Pull Pole Points to Text Contour by Magnetism: A Real-Time Scene Text Detector

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

***Abstract*—Scene text reading plays a crucial role in scene understanding. As its precondition task, scene text detection has garnered increasing interest from researchers. Segmentation-based text detection methods have gained prominence due to their adaptable pixel-level predictions. Many existing methods predict the shrink mask and utilize the Vatti clipping algorithm to reconstruct text contours. However, the shrink mask only focuses on the global geometry feature and shrinks the same distance everywhere, which neglects local contour information and disrupts the instance shape feature. In addition, the post-processing based on the Vatti clipping algorithm heavily relies on the predictions and is relatively complex, causing suboptimal performance in both detection accuracy and efficiency. To address the above problems, we propose an efficient and effective method named Magnetic Text Detector (MTD), inspired by magnetism. It is constructed by a text representation method flexible mask (FM) and a magnetic pull module (MPM). Unlike the shrink mask and concentric mask, the former concerns the local contours and shrinks unfixed distances on different positions, which avoids the truncation issue while preserving distinctiveness from the text regions. The latter generates magnetic fields and pulls pole points of FM to the text contour by magnetism. This allows accurate reconstruction of text contours, even when predictions deviate from the actual text severely, while saving 50% of the post-processing time approximately. Several ablation studies verify the effectiveness of the proposed FM and MPM. Extensive experiments show that our MTD achieves state-of-the-art (SOTA) methods on multiple datasets from different scenes. The code is available at https://github.com/fengmulin/MTD.**

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

I. Introduction

In recent years, scene text reading has garnered significant attention due to its wide-ranging applications, including image understanding, autonomous driving, and visual search. Text detection [1]–[4], a crucial step in scene text reading, aims to accurately identify text regions within the background and distinguish between different text instances. It is still a tough challenge as the various sizes, aspect ratios, shapes, fonts, and lighting conditions in scene texts. Among them, the irregular shape is the most tricky problem to be addressed.

The rapid advancement of deep learning has significantly boosted scene text detection. Among mainstream methods, segmentation-based approaches have shown superior capability in handling arbitrary-shaped text, thanks to their pixel-level predictions. However, these methods often struggle with overlapping nearby text instances. To address this, prior works introduced shrink mask-based strategies, such as PSE-Net [6], DB-Net [7], and PAN [8], which shrink the text region and then recover its contour through expansion or clustering. Despite their effectiveness, shrink-mask-based methods apply a uniform shrinking distance computed from global features like area or perimeter, neglecting local geometry. This leads to contour distortion, as shown in Fig. 1(a), and increases learning difficulty—sometimes even splitting a single instance into multiple masks. To overcome this, works like Concentric Mask (CM) [3] and Han *et al.* [9] attempt local adaptation but still use uniform offset within each instance. As shown in Fig.

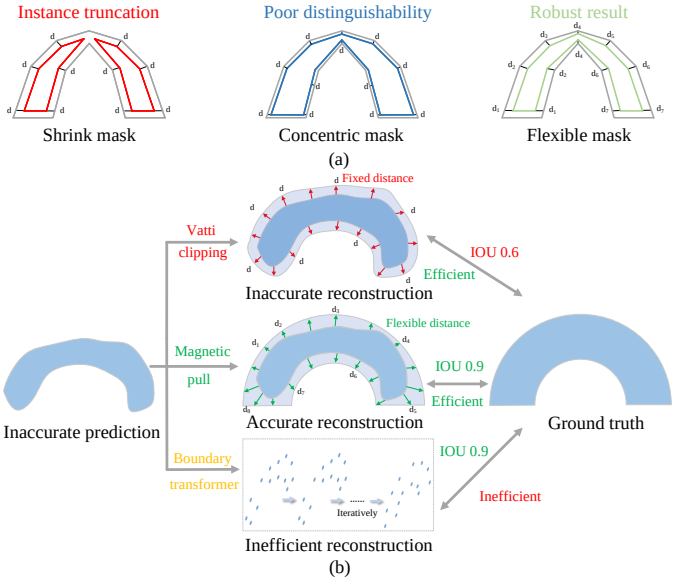


Fig. 1. (a) The comparisons of existing text representation methods with the proposed flexible mask (FM). (b)The comparisons of existing post-processing methods with the proposed magnetic pull module (MPM). The shrink mask shrinks based on global geometric features, causing some instances to shrink into two masks. The concentric mask shrinks based on local geometric features, resulting in minimal differences between the mask and the text region, making it difficult to distinguish. The above methods shrink the same distance in all directions. Different from them, our method adapts to the situation by shrinking different distances at different positions, which ensures instances are not truncated and maintains the distinctiveness of the text region. Vatti clipping expands a fixed distance in fixed directions, resulting in poor correction capability for prediction. Boundary transformer (TextBPN [5]) updates contour points iteratively, which fixes the number of points and is inefficient. The proposed MPM adaptively adjusts the magnitude and direction of the offset based on the prediction in one step, reducing reliance on predictions and maintaining high efficiency.

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 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, crabwq@gmail.com). (Xu Han and Chuang Yang contributed equally to this work.) (Corresponding author: Qi Wang.)

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

TABLE I
COMPARISON OF THREE SHRINK-BASED METHODS. "IOU" DENOTES THE AVERAGE IOU BETWEEN MASKS AND TEXTS, WHILE "TRUNCATION" INDICATES THE NUMBER OF TRUNCATED INSTANCES IN THE DATASET.

Methods	IoU		Truncation	
	CTW1500	TotalText	CTW1500	TotalText
Shrink mask	0.257	0.289	44	29
Concentric mask	0.545	0.627	5	4
Flexible mask	0.414	0.374	0	0

1(a), when encountering instances with locally narrow widths, according to the barrel effect, the concentric mask and the text region will be very close, which deviates from the original intention of the shrinkage-based method. We propose a new text representation called the flexible mask, which shrinks the instance by non-uniform distances derived from local contour features. Compared to prior methods, the flexible mask offers two key benefits: (1) It preserves both local shape integrity and separation between masks, combining the strengths of shrink and concentric masks. (2) It reduces the geometric distortion and model learning difficulty, leading to more robust training and inference. Table I reports the average IoU, which quantifies the difference between the different mask and the text region, and the number of truncated instances, which reflects the severity of instance truncation. The Concentric Mask achieves the highest average IoU, indicating limited separation from the text region. The Shrink Mask suffers from the truncation problem. The Flexible Mask attains a moderate average IoU, maintaining sufficient distinction from the text, while completely eliminating instance truncation—demonstrating its effectiveness in preserving complete text instances

In addition to flexible masks, we also address the limitations of post-processing in existing works. Traditional methods (e.g., PSENet, DB-Net, ADNet) perform expansion using handcrafted rules or predicted expansion distances, but typically apply uniform expansion in all directions. As shown in Fig. 1(b), this leads to over- or under-expansion, degrading accuracy. To this end, we introduce the Magnetic Pull Module (MPM), a physically inspired mechanism that simulates opposing magnetic forces along the long sides of the text instance. This module enables direction- and distance-adaptive offset at each pixel, offering: (1) Reduced reliance on shrink mask accuracy, as adaptive magnetic fields improve localization robustness; (2) Lightweight and fast post-processing, reducing latency by over 50% while outperforming previous segmentation-based approaches. While boundary-based methods such as TextBPN [5] and TextBPN++ [10] utilize transformer modules to refine contours, they differ significantly from our segmentation-based framework. Their reliance on fixed contour points and multiple optimization iterations limits efficiency. Moreover, their distance fields serve only as coarse proposals, whereas our magnetic fields directly guide instance reconstruction. The prominent contributions of this paper are as follows:

- A robust and adaptable text representation method flexible mask (FM) is proposed. Unlike previous methods that shrink the same distance everywhere, it focuses on the local contour to shrink different distances. It maximally preserves the geometric features of the text contour, ensuring instances are not shrunk into two masks, while also maintaining the distinction from the text region.
- A magnetic pull module (MPM) is proposed, which divides the text region into bipolar to generate magnetic fields like magnets. It pulls the pole points of FM to text contour to reconstruct instances accurately, even if the predictions deviate from the ground truth. It reduces the reliance on the prediction of FM and saves about half of the post-processing time, enhancing flexibility and improving the efficiency of the entire framework compared to previous methods.
- An effective and efficient scene text detector named Magnetic Text Detector (MTD) based on the above modules is proposed, which achieves state-of-the-art (SOTA) performance and high speed across multiple datasets.

