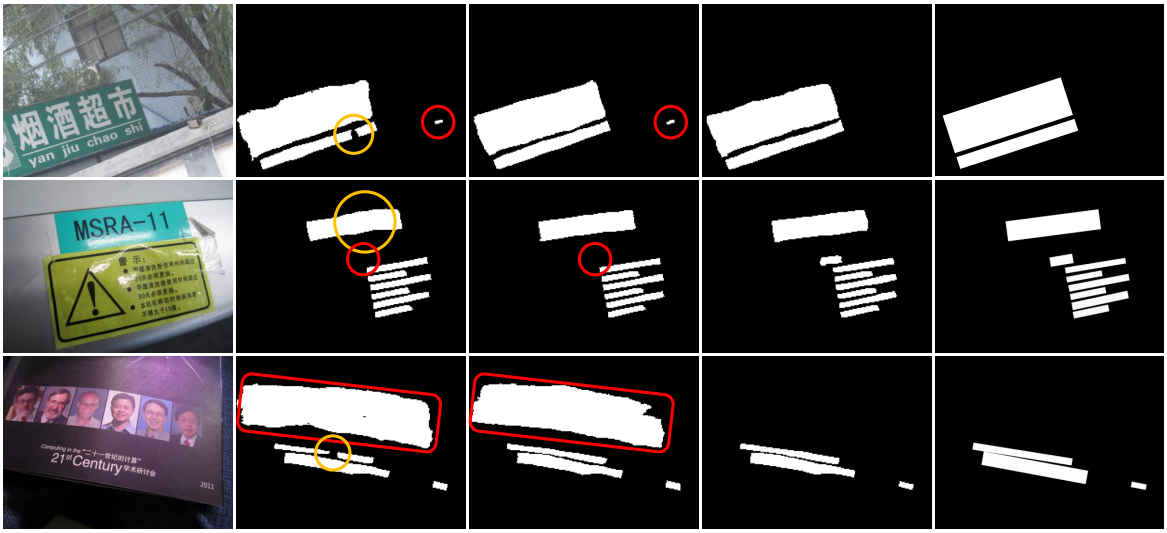


Fig. 9. Visualization of the detection results of the models with different settings. The orange circled regions are the high-quality results improved by REU. The red circled regions show the hard positive samples are detected and the false-positive samples are suppressed with the help of strong distinguishment ability to the foreground brought by FPU.

TABLE V
PERFORMANCE ANALYSIS OF THE MODELS WITH DIFFERENT-SCALED SAMPLES ON THE MSRA-TD500 DATASET. 'SCALE' DENOTES THE SHORT SIDE SIZE OF THE TESTING IMAGE. ‡ MEANS THE MODEL IS PRE-TRAINED ON THE SYNTHTEXT DATASET.

Scale	Precision	Recall	F-measure	FPS
512‡	91.7	82.9	87.1	63.7
640‡	92.7	82.3	87.2	45.2
736‡	91.2	84.6	87.8	34.5

points on model performance, we tune the number of the sampled center points from 5 to 10 and 15, respectively. As visualized in Table IV, a larger number of the sampled center points can bring gains for the detection quality. Meanwhile, we show some representative results that are predicted by the models with different center point numbers in Fig 10. It is observed that increasing the number of center point helps distinct the feature difference between the foreground and background (as shown the red circled regions in Fig. 10(b)–(c)). The above experimental results demonstrate that more center points help enhance the recognition of the foreground. However, it leads to our model generating more filters in the inference process, which brings extra computational costs. To achieve a better comprehensive performance between detection accuracy and speed, we choose to sample 5 center points from each text in the experiments of Section IV-D.