II. RELATED WORK

Scene text detection has received increased attention in computer vision. In recent years, a plethora of outstanding work has emerged, which can be broadly categorized into non-real-time and real-time methods.

A. Non-Real-Time Methods

Non-real-time text detection methods focus on high detection accuracy with paying no attention to efficiency. Some methods are inspired by general object detection frameworks, such as Faster R-CNN [11]. For example, TextBoxes [12] modified the shrink scales and aspect ratios of anchors based on SSD [13]. Textboxes++ [14] added an angle parameter to cope with multi-directional texts. RRPN [15] adopted a modified framework based on Faster-RCNN and utilized a region-proposal-based method to predict the orientation. EAST [16] predicted score maps and multi-channel geometry maps, which include rotated box and quadrangle. With the rapid advancement of scene text detection, detecting text in arbitrary shapes has emerged as a research hotspot. However, the above methods are not well-suited for coping with it. To address this problem, PCR [17] proposed a progressive contour regression method to evolve the prediction from horizontal texts to irregular shapes. CTNet [18] also adopted a contour refinement strategy, which proposed an adaptive training method to learn various offsets and a re-score method to suppress false positives. FCE-Net [19] and ABC-Net [20] represented text instances based on Fourier signature vectors and the Bezier curve, respectively. CTPN [21] detected the fixed-sized width text components based on Faster-RCNN and reconstructed text instances by connecting them. SegLink [22] represented text instances as segments and links. Then it rebuilt instances by connecting segments according to the predictions of links. TextSanke [23] represented text instances via a series of disks that predict their center line, angle, and radius. To address the complex post-processing typically required by connected component (CC)-based methods, ERR-Net [24] innovatively reformulates text detection as an object tracking task without post-processing. PSENet [6] predicted different scale kernels and recovered text instances via the progressive scale expansion algorithm. CBNet [25] proposed

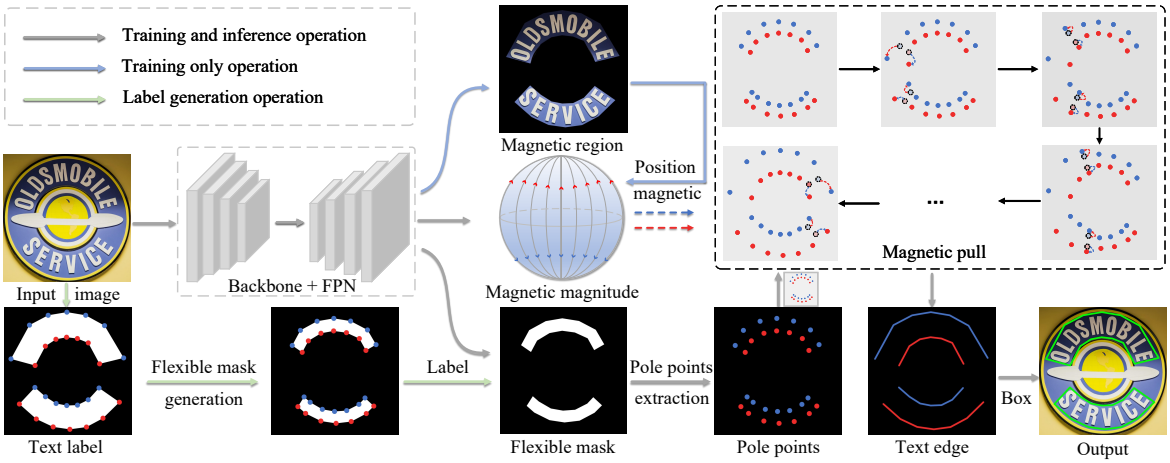


Fig. 2. The overall structure of the proposed MTD. It includes the backbone, FPN, magnetic region prediction head, flexible mask prediction head, magnetic magnitude prediction heads, and simplified post-processing.

a context-aware module to help the model capture both global and local contexts. In addition, an adaptive expansion value was introduced to guide the reconstruction of text contours, and text region predictions were used to further refine the reconstruction results. LeafText [26] proposed a novel contour modeling approach inspired by leaf vein biomimicry, which accurately fits text of various shapes. LRANet [27] proposed a novel text contour representation method based on the singular value decomposition. EdgeText [28] introduced a quadratic polynomial fitting scheme to model text contours in an elegant and effective manner. To advance research in industrial text detection, Guan *et al.* [29] established an industrial text dataset and a corresponding synthetic dataset. In addition, they proposed a feature integration strategy to extract different scale features. Although their methods cope with irregular-shaped text well, their efficiency limits application in the real world.

B. Real-Time Methods

To make scene text detection obtain faster speed and apply it in the real world, numerous real-time detection methods have been proposed. These methods commonly utilize a lightweight backbone and a shrink-based approach to represent text instances. PAN [8] predicted shrink masks and rebuilt text instances by clustering pixels. DBNet [7] predicted shrink masks and expanded stable distance in every direction, which was calculated by the area and perimeter of instances. DBNet++ [30] introduced an adaptive scale fusion module based on DBNet to improve detection accuracy. To address the instability issue of directly expanding based on the area and perimeter of the shrink mask in DBNet, ADNet [31], and RSMTD [32] propose different solutions. The former suggests that text instances of different shapes should use different expansion factors, predicting these factors accordingly. The latter takes a more direct approach by predicting the distances that different instances need to be expanded. CT [33] proposed an efficient text instance representation method and a corresponding post-post-processing.FEPE [34] proposed a focus entirety module and a perceive environment module, used exclusively during training, to extract region-level and instance-level features. CM-Net [3] also proposed a novel text representation concentric mask and a multi-perspective feature module to assist in feature learning. The concentric mask considers the local geometry feature and calculates the shrinkage distance based on the barrel effect, preventing instances from being truncated in certain experiments. However, this may lead to some concentric masks overlapping excessively with the text region, thereby losing the significance of shrinkage. Different from the above methods, Wang *et al.* [1] directly predicted the text region and proposed a text separatrix to separate different instances. The above methods improve the speed to some extent. However, their post-progressing still takes a lot of time in the whole process.

III. Methodology

The overall framework of the MTD is introduced first. Then, the proposed FM and MPM are described in detail. Finally, the label generation and loss functions are represented.

A. Overall Architecture

The overall framework of MTD is shown in Fig. 2. The image is fed into the backbone to extract different scale feature maps. Then, a fused feature map $\mathbf{F}$ is obtained through a feature pyramid network (FPN) [35]. Subsequently, $\mathbf{F}$ is used to predict the magnetic magnitude, flexible mask map, and magnetic region. The magnetic region is defined as the area where the magnetic magnitude is non-zero. In other words, the magnetic region prediction head transforms the regression task of magnetic magnitude into a binary classification task, thereby providing additional supervisory information that facilitates the learning of magnetic magnitude. The pole point is extracted from the flexible mask map through morphological operations. The pole points are pulled to generate pulling pole points according to the magnetic fields that MPM predicts. Finally, the pulling pole points form the positive and negative poles to reconstruct contours. It is worth noting that, in order to reduce computational overhead, the flexible mask and the magnetic magnitude share the same prediction head, with the

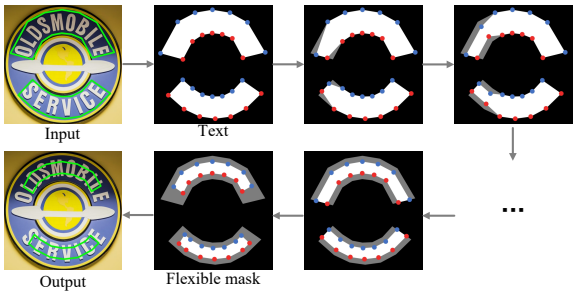


Fig. 3. The label generation process of the flexible mask.

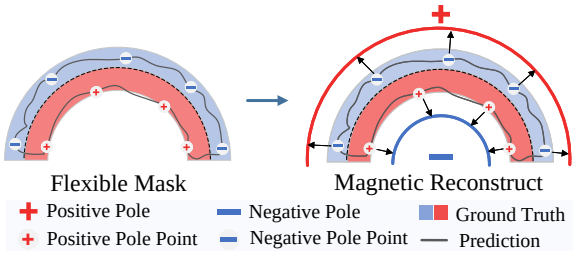


Fig. 4. Illustration of the reconstruction process of instances.

only difference being the addition of a sigmoid layer for the former. This can be formulated as follows:

$$\mathrm{H}_1 = \mathrm{ReLU}_{\mathrm{BN}}(\mathrm{Conv}_{3\times3}(\mathrm{F})), \tag{1}$$

$$\mathrm{Res} = \mathrm{ConvT}_{2\times2}(\mathrm{ReLU}_{\mathrm{BN}}(\mathrm{ConvT}_{2\times2}(\mathrm{H}_1))), \tag{2}$$

$$\mathrm{F}_{\mathrm{m}} = \mathrm{Res}[0,:,:], \mathrm{M}_{\mathrm{m}} = \mathrm{Res}[1:3,:,:], \tag{3}$$

where F_{m} and M_{m} represent the prediction of the flexible mask and magnetic magnitude. The magnetic region prediction head shares the same structure as the flexible mask prediction head, except that its final output layer generates a single-channel prediction.

B. Text Representation

The proposed method represents text by the flexible mask. With its help, irregular-shaped text can be rebuilt robustly, which is superior to the shrink mask that the existing state-of-the-art (SOTA) real-time methods used.

1) Shrink Mask: PAN [36], and DBNet++ [30] utilize the Vatti clipping algorithm [37] to generate shrink masks. It is generated by shrinking the instance for a distance d_{sm}, which is calculated as follows:

$$d_{sm} = \frac{A}{L}\left(1-\gamma^2\right), \tag{4}$$

where A and L represent the area and perimeter of the instance. γ is the shrinkage coefficient. The shrink mask mainly has two shortcomings: 1) The shrink distance is computed according to the global characteristics, which ignore local features and fail to rebuild some instances. 2) It shrinks the same distance everywhere, which is not flexible and not suitable for different local contours.

2) Concentric Mask: CMNet [3] proposes a concentric mask to replace the shrink mask, which is derived by shrinking the text inward by d_{cm}. It can be formulated as follows:

$$d_{cm} = \frac{1}{2}\min\left(\|p_{cp}, p_n\|_2^2\right), \quad n = 1, 2, \ldots, N, \tag{5}$$

where p_{cp} and p_n represent the center point and the text contour point, respectively. $\min(\cdot)$ and $\|\cdot\|_2^2$ represent the minimization operation and Euclidean distance, respectively. Although the concentric mask has improved compared to the shrink mask, it still has some drawbacks: 1) It utilizes the barrel effect to avoid truncating some instances, but this leads to some concentric masks being almost identical to the text region, losing the significance of shrinking. 2) The shrinkage amount is the same in all directions, failing to adequately preserve the original geometric features of the instances.

3) Flexible Mask: Different from the aforementioned methods, the flexible mask adaptively shrinks to different extents based on the varying local geometric features of instances. Each instance is annotated by n points, and the generation of the flexible mask is shown in Fig. 3. Specifically, the generation process can be formulated as follows.

$$d_{fs}^i = \mathcal{D}(p^i, p^{n-i+1}) \times \sigma, \quad i = 1, ..., \frac{n}{2}, \tag{6}$$

where n is the number of points. d_{fs}^i and $\mathcal{D}(p^i, p^{n-i+1})$ represent i-th shrink distance and the horizontal and vertical distances from point p^i to point p^{n-i+1}. p^i and σ represent the coordinate of the i-th point and corresponding ratio. Horizontal instances are labeled starting from the top-left corner and proceeding clockwise. The vertical text starts from the top-right corner. The flexible mask can be described as follows:

$$p_{fs}^i = p^i + d_{fs}^i, \quad i = 1, 2, ..., \frac{n}{2}, \tag{7}$$

$$p_{fs}^{n-i+1} = p^{n-i+1} - d_{fs}^i, \quad i = 1, 2, ..., \frac{n}{2}, \tag{8}$$

$$p_{fs}^i = \frac{p_{fs}^i + p_{fs}^{i+1}}{2}, \quad i = 1, \frac{n}{2}+1, \tag{9}$$

$$p_{fs}^i = \frac{p_{fs}^i + p_{fs}^{i-1}}{2}, \quad i = \frac{n}{2}, n \tag{10}$$

where p_{fs}^i is the coordinate of the i-th point in flexible mask. The proposed FM has three advantages: 1) The shrink distance concerns the local contour and robustly represents the arbitrary-shaped text. 2) It shrinks distinct distances in different regions, which is more flexible than the shrink mask and the concentric mask. 3) It has significant differences from the text region, making it easy for the model to distinguish between the two and separate adjacent instances.

C. Magnetic Pull Module

Previous shrink-based methods are susceptible to the prediction of shrink masks. However, the shrink mask is an artificial geometry concept that enjoys incomplete semantic features and is difficult to detect accurately. To address this problem, we propose a magnetic pull module (MPM) that substantially reduces the reliance on flexible mask prediction and exhibits varying expansion in different directions, greatly enhancing flexibility compared to previous methods. Specifically, it includes two regression heads used to predict magnetic fields. As

Algorithm 1 Magnetic Field Calculation

Require: q instances with contours $P^i = \{p^{i,1}, \ldots, p^{i,n}\}$
Ensure: Magnetic field $B_x \in \mathbb{R}^{H\times W}$, $B_y \in \mathbb{R}^{H\times W}$

for $s = 1$ to q **do**
  Midpoint $p_m^i = \frac{1}{2}(p^i + p^{n-i+1}),\ i = 1$ to $n/2$
  for $k = 1$ to $n-2$ **do**
    if $k < n/2$ **then**
      Region $r^k \leftarrow \{p^k, p^{k+1}, p_m^k, p_m^{k+1}\}$
      Edge $e^k \leftarrow (p^k, p^{k+1})$
    else
      $j = n - k$; $r^k \leftarrow \{p^{k+1}, p^{k+2}, p_m^j, p_m^{j-1}\}$
      $e^k \leftarrow (p^{k+1}, p^{k+2})$
    end if
    if $k \in \{1, n/2-1, n/2, n-2\}$ **then**
      Target $\leftarrow p^k$
    else
      Target $\leftarrow e^k$
    end if
    for each point $(x, y) \in r^k$ **do**
      $\vec{v}$ = Target $- (x, y)$
      $B_x[y, x] = v^x, \quad B_y[y, x] = v^y$
    end for
  end for
end for

Algorithm 2 Text Contour Reconstruction

Require: Shrink mask P, magnetic fields B_x, B_y
Ensure: Reconstructed contours $\{C^i_{\text{recon}}\}_{i=1}^n$