Ability for Dealing with Different-Scaled Samples. REU is designed to enhance the filter's recognition capability, enabling the model to effectively handle samples of varying scales and improve its generalization. In this paragraph, we evaluate the model on the MSRA-TD500 dataset using multiple image sizes to demonstrate its scale invariance within a specific range. Specifically, as shown in Table V, the model's performance remains consistent when the image size is resized from 512 to 640 pixels, achieving the highest accuracy at an image size of 736 pixels. This analysis of both the testing and training processes validates the effectiveness of REU and confirms that TPF performs well across different scales.

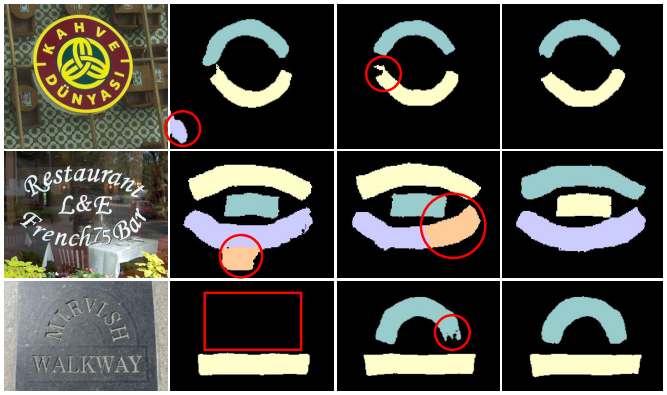


Fig. 10. Visualization of the detection results of the models with different numbers of the sampled center points.

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

The efficiency of TPF for arbitrary-shaped text detection and the effectiveness of REU and FPU are verified in Section IV-C. Meanwhile, we explore the impacts of different weighted FPU losses and different numbers of center points on model performance. Furthermore, the model's ability for dealing with different-scaled samples is verified. The above ablation studies help to understand TPF details and to construct the most efficient framework. In this section, we further to show the superior comprehensive performance of TPF by comparing it with existing state-of-the-art (SOTA) methods under the guidance of the experimental results before. Our model is pre-trained on SynthText (unless otherwise noted) in the next experiments to compare with existing methods.

Fig. 11. Qualitative detection results of TPF on four public datasets (including the MSRA-TD500, ICDAR2015, Total-Text, and CTW1500 datasets).

TABLE VI
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE MSRA-TD500 BENCHMARK. WE HIGHLIGHT THE BEST RESULT OF THE EXISTING SOTA METHOD AND THE PROPOSED TPF THROUGH "**GRAY BACKGROUND**". "OURS-640" AND "OURS-736" DENOTE THE SHORT SIZES OF IMAGES ARE RESIZED AS 640 AND 736 IN THE INFERENCE PROCESS, RESPECTIVELY. RESNET18 IS EMPLOYED FOR MULTI-SCALE FEATURE EXTRACTION UNLESS STATED OTHERWISE.

Methods	Venue	Precision	Recall	F-measure	FPS
PAN [46]	ICCV'19	84.4	83.8	84.1	30.2
DB [48]	AAAI'20	91.5	79.2	84.9	32.0
MTS-v3 [39]	ECCV'20	90.7	77.5	83.5	–
GV [27]	TPAMI'20	88.8	84.3	86.5	15.0
MCN [59]	IJCV'20	89.1	80.7	85.2	–
ABPN [60]	ICCV'21	85.4	80.7	83.0	12.7
CM-Net [19]	TIP'22	89.9	80.6	85.0	41.7
RFN [6]	TCSVT'22	88.4	80.0	84.0	–
PAN++ [47]	TPAMI'22	85.3	84.0	84.7	32.5
SPM [26]	TPAMI'23	88.6	82.7	85.5	8.3
DB++ [20]	TPAMI'23	91.5	83.3	87.2	29.0
MorphText [1]	TMM'23	88.5	82.7	85.5	–
ABPN [2]	TMM'24	89.2	85.4	87.3	15.2
Ours-640	–	92.7	82.3	87.2	45.2
Ours-736	–	91.2	84.6	87.8	34.5

Evaluation on MSRA-TD500. To demonstrate the superiority of TPF for detecting long multi-lingual texts, we compare the detection results with other SOTA works published recently. Particularly, We show the model performance for dealing with different-scaled images to illustrate the superior comprehensive performance of TPF.