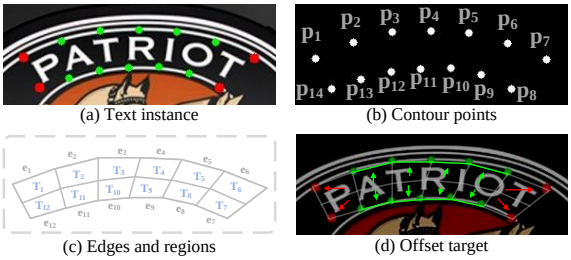
Extract instance masks $\{M^i\}_{i=1}^n$ from P
for $i = 1$ to n **do**
  Extract contour $C^i = \{(x^j, y^j)\}_{j=1}^{N_i}$ from M^i
  for $j = 1$ to N^i **do**
    $(x, y) \leftarrow (x^j, y^j)$
    $(\Delta x, \Delta y) \leftarrow (B^x[y, x], B^y[y, x])$
    $(x^j_{\text{recon}}, y^j_{\text{recon}}) \leftarrow (x + \Delta x, y + \Delta y)$
  end for
  $C^i_{\text{recon}} \leftarrow \{(x^j_{\text{recon}}, y^j_{\text{recon}})\}_{j=1}^{N_i}$
end for

shown in Fig. 4, the text region is divided into two parts, like the poles of the magnet. During the test stage, the pole points are generated by the Douglas-Peucker algorithm [38] based on the flexible mask prediction. The positive pole points of FM are pulled into the negative pole by magnetism, and the negative pole points are pulled to the positive pole, which is the process of obtaining pulling pole points to reconstruct text contour. It can be formulated as follows:

$$x_p^i = x^i + m_x^i, \quad i = 1, ..., t, \tag{11}$$

$$y_p^i = y^i + m_y^i, \quad i = 1, ..., t, \tag{12}$$

where x_p^i, y_p^i, x_i, and y_i represent the horizontal and vertical coordinates of i-th pulling pole points and pole points. m_x^i and m_y^i represent the magnetic field values of the horizontal and vertical axes at the i-th point. t is the number of pole points, which depends on the parameter ϵ of the Douglas-Peucker algorithm for curved datasets, which can be computed by "$\epsilon = 0.004 \times L$". L represents the perimeter of the flexible mask. The magnetic field denotes the distance on the horizontal and vertical axes between pixels and the corresponding edge or point. We denote the middle point p_m and edge e as:



(a) Text instance (b) Contour points (c) Edges and regions (d) Offset target

Fig. 5. The visualization of the magnetic field. p_i, e_i and T_i represent the i-th point, edge, region.

$$p_m^i = \frac{p^i + p^{n-i+1}}{2}, \quad i = 1, 2, ..., \frac{n}{2}. \tag{13}$$

$$\text{e}^i = \begin{cases} \mathcal{F}(p^i, p^{i+1}), & i < \frac{n}{2}, \\ \mathcal{F}(p^{i+1}, p^{i+2}), & i >= \frac{n}{2}, \end{cases} \tag{14}$$

where $\mathcal{F}(p^i, p^{i+1})$ represent the edge fromed by points p^i and p^{i+1}. As shown in Fig. 5, we divide the text as n-2 regions. As we can see from Fig. 5(d), the magnetic field in the middle region points towards the corresponding edges, while the magnetic field in the end regions points towards the corresponding points. Each region T is enclosed with 2 contour points and 2 middle points, which can be described as follows:

$$\text{T}^i = \begin{cases} \mathcal{E}(p^i, p^{i+1}, p_m^i, p_m^{i+1}), & i < \frac{n}{2} \\ \mathcal{E}(p^{i+1}, p^{i+2}, p_m^{n-i}, p_m^{n-i-1}), & i >= \frac{n}{2} \end{cases} \tag{15}$$

where T^i and $\mathcal{E}()$ represent i-th region and "enclose" operation, respectively.The ground truth of the magnetic magnitude can be formulated as:

$$\hat{m}^i = \begin{cases} \delta(P^i, e^j), & P^i \in \text{T}^j, j \neq 1, \frac{n}{2}-1, \frac{n}{2}, n-2 \\ \mathcal{D}(P^i, p^k), & P^i \in \text{T}^k, k = 1, \frac{n}{2}-1, \frac{n}{2}, n-2 \\ (0, 0), & \text{otherwise}, \end{cases} \tag{16}$$

where $\delta(P^i, e^j)$ denotes the horizontal and vertical distances from point P^i to edge e^j, $\mathcal{D}(P^i, p^k)$ represents the horizontal and vertical distances from point P^i to point p^k. Note that the size of $\hat{m}$ is $H \times W \times 2$, where H and W are the height and width of the image. For better clarity, the pseudocode of the magnetic field generation algorithm and the text contour reconstruction algorithm is provided in Algorithm 1 and Algorithm 2, respectively. In addition, we further discuss the differences between the proposed magnetic pull module and existing post-processing methods. As illustrated in Fig. 6, DBNet [7] solely relies on predictions for post-processing. When encountering overly small or irregular predictions, it fails to accurately reconstruct text contours. ADNet [31] and RSMTD [32] address this issue by adaptively predicting the expansion distance, effectively mitigating the first problem but

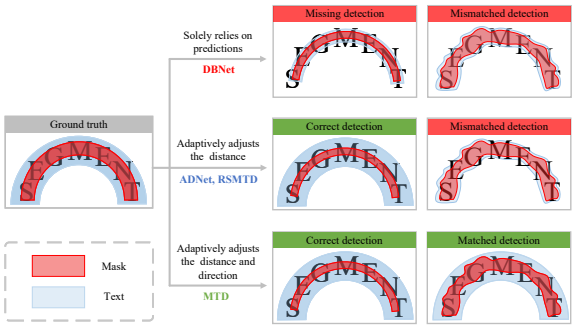


Fig. 6. Illustration of essential differences between existing methods.

unable to handle some irregular predictions. The MTD builds on this by adaptively predicting both the expansion direction and distance, thereby effectively avoiding the above problems.

D. Loss Function

To optimize our method, a multi-factor constraint loss is proposed. It includes magnetic region prediction loss $\mathcal{L}_{mr}$, flexible mask prediction loss $\mathcal{L}_{fm}$, magnetic magnitude on the horizontal axes and vertical axes prediction loss $\mathcal{L}_{mx}$ and $\mathcal{L}_{my}$. The magnetic region prediction head adopts the dice loss, which can be described as follows:

$$\mathcal{L}_{mr} = 1 - \frac{2 \times \sum(T_y \times T_x)}{\sum T_y^2 + \sum T_x^2 + \theta}, \tag{17}$$

where T_y and T_x represent the prediction and ground truth of the magnetic region, respectively. θ is a minimal value to avoid zero denominators. For the flexible mask prediction head, it applies a binary cross-entropy (BCE) loss. Hard negative mining [39] is adopted to alleviate the imbalance of the positive and negative samples. The $\mathcal{L}_{fm}$ can be formulated as:

$$\mathcal{L}_{fm} = \sum_{i \in S} -y_i \times log(x_i) - (1 - y_i) \times log(x_i), \tag{18}$$

where S represents the selected sample set. y_i and x_i are the ground truth and prediction of the flexible mask map.

For magnetic magnitude prediction heads, the L1 loss is utilized to optimize them, which can be represented as follows:

$$\mathcal{L}_{mx} = \sum |\hat{m}_x - m_x| \times M_{mag}, \tag{19}$$

$$\mathcal{L}_{my} = \sum |\hat{m}_y - m_y| \times M_{mag}, \tag{20}$$

$${M_{mag}}^i = \begin{cases} 1, & i \in \sum \mathrm{I}^j, j = 1, 2, ..., N \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

where $\hat{m}_x$, $\hat{m}_y$, m_x, and m_y are the ground truth and prediction of horizontal and vertical axes of the magnetic magnitude. The N and M_{mag} represent the number of the instance and the magnetic mask. I^j represent the j-th instance. The total loss function $\mathcal{L}$ is a weighted sum of the above loss, which can be described as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{fm} + \lambda_2 \mathcal{L}_{mr} + \lambda_3 \mathcal{L}_{mx} + \lambda_4 \mathcal{L}_{my}, \tag{22}$$

where λ_1, λ_2, λ_3 and λ_4 are the weight coefficient for $\mathcal{L}_{fm}$, $\mathcal{L}_t$,$\mathcal{L}_{ox}$ and $\mathcal{L}_{oy}$, respectively. The λ_1, λ_2, λ_3 and λ_4 are set to 6, 3, 0.1, 0.1 by the numeric values of corresponding loss.

E. Inference

The post-progressing includes three steps. (1) The flexible mask map is binarized. (2) The connected regions and their pole points are extracted through some morphological operations [40] and the Douglas-Peucker algorithm [38]. (3) The pole points are offset to rebuild the instance by the predictions of the magnetic magnitude, which is shown in Fig. 4.

IV. Experiment

A. Datasets

SynthText150K [20] is a synthetic dataset composed of 150k images. This dataset is used solely for pre-training the proposed model to enhance its robustness.

MSRA-TD500 [41] is a multi-oriented dataset that includes 300 training images and 200 testing images. Following previous methods, we introduce HUST-TR400 [42] for training.

CTW-1500 [43] primarily consists of arbitrary-shaped text instances. The dataset includes 1000 training images and 500 testing images, and each text instance is labeled with 14 points.

ICDAR2015 [44] is a word-level annotated dataset, with each image fixed at a scale of 1280 × 720.

Total-Text [45] is a word-level dataset that contains horizontal text, skewed text, and text with irregular shapes. It includes 1,255 training images and 300 testing images.

ASAYAR-TXT [46] is a subset of ASAYAR. The text in this subset is mainly Arabic or French, and the images are primarily sourced from highways in Mexico.

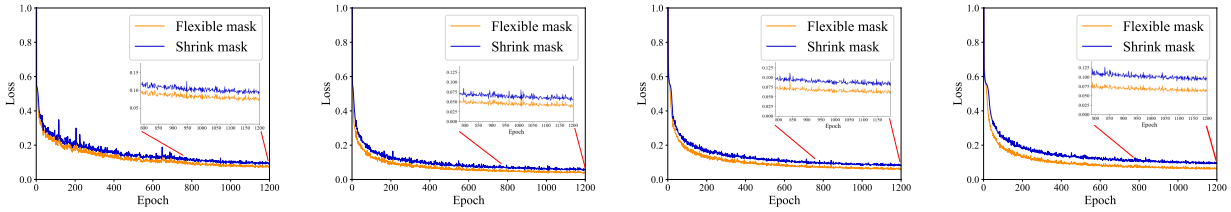
MPSC [47] is an industrial text dataset featuring low-contrast, low-brightness backgrounds and uneven surfaces. It consists of 2,555 training images and 639 testing images.

B. Implementation details

ResNet [49] with deformable convolution [50] is selected as the backbone. We adopt two training strategies: (1) train the model on the real-world dataset with 1000 epochs directly (used to ablation study). (2) pre-train the model on the SynthText150K dataset for 10 epochs and then fine-tune it for 660 epochs on the real-world dataset (except MSRA-TD500 and MPSC for 800 and 400 epochs). The batch size is set to 16. When using no extra datasets to pre-train, the stochastic gradient descent (SGD) with a weight decay of 0.0001 and a momentum of 0.9 is adopted. As for the fine-tuning stage, we utilize the Adam [51] optimizer. The initial learning rate for ResNet18 is set to 0.001. For ResNet50, the initial learning rate is set to 0.0001. Additionally, the "poly" learning rate strategy is applied, where the current learning rate is calculated as the initial learning rate multiplied by $(1 - \frac{iter}{iter_{max}})^{0.9}$. To augment the data, we employ random rotation, cropping, and flipping [10]. All the speed of experiments is tested on a single 1080Ti with an i7-6800K.

TABLE II
ABLATION STUDY ON THE EFFECT OF FLEXIBLE MASK AND MAGNETIC PULL MODULE ON DETECTION PERFORMANCE ON THE MSRA-TD500 AND CTW1500, WHERE "NET" AND "POST" REPRESENT THE TIME CONSUMPTION OF THE NETWORK AND POST-PROCESSING.

Methods	MSRA-TD500						CTW1500					
	P	R	F	FPS	Net	Post	P	R	F	FPS	Net	Post
Shrink mask + Vatti clipping	86.0	78.0	81.8	63.3	11.9	4.0	87.6	80.4	83.8	69.0	9.2	5.8
Shrink mask + Magnetic pull	91.3	79.6	85.0	69.7	12.3	2.2	88.3	81.3	84.7	80.7	9.5	3.0
Concentric mask + Magnetic pull	89.0	77.5	82.8	69.9	12.2	2.2	86.6	78.5	82.3	81.0	9.6	2.9
Flexible mask + Magnetic pull	90.5	83.2	86.7	70.2	12.2	2.1	87.9	84.1	85.9	82.7	9.5	2.7



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

Fig. 7. The loss of the shrink mask and the proposed flexible mask on the ICDAR2015, MSRA-TD500, Total-Text, and CTW1500, respectively.

TABLE III
DETECTION RESULTS OF DIFFERENT SHRINK RATIOS σ.

	σ	MSRA-TD500			CTW1500		
		P	R	F	P	R	F
MTD with	0.15	85.9	81.7	86.0	85.9	81.7	83.7
	0.2	90.3	81.4	85.6	86.9	82.9	84.8
	0.25	90.5	83.2	86.7	87.5	84.1	85.8
	0.3	91.3	81.6	86.2	87.9	84.1	86.0
	0.35	91.6	82.1	86.6	87.8	83.8	85.8

TABLE IV
DETECTION RESULTS ON DIFFERENT HEADS. CONV AND CONVT REPRESENT THE CONVOLUTION OPERATION AND THE TRANSPOSE CONVOLUTION OPERATION. $\mathrm{Up}_{\times \mathrm{n}}$ REPRESENT N TIMES UPSAMPLE.

Prediction head	MSRA-TD500				CTW1500			
	P	R	F	FPS	P	R	F	FPS
Conv + $\mathrm{Up}_{\times 4}$	89.2	82.5	85.7	74.1	87.3	84.2	85.7	86.2
Conv + ConvT + $\mathrm{Up}_{\times 2}$	91.0	83.2	86.9	72.0	89.3	82.7	85.9	84.2
Conv + ConvT $\times$ 2	90.5	83.2	86.7	70.2	87.9	84.1	85.9	82.7

TABLE V
COMPARISON UNDER DIFFERENT PRE-TRAINING SETTINGS

Dataset	MSRA-TD500			CTW1500			TotalText			IC15		
	P	R	F	P	R	F	P	R	F	P	R	F
800k [48]	92.6	86.1	89.2	88.3	84.1	86.2	89.1	83.0	86.0	88.5	79.2	83.6
150k [20]	93.4	88.8	89.1	90.4	82.9	86.5	90.5	82.7	86.4	87.7	80.5	84.0

C. Ablation Study

To demonstrate the effectiveness of the proposed model, we conduct ablation studies on CTW1500 and MSRA-TD500, respectively. In addition, all abstudy experiments adopt ResNet18 as the backbone.

1) Magnet pull module: As shown in Table II, for the MSRA-TD500, the MPM brings 3.2% improvement in F-measure, while saving 45% of the post-processing time. For the dataset that includes irregular-shaped texts, the MPM achieves 0.9% gains on the CTW1500 dataset while saving 50% of the post-processing time. The above experiments strongly demonstrate the superiority of the proposed MPM.

2) Flexible mask: As we can see from Table II, it brings 1.7% and 3.9% improvements in F-measure on the MSRA-TD500 dataset compared to the shrink mask and concentric mask. For the irregular-shaped dataset CTW1500, the F-measure is increased by 1.2% and 3.6% compared to the shrink mask and concentric mask after adopting the flexible mask. The detailed experiment discussion proves the effectiveness of the flexible mask. To further verify the superiority of the flexible mask, we conducted experiments to predict both the shrink mask and the flexible mask. The loss functions are illustrated in Fig. 7. As shown, the loss for the flexible mask consistently remains significantly lower than that of the shrink mask on the four public benchmarks. This demonstrates that the flexible mask is more learnable for the model and aligns better with both human and model intuitions.

3) Shrink ratio σ: As shown in Table IV, the CTW1500 dataset is very sensitive to the shrink ratio σ. When it is too low, performance drops significantly. Specifically, compared to 0.3, setting it to 0.15 results in decreases of 1.6%, 2.4%, and 2.1% in precision, recall, and F-measure, respectively. For the TD500 datasets, performance does not vary significantly with changes in the shrink ratio σ. We refer to the above experiments to set the shrink ratio in subsequent experiments.

4) Prediction head: Due to the simultaneous presence of positive and negative values in the magnetic field, we explore the setup of prediction heads. As shown in Table IV, for the irregular dataset CTW1500, as the number of deconvolution layers decreases and the number of bilinear interpolation layers increases, the FPS gradually improves, while performance is only slightly affected, with a mere 0.2% difference. For the MSRA-TD500, the inference speed also increases progressively. However, the precision, recall, and F-measure are decreased by 0.7%, 1.7%, and 1.0%, respectively.