Specifically, as we can see from Table VI, our model achieves 87.8% in F-measure when the input image is resized to 736, which outperforms the SOTA method (BoundTrans [2]) 0.5% in F-measure. Particularly, benefiting from the advantages of REU, TPF enjoys a strong ability for dealing with multi-scaled samples, which helps our method achieves comparable detection accuracy with DB++ [20] when the input is resized to 640 while running 16.2 FPS faster than it. For DB [48], PAN [46], and PAN++ [47], TPF outperforms them a lot in both detection accuracy and speed simultaneously, where the important reason is that different from previous whole region-based methods, TPF simulates the operation mode of band-pass filter to detect texts. Therefore, it can separate adhesive texts naturally without extra decoding or post-processing processes. To better show the model's ability to detect long multi-lingual texts, we show some qualitative detection results in Fig. 11(a). The above experimental results verify the effectiveness of TPF for detecting long text lines on the samples of the MSRA-TD500 benchmark.

Evaluation on ICDAR2015. We have demonstrated the TPF's strong ability to detect line-level multi-oriented texts before. In this section, we testing our method on ICDAR2015 benchmark to further verify the superiority for detecting word-level multi-oriented texts from the complex background.

In Table VII, existing SOTA methods, such as PFText [3], PAN++ [47], and SLN [9], can achieve 85.9%, 84.5%, and 85.0% in F-measure respectively. Specifically, PAN++ and SLN can run 19.2 and 11.2 FPS in the inference process. They enjoy a comparable comprehensive performance. For the proposed TPF, FPU encourages it to recognize the center points more accurately, which helps detect text instances from the background with plenty of noise. Based on the above advantages, TPF performs better than previous works for the

TABLE VII
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE ICDAR2015 BENCHMARK. WE HIGHLIGHT THE BEST RESULT OF THE EXISTING SOTA METHOD AND THE PROPOSED TPF THROUGH "**GRAY BACKGROUND**". "OURS-736-R50" MEANS THE SHORT SIZES OF IMAGES ARE RESIZED AS 736 AND THE RESNET50 IS ADOPTED AS THE BACKBONE. RESNET18 IS EMPLOYED FOR MULTI-SCALE FEATURE EXTRACTION UNLESS STATED OTHERWISE.

Methods	Venue	Precision	Recall	F-measure	FPS
MSR [41]	IJCAI'19	86.6	78.4	82.3	4.3
PAN [46]	ICCV'19	84.0	81.9	82.9	26.1
DR-PSS [10]	TCSVT'20	87.0	81.0	84.0	6.7
Boundary [43]	AAAI'20	82.2	88.1	85.0	–
FCE-Net[16]	CVPR'21	85.1	84.2	84.6	–
SLN [9]	TCSVT'21	88.0	83.0	85.0	11.2
TextDCT [4]	TMM'22	86.9	83.7	85.3	7.6
PATD [5]	TCSVT'22	85.9	82.7	84.3	–
RFN [6]	TCSVT'22	88.4	80.0	84.0	–
KPN [61]	TNNLS'22	84.1	83.2	83.6	12.2
PAN++ [47]	TPAMI'22	88.7	80.7	84.5	19.2
SPM [26]	TPAMI'23	88.2	83.3	85.7	–
PFText [3]	TMM'23	89.6	82.4	85.9	–
Ours-736	–	87.8	84.0	85.9	28.7
Ours-736-R50	–	91.1	84.4	87.6	11.3

TABLE VIII
COMPARISONS ON THE TOTAL-TEXT AND CTW1500. WE HIGHLIGHT THE BEST RESULT OF DIFFERENT METHODS THROUGH "**GRAY BACKGROUND**". "OURS-640-R50" MEANS THE SHORT SIZES OF IMAGES ARE RESIZED AS 640 AND THE RESNET50 IS ADOPTED AS THE BACKBONE. RESNET18 IS EMPLOYED BY DEFAULT FOR MULTI-SCALE FEATURE EXTRACTION. METHODS IN THE BOTTOM SECTION DO NOT ADOPT A TRANSFORMER STRUCTURE.