5) Pre-training: To evaluate the impact of different pre-training datasets on the proposed MTD, we conduct experiments using SynthText [48] and SynthText150k [20] for pre-training. As shown in Table V, the proposed MTD exhibits low sensitivity to the choice of pre-training data, with only

TABLE VI
ABLATION STUDY ON THE EFFECT OF ϵ, WHERE MAX, MIN, AND AVG. REPRESENT THE MAXIMUM, MINIMUM, AND AVERAGE NUMBER OF POINTS.

ϵ	CTW1500							TotalText						
	P	R	F	Max	Min	Avg.	FPS	P	R	F	Max	Min	Avg.	FPS
None	90.4	83.1	86.6	1584	4	162.2	80.3	90.3	82.6	86.3	1869	4	92.8	74.1
$0.001\times L$	90.4	83.1	86.6	252	4	76. 7	80.3	90.3	82.6	86.3	251	4	67.3	74.5
$0.002\times L$	90.4	83.1	86.6	131	4	35.9	81.0	90.3	82.6	86.3	132	4	44.0	73.6
$0.004\times L$	90.4	82.9	86.5	74	4	17.5	81.7	90.5	82.6	86.4	71	4	23.5	74.7
$0.006\times L$	90.2	82.7	86.2	48	4	12.4	82.6	90.4	82.5	86.2	48	4	15.8	74.4
$0.01\times L$	89.8	80.6	85.0	34	4	8.6	82.8	90.0	82.3	86.0	30	4	10.1	75.5

TABLE VII
COMPARISON WITH EXISTING STATE-OF-THE-ART (SOTA) REAL-TIME METHODS.

Methods	Venue	Back.	CTW1500				TotalText				MSRA-TD500				ICDAR2015			
			P	R	F	FPS	P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
PAN [8]	ICCV'19	Res18	86.4	81.2	83.7	39.8	89.3	81.0	85.0	39.6	84.4	83.8	84.1	30.2	84.0	81.9	82.9	26.1
DBNet [7]	AAAI'20	Res18	84.8	77.5	81.0	55	88.3	77.9	82.8	50	90.4	76.3	82.8	62	86.8	78.4	82.3	48
CT [33]	NeurIPS'21	Res18	88.3	79.9	83.9	40.8	90.5	82.5	86.3	40.0	90.0	82.5	86.1	34.8	-	-	-	-
PAN++ [36]	TPAMI'22	Res18	87.1	81.1	84.0	36.0	89.9	81.0	85.3	38.3	85.3	84.0	84.7	32.5	85.9	80.4	83.1	28.2
CM-Net [3]	TIP'22	Res18	86.0	82.2	84.1	50.3	88.5	81.4	84.8	49.8	89.9	80.6	85.0	41.7	86.7	81.3	83.9	34.5
HFENet [52]	TITS'23	Res18	85.1	81.2	83.1	32.2	85.7	81.7	83.7	22.0	89.7	81.1	85.2	40.9	-	-	-	-
FS [1]	TIP'23	Res18	84.6	77.7	81.0	35.2	85.8	77.0	81.1	33.5	90.0	80.4	84.9	35.5	88.1	78.8	83.2	15.3
RSMTD [32]	TMM'23	Res18	87.8	80.3	83.9	72.1	88.5	83.8	86.1	70.9	89.8	83.1	86.3	62.5	-	-	-	-
DBNet++ [30]	TPAMI'23	Res18	86.7	81.3	83.9	40	87.4	79.6	83.3	48	87.9	82.5	85.1	55	90.1	77.2	83.1	44
ZTD [53]	TNNLS'24	Res18	88.4	80.2	84.1	76.9	90.1	82.3	86.0	75.2	91.6	82.4	86.8	59.2	87.5	79.0	83.1	48.3
FEPE [34]	TMM'25	Res18	88.8	83.0	85.5	55	90.8	79.5	84.8	50	89.4	82.8	86.0	62	87.3	79.4	83.2	48
MTD	**Ours**	Res18	90.4	82.9	**86.5**	81.7	90.5	82.7	**86.4**	74.7	93.4	85.2	**89.1**	69.3	87.7	80.5	**84.0**	60.7

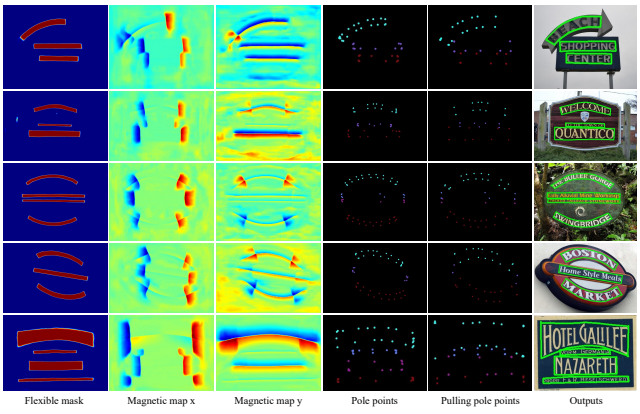


Fig. 8. Some visualization detection results for scene texts. From left to right, the sequence includes the flexible mask, magnetic magnitude along the horizontal and vertical axes, pole points, pulling pole points, and outputs.

minor differences in performance, indicating its robustness under varying pre-training conditions.

6) Contour coefficient ϵ: Contour coefficient ϵ influences the number of contour points significantly. As we can see from Table VI, when the maximum number of contour points is reduced from nearly 2,000 to just dozens, the model's performance remains almost unchanged, and the impact on speed is minimal. These experiments demonstrate that the proposed method is not sensitive to the number of extracted contour points, indirectly illustrating its superiority.

D. Comparisons with previous methods

To show more detail and the superiority of the proposed module, we compared it with previous methods on multiple public benchmarks. The results on datasets ASAYAR-TXT and MPSC are presented in the Supplemental Material.

1) Evalution on CTW1500: As shown in Table VII, the proposed MTD-ResNet18 achieves SOTA performance among real-time methods. Specifically, it achieved 90.4%, 82.9%, and 86.5% in precision, recall, and F-measure, respectively. With a simple network and efficient post-processing, it achieves an inference speed of 81.7 FPS. It outperforms the existing state-of-the-art methods DBNet++ [30] and ZTD [53] by 2.6% and 2.4% in F-measure, respectively. Additionally, it surpasses most non-real-time methods, such as LeafText [26], KPN [61], and LEMNet [58]. As shown in Table VIII, when adopting ResNet50 as the backbone, benefiting from the flexible magnetic reconstruction of MPM, the proposed MTD achieves suboptimal detection performance while maintaining a competitive inference speed.

2) Evalution on Total-Text: As shown in Table VII, the proposed method outperforms existing state-of-the-art real-time methods. Existing real-time methods DBNet++ [30], RSMTD [32], and PAN++ [36] achieve F-measures of 83.3%, 86.1%, and 85.3%, respectively. Our method surpasses them by 3.1%, 0.3%, and 1.1% while maintaining the high inference speed, respectively. As shown in Table VIII, although the MTD-ResNet50 is slower than TexBPN++ and ERRNet, it achieves 90.1%, 87.1%, and 88.6% in precision, recall, and F-measure, which outperforms most existing advanced methods and enjoys a competitive inference speed.

3) Evalution on MSRA-TD500: As we can see from Table VII, the proposed MTD-ResNet18 significantly outperforms existing real-time methods. It surpasses DBNet++ [30], ZTD [53], and RSMTD [32] by 4.0%, 2.3%, and 2.8% in the F-measure, respectively. In addition, it even outperforms most non-real-time methods. MTD-ResNet18 achieves a fast speed

TABLE VIII
COMPARISON WITH EXISTING NON-REAL-TIME STATE-OF-THE-ART METHODS. CBNET USES RES18 AS THE BACKBONE FOR CTW, TOTAL, AND TD500, AND RES50 FOR IC15. LEAFTEXT ADOPTS RES18 FOR CTW/TOTAL AND RES50 FOR IC15/TD500.