Methods	Total-Text				CTW1500			
	P	R	F	FPS	P	R	F	FPS
STKM [14]	86.3	78.4	82.2	–	85.1	78.2	81.5	–
ABPN [2]	89.9	85.3	87.5	12.0	88.1	81.1	84.5	14.7
TextPMs [22]	89.8	87.8	87.2	6.8	87.6	80.8	84.1	9.0
DPText [23]	91.8	86.4	89.0	–	91.7	86.2	88.8	–
DeepSolo++ [24]	93.9	82.1	87.6	–	92.5	86.3	89.3	–
OPMP [13]	87.6	82.7	85.1	1.4	85.1	80.8	82.9	1.4
FCE-Net [16]	87.4	79.8	83.4	–	85.7	80.7	83.1	–
RDSS [62]	87.1	80.3	83.5	–	87.3	81.8	84.5	–
NASK [8]	91.1	52.5	66.6	–	83.4	80.1	81.7	12.1
ABS-Net [7]	85.6	83.2	84.4	8.4	92.7	74.4	82.4	–
TextDCT [4]	85.8	80.5	83.0	15.2	84.7	81.5	83.1	17.3
KPN [61]	88.0	82.3	85.1	22.7	84.0	82.9	83.4	24.3
PAN++ [47]	89.9	81.0	85.3	38.3	87.1	81.1	84.0	36.0
DB++ [20]	87.4	79.6	83.3	48.0	84.3	81.0	82.6	49.0
MorphText [1]	88.4	85.5	86.9	–	89.0	83.2	86.0	–
Ours-512	86.4	82.5	84.4	58.9	85.0	83.1	84.0	60.6
Ours-640	87.1	84.3	85.7	34.0	86.8	82.5	84.6	41.2
Ours-640-R50	88.5	85.3	86.9	17.4	87.9	84.5	86.2	19.1

detection task on the ICDAR2015 dataset. Concretely, it outperforms the SOTA method SLN 0.9% in detection accuracy and 17.5 FPS in detection speed. Though PAN [46] can run 26.1 FPS, our method outperforms it 3.0% in F-measure. Furthermore, some results are shown in Fig. 11(b), where text instances can be recognized clearly even though there are many interference regions in the background. Combining the detection results illustrated in Table VII and visualized in Fig. 11(b), we verify the remarkable performance of TPF to detect word-level multi-oriented texts successfully.

Evaluation on Total-Text and CTW1500. Different from the datasets before, Total-Text consists of plenty of word-level curved text instances, which are used for testifying the model's ability to detect texts with irregular shapes separately. Meanwhile, except for the Total-Text dataset, we further testing TPF on the CTW1500 benchmark and compare it with previous methods to show the superior performance for recognizing ribbon-like curved texts and ensuring the integrity of them.