Methods	Venue	Back.	CTW1500				TotalText				MSRA-TD500				ICDAR2015			
			P	R	F	FPS	P	R	F	FPS	P	R	F	FPS	P	R	F	FPS
TextField [2]	TIP'19	VGG16	83.0	79.8	81.4	-	81.2	79.9	80.6	-	87.4	75.9	81.3	-	84.3	83.9	84.1	1.8
MTTD [4]	TIP'20	Res50	79.7	79.0	79.4	-	79.1	74.5	76.7	-	85.7	81.1	83.3	-	87.6	86.6	87.1	
OPMP [54]	TMM'21	Res50	85.1	80.8	82.9	1.4	87.6	82.7	85.1	1.4	86.0	83.4	84.7	1.6	89.1	85.5	87.3	1.4
FCE [19]	CVPR'21	Res50	87.6	83.4	85.5	-	89.3	82.5	85.8	-	-	-	-	-	90.1	82.6	86.2	-
DText [55]	PR'22	Res50	86.9	82.7	84.7	-	90.5	82.7	86.4	-	87.9	83.1	85.4	-	88.5	85.6	87.0	-
I3CL [56]	IJCV'22	Res50	88.4	84.6	86.5	-	89.8	84.2	86.9	-	-	-	-	-	-	-	-	-
EMA [57]	TIP'22	DLA34	86.1	82.1	84.1	-	88.2	83.3	85.6	-	88.7	81.1	84.7	-	89.4	82.4	85.8	-
LEMNet [58]	TMM'22	Res50	86.6	83.8	85.2	-	89.9	85.4	87.6	-	85.6	84.8	85.2	-	88.3	85.9	87.1	
HFENet [52]	TITS'23	Res50	88.1	83.4	85.7	18.1	89.0	84.0	86.4	12.2	92.8	84.0	88.2	21.4	-	-	-	-
TextDCT [59]	TMM'23	Res50	85.0	85.3	85.1	17.2	87.2	82.7	84.9	15.1	88.9	86.8	87.5	17.2	88.9	84.8	86.8	7.5
DBNet++ [30]	TPAMI'23	Res50	87.9	82.8	85.3	26	88.9	83.2	86.0	28	91.5	83.3	87.2	29	90.9	83.9	87.3	10
RP-Text [60]	TMM'23	Res18	87.8	81.6	84.7	-	89.4	82.8	86.0	-	88.4	84.6	86.5	-	89.6	82.4	85.9	-
KPN [61]	TNNLS'23	Res50	84.4	84.2	84.3	16.3	88.7	85.6	87.1	15.0	-	-	-	-	88.3	84.8	87.4	6.3
FS [1]	TIP'23	Res50	85.3	82.5	83.9	25.1	88.7	79.9	84.1	24.3	89.3	81.6	85.3	25.4	89.8	82.7	86.1	12.1
MorphText [62]	TMM'23	Res50	90.0	83.3	86.5	-	90.6	5.2	87.8	-	90.7	83.5	87.0	-	-	-	-	-
LeafText [26]	TMM'23	Res18/50	87.1	83.9	85.5	-	90.8	84.0	87.3	-	92.1	83.8	87.8	-	88.9	82.9	86.1	-
ADNet [31]	TMM'23	Res50	88.2	83.1	85.6	-	90.6	84.4	87.4	-	92.0	83.2	87.4	-	92.5	83.7	87.9	-
SMNet [63]	TITS'24	Res50	-	-	-	-	-	-	-	-	91.0	86.8	88.8	23.1	89.7	85.5	87.6	8.9
TPPAN [64]	TCSVT'24	Res50	88.7	86.3	87.5	-	91.2	85.0	88.0	-	93.4	88.2	90.7	-	90.7	86.8	**88.7**	-
VTD [65]	TCSVT'24	Res50	-	-	-	-	-	-	-	-	89.2	81.5	85.2	-	88.5	85.8	87.1	-
TTDNet [66]	TITS'24	Res50	87.4	82.2	84.7	4.6	-	-	-	-	90.4	83.9	87.0	-	90.0	85.6	87.7	-
LRANet [27]	AAAI'24	Res50	89.4	85.5	87.4	-	90.3	87.8	89.0	-	92.3	86.3	89.2	-	-	-	-	-
CBNet [25]	IJCV'24	Res18/50	89.0	81.9	86.0	-	90.1	82.5	86.1	-	91.1	84.8	87.8	-	91.0	85.4	88.1	-
TextBPN++ [10]	TMM'24	Res50	88.3	84.7	86.5	16.5	92.4	87.9	**90.1**	13.2	93.7	86.8	90.1	15.3	-	-	-	-
CT-Net [18]	TCSVT'24	Res50	88.5	83.8	86.1	11.2	90.8	85.0	87.8	10.1	90.8	84.4	87.5	11.6	90.9	86.4	88.6	6.5
EdgeText [28]	TCSVT'25	Res50	86.9	84.3	85.6	-	89.4	85.9	87.6	-	93.3	87.1	90.1	-	89.8	83.6	86.6	-
FEPE [34]	TMM'25	Res50	88.8	83.5	86.0	22	91.3	81.9	86.4	-	90.5	85.4	88.0	-	89.8	84.9	87.3	12
STD [67]	TMM'25	Res50	88.5	84.9	86.7	12.1	90.7	83.9	87.2	12.1	92.8	86.9	89.8	13.4	88.9	85.2	87.0	4.1
S3INet [68]	TNNLS'25	Res50	89.2	83.0	86.2	-	91.2	86.2	88.7	-	92.9	85.6	89.1	-	91.1	84.8	87.9	-
DCTNet [69]	TITS'25	Res50	87.8	85.5	86.6	-	89.2	87.6	88.4	-	90.5	84.6	87.5	-	-	-	-	-
ERRNet [24]	AAAI'25	Res50	90.1	87.9	**89.4**	-	92.6	87.3	89.9	-	93.8	87.1	90.3	-	-	-	-	-
MTD	**Ours**	Res50	90.2	87.1	88.6	35.4	90.1	87.1	88.6	25.2	94.6	88.0	**91.2**	24.6	90.8	86.2	88.5	11.3

Fig. 9. The visual comparison with existing methods.

of 69.3 FPS. As listed in Table VIII, the MTD-ResNet50 achieves 94.6%, 88.0%, and 91.2% in precision, recall, and F-measure, respectively. It surpasses the pervious method 5.0% (DB++-ResNet50 [30]), 0.9% (ERRNet [24]), and 5.9% (FS-ResNet50 [1]) on MSRA-TD500. The above experiments effectively demonstrate the superiority of the proposed MTD.

4) Evalution on ICDAR2015: As shown in Table VII, the MTD-ResNet18 significantly outperforms existing real-time methods, achieving precision, recall, and F-measure of 87.7%, 80.5%, and 84.0%, respectively. Specifically, it surpasses DBNet++ [30], FS [1], and HFENet [52] by 0.9%, 0.8%, and 0.8% in F-measure. Additionally, it achieves the fastest inference speed. As we can see from Table VIII, when using ResNet50 as the backbone, although the proposed MTD slightly underperforms TPPAN [64] (0.2%) and CT-Net [18] (0.1%), it still surpasses the vast majority of existing methods, such as DBNet++ [30], KPN [61], and ADNet [31]. Note that TPPAN does not mention the inference speed, and the proposed method achieves 11.3 FPS.

E. Comparisons of computational cost

To demonstrate the simplicity of our method, we compare the computational effort and parameters with previous real-time methods. As shown in Table IX, our method enjoys the lowest number of parameters among these real-time methods. For the CTW1500 dataset, the GFLOPs of MTD is about 40% and 24% of PAN [8] and DB [7], respectively. This is because DBNet uses a large size of input image and PAN uses a complex network. In addition, MTD surpasses PAN 2.8% in terms of F-measure. For the TotalText dataset, the GFLOPs of MTD is about 44.5%, 44.5%, 53.9%, and 47.4% of PAN [8], CT [33], DB [7], and DB++ [30]. The proposed MTD surpasses the above methods by 3.1%, 0.1%, 3.6%, and 3.1% in terms of F-measure. For MSRA-TD500, the proposed method also outperforms existing real-time approaches in terms of performance. In addition, our method requires only 50% of the GFLOPs and 68% of the parameters compared to the boundary-based method TextBPN++ [10] and achieves approximately twice the FPS. Overall, the computational cost primarily depends on the input image size and the model

TABLE IX
COMPARISONS OF COMPUTATIONAL COST AND PARAMETERS ON DIFFERENT REAL-TIME SCENE TEXT DETECTORS. "PARAMS" REPRESENT THE PARAMETERS OF MODELS. * DENOTES THE METHOD USING ICDAR2017MLT [70] TO PRE-TRAIN.