Considering the different features between Transformer [63] and CNN, we show some representative methods of the two structures in Table VIII top and bottom sections, respectively. As the best CNN-based method for arbitrary-shaped text detection published recently, MorphText [1] can achieve 86.9% and 86.0% in F-measure on Total-Text and CTW1500 datasets. Though the above method surpasses the other SOTA methods a lot in F-measure, the slow running speed limits further performance improvement. Furthermore, in the aspect of detection speed, DB++ [20] and PAN++ run faster almost 4 times than existing related works [4]. Though DB++ performs well in both detection accuracy and speed on the Total-Text and CTW1500 datasets, our method still can achieve a better comprehensive performance than it. As we introduced before, TPF segments the whole regions of texts directly, which avoids the limitation of destroyed semantic integrity and confused feature differences between the foreground and background that exists in shrink-mask-based methods. Meanwhile, it can separate adhesive texts without complex decoding or post-processing processes. The above advantages bring gains to the comprehensive performance of TPF. Specifically, our model outperforms DB++ 2.3% and 2.0% in F-measure on Total-Text and CTW1500 benchmarks respectively and runs 10 FPS faster than it at least. Moreover, for the accuracy prior method (ContourNet), TPF still can surpass it 0.2% in F-measure while running almost 10 times faster than it. Compared to current SOTA transformer-based methods (DPText [23] and DeepSolo++ [24]), our method performs worse in terms of F-measure but achieves faster inference speed. The differences in accuracy and efficiency among these methods primarily stem from the distinctions between transformer and CNN architectures. In addition to performance differences, their hardware resource dependencies vary significantly. Transformer-based methods require GPUs with large memory (e.g., 4090 GPU or A100 GPU), which may be inaccessible to many researchers. In contrast, CNN-based detectors can typically be trained and evaluated effectively on more accessible hardware like the 1080Ti GPU. The CNN-based methods allow researchers without enough hardware resources to promote the development of efficient scene text methods in the aspect of framework design.

E. Analysis of Model Limitations

We have explored the importance of different factors affecting model performance and have demonstrated the superior performance of the proposed TPF on multiple public datasets before. Further, to help understand TPF comprehensively, we

(a) text overlay (b) adjacent similar text

Fig. 12. Visualization of challenging samples. The green contours are the correct predicted results and the red ones are the failure cases.

discuss the weaknesses of TPF in this section, illustrating scenarios that our model finds hard to handle. As shown in Fig. 12, there are two types of representative challenging samples for our method. Firstly, text overlay (referred to Fig. 12(a)) is a classic challenge to scene text detection. Current methods distinguish texts from the background via the recognition of visual features. These methods lack the semantic analysis from high-level for each latent text region, which is an intrinsic deficiency. Similarly, our TPF is hard to handle this case either. Besides, as we described in Section III, REU merges multiple filters that belong to the same text to a strengthen filter to improve the accuracy of predicted results. However, during the strengthening process, some filters with similar visual features that belong to different texts may be merged together by REU. This causes the strengthened filter to allow multiple texts to pass instead of just one, leading to an overdetection problem (referred to Fig. 12(b)). Therefore, there are still challenges to be considered when applying existing text detection algorithms in real-world scenarios, and vision-language models will be introduced in this field to help alleviate residual problems.

V. Conclusion

In this paper, we are inspired by electronics and signal processing to design a text detector, namely Text-Pass Filter (TPF), to simulate the band-pass filter for detecting arbitrary-shaped texts efficiently even though texts are close to each other. It ensures the text's semantic integrity and avoids confusion about the feature difference between the foreground and background to enhance the model's text recognition ability. Meanwhile, TPF can separate adhesive texts according to their unique features without complex decoding and post-processing processes to make it possible for real-time text detection. Furthermore, a Reinforcement Ensemble Unit (REU) is designed to enhance the text feature consistency and to enlarge the filter's recognition field for improving the precision performance. In the end, a Foreground Prior Unit (FPU) is introduced to guide our model to distinguish texts from the background for bringing gains in the recall rate of detection results. Experiments on multiple public datasets demonstrate the superior efficiency of TPF compared with existing methods and the effectiveness of REU and FPU.

References

[1] 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*, 2023.

[2] 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, 2023.

[3] 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*, 2023.

[4] 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*, p. Advance online publication, 2022.

[5] P. Keserwani, R. Saini, M. Liwicki, and P. P. Roy, "Robust scene text detection for partially annotated training data," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8635–8645, 2022.

[6] 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.

[7] L. Nandanwar, P. Shivakumara, R. Ramachandra, T. Lu, U. Pal, A. Antonacopoulos, and Y. Lu, "A new deep wavefront based model for text localization in 3d video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3375–3389, 2022.

[8] 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.

[9] 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.

[10] P. Cheng, Y. Cai, and W. Wang, "A direct regression scene text detector with position-sensitive segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4171–4181, 2020.

[11] 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.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

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

[14] Q. Wan, H. Ji, and L. Shen, "Self-attention based text knowledge mining for text detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 5983–5992.

[15] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Text growing on leaf," *IEEE Transactions on Multimedia*, p. Advance online publication, 2023.

[16] 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.

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

[18] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Zoom text detector," *IEEE Transactions on Neural Networks and Learning Systems*, p. Advance online publication, 2023.

[19] 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*, vol. 31, pp. 2864–2877, 2022.

[20] 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.

[21] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Reinforcement shrink-mask for text detection," *IEEE Transactions on Multimedia*, p. Advance online publication, 2022.

[22] 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, 2022.

[23] 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.

[24] M. Ye, J. Zhang, S. Zhao, J. Liu, T. Liu, B. Du, and D. Tao, "Deepsolo++: Let transformer decoder with explicit points solo for multilingual text spotting," *arXiv preprint arXiv:2305.19957*, 2023.

[25] 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.

[26] 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.

[27] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 4, pp. 1452–1459, 2020.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Conference on Neural Information Processing Systems*, 2015, pp. 91–99.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[31] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.

[32] 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.

[33] W. He, X. Zhang, F. Yin, and C. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 745–753.

[34] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4161–4167.

[35] 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.

[36] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5909–5918.

[37] 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*, 2018, pp. 6773–6780.

[38] 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.

[39] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 706–722.

[40] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Bip-net: Bidirectional perspective strategy based arbitrary-shaped text detection network," in *2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 2255–2259.

[41] C. Xue, S. Lu, and W. Zhang, "Msr: Multi-scale shape regression for scene text detection," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 989–995.

[42] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 10 552–10 561.

[43] 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.

[44] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 11 753–11 762.

[45] 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.

[46] 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.

[47] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5349–5367, 2022.

[48] 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*, 2020, pp. 11 474–11 481.

[49] 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.

[50] T.-Y. 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.

[51] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[52] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Fourth International Conference on 3D Vision*, pp. 565–571.

[53] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2315–2324.

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

[55] 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.

[56] D. Karatzas, L. Gomez, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. Chandrasekhar, and S. Lu, "Icdar 2015 competition on robust reading," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2015, pp. 1156–1160.

[57] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, 2017, pp. 935–942.

[58] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[59] Z. Liu, G. Lin, and W. L. Goh, "Bottom-up scene text detection with markov clustering networks," *International Journal of Computer Vision*, vol. 128, no. 6, pp. 1786–1809, 2020.

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

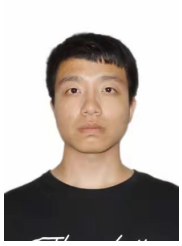
[61] S.-X. Zhang, X. Zhu, J.-B. Hou, C. Yang, and X.-C. Yin, "Kernel proposal network for arbitrary shape text detection," *IEEE Transactions on Neural Networks and Learning Systems*, p. Advance online publication, 2022.

[62] W. Feng, F. Yin, X. Zhang, W. He, and C. Liu, "Residual dual scale scene text spotting by fusing bottom-up and top-down processing," *International Journal of Computer Vision*, vol. 129, no. 3, pp. 619–637, 2021.

[63] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

Chuang Yang received 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, in 2024. He is currently a Postdoc Fellow with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include signmark-guided VLN and AIGC for remote sensing.

Haozhao Ma received the B.E. degree in computer science and technology from Hefei University of Technology, Hefei, China, in 2022. He is currently pursuing the M.E. degree in the School of Software and School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and machine learning.



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 School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, pattern recognition and text detection.

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

Qi Wang (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, machine learning, and remote sensing. For more information, visit the link (https://crabwq.github.io/)