Method	Params(M)	CTW1500			TotalText			MSRA-TD500		
		GFLOPs	FPS	F-measure	GFLOPs	FPS	F-measure	GFLOPs	FPS	F-measure
PAN [8]	12.25	62.69	39.8	83.7	64.38	39.6	83.5	78.21	30.2	84.1
CT [33]	12.25	62.73	40.8	83.9	64.43	40.0	86.3	78.3	34.8	86.1
DB [7]	12.78	91.8	55	81.0	53.14	50	82.8	48.60	32	84.9
DB++ [30]	12.93	104.54	40	83.9	60.52	48	83.3	55.35	55	85.1
TextBPN++* [10]	17.53	59.1	35.3	85.7	59.2	32.5	84.5	52.15	34.6	**89.7**
Ours	**11.97**	**25.14**	**81.7**	**86.5**	**28.66**	**74.7**	**86.4**	**36.4**	**69.3**	89.1

TABLE X
COMPARISON OF PERFORMANCE AND FPS AT DIFFERENT RESOLUTIONS

Method	TotalText				CTW1500			
	512˜1024		640˜1024		512˜1024		640˜1024	
	F	FPS	F	FPS	F	FPS	F	FPS
PAN [8]	81.7	61	82.8	42	83.3	46	83.1	41
DB [7]	83.5	60	83.7	50	-	-	-	-
CT [33]	84.2	45	85.4	40	-	-	-	-
DB++ [30]	83.5	56	83.5	48	82.1	64	82.5	51
TextBPN++ [10]	84.4	48	85.6	45	82.7	57	84.5	49
Ours	86.0	81	86.4	74	85.0	96	86.5	82

(a) (b) (b) (d)

Fig. 10. The visualization of failure cases.

complexity. Furthermore, Table X compares the performance and inference speed (FPS) of different methods at a fixed input resolution. The above experiment results strongly demonstrate the superiority of the proposed MTD.

F. Visual analysis

To further demonstrate the superiority and effectiveness of the proposed MTD, we present the prediction of the flexible mask, magnetic magnitude, poly points, pulling pole points, and corresponding results in Fig. 8. To demonstrate the effectiveness of MTD, we present a visual comparison with existing methods in Fig. 9. As shown in Fig. 9(a) and (b), TextBPN++, FEPE, and DB++ fail to fully detect complete text instances, resulting in missed detections and truncated boundaries. In Fig. 9(c), TextBPN++ incorrectly merges multiple instances into a single region, while FEPE continues to miss certain instances. As illustrated in Fig. 9(d), both TextBPN++ and FEPE misidentify texture patterns that resemble text as true text instances. In Fig. 9(e) and (f), TextBPN++ fails to model both sides of elongated text instances accurately. In contrast,

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

Method	P	R	F	P	R	F
	MSRA → CTW			CTW → MSRA		
TextField [2]	75.3	70.0	72.6	85.3	75.8	80.3
CM-Net [3]	77.2	69.7	72.8	85.8	77.1	81.2
ZTD [53]	84.1	73.4	**78.4**	86.8	77.9	82.1
MTD-Res18	82.7	72.3	77.2	88.8	79.7	**84.1**
	IC15 → Total			Total → IC15		
TextField [2]	61.5	65.2	63.3	77.1	66.0	71.1
CM-Net [3]	75.8	64.5	69.7	76.5	68.1	72.1
ZTD [53]	78.5	64.1	70.6	79.8	69.3	74.2
MTD-Res18	78.7	74.2	**76.4**	82.0	71.6	**76.5**

benefiting from the magnetic field-based representation, our method accurately models both sides of long text instances. Furthermore, we visualize results on six public benchmarks in the supplemental material. Finally, we visualize several failure cases to analyze the limitations of our method. As shown in Fig. 10(a), certain background patterns with texture characteristics similar to text and located in close proximity to actual text regions are mistakenly identified as characters. Fig. 10(b) and Fig. 10(c) illustrate two additional types of errors: (b) missed detections of certain characters whose distinctive features deviate significantly from others within the same instance, and (c) incorrect instance separation caused by occlusions. These failure cases primarily stem from the segmentation-based approach's reliance on pixel-level optimization, which lacks sufficient modeling of instance-level features. As depicted in Fig. 10(d), inaccurate field predictions may result in slight deviations between the reconstructed results and the actual text contours. Future work could improve the magnet field representations to address this issue.

G. Cross dataset experiments

The above experiments prove the superiority of MTD. We further verify its generalization ability and robustness through cross-dataset experiments. Specifically, the module is first trained on MSRA-TD500 and tested on CTW1500. Although CTW1500 includes many irregular-shaped text instances and MSRA-TD500 without them, the proposed MTD achieves 77.2% (with ResNet18) in F-measure, which is a competitive result. The detection performance is superior to TextField [2] and CM-Net [3]. Then MTD is trained on CTW1500 and evaluated on MSRA-TD500. It achieves 84.1% (with ResNet18) that surpasses DB [7] 1.3%, which is directly training on MSRA-TD500. The above experiments verify

the shape robustness and generalization ability of MTD on different datasets. Additionally, we conducted experiments on Total-Text and ICDAR2015, which are word-level datasets. When the model is trained on ICDAR2015 and tested on Total-Text, the proposed MTD achieves 78.7%, 74.2%, and 76.4% in precision, recall, and F-measure, which is a competitive result. When swapping the training and testing datasets, the proposed method achieved 76.5% in F-measure, surpassing the existing SOTA method TextField [2], CMNet [3], and ZTD [53].

V. Conclusion

In this paper, we propose an effective real-time scene text detector that introduces several novel components to enhance detection accuracy and efficiency. A text representation flexible mask is proposed first, which has a clear distinction from the text region. It shrinks different distances on different positions, which can cope with some extreme instances. Then, we propose a magnetic pull module (MPM) that significantly reduces the deep reliance of the model on flexible mask prediction while saving about 50% of the post-processing time. It pulls the pole points of FM to the text contour to reconstruct instances accurately, even if the predictions deviate from the ground truth. Benefiting from the above module, our method achieves the trade-off between speed and performance when adopting a lightweight backbone, which is demonstrated by a series of experiments. In the future, we are interested in exploring an efficient scene text spotter.

References

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

[2] 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 Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, 2019.

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

[4] Y. Liu, L. Jin, and C. Fang, "Arbitrarily shaped scene text detection with a mask tightness text detector," *IEEE Trans. Image Process.*, vol. 29, pp. 2918–2930, 2020.

[5] S. Zhang, X. Zhu, C. Yang, H. Wang, and X. Yin, "Adaptive boundary proposal network for arbitrary shape text detection," in *Proc. IEEE Int. Conf. Comput. Vis.n*, 2021, pp. 1305–1314.

[6] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9336–9345.

[7] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[8] 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 *Proc. IEEE Int. Conf. Comput. Vis.n*, 2019, pp. 8440–8449.

[9] X. Han, J. Gao, Y. Yuan, and Q. Wang, "Text kernel calculation for arbitrary shape text detection," *The Visual Computer*, vol. 40, no. 4, pp. 2641–2654, 2024.

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

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

[12] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2017.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. ECCV*. Springer, 2016, pp. 21–37.

[14] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, 2018.

[15] 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.

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5551–5560.

[17] P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7393–7402.

[18] 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 Trans. Circuits Syst. Video Technol.y*, vol. 34, no. 3, pp. 1815–1826, 2024.

[19] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3123–3131.

[20] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9809–9818.

[21] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*. Springer, 2016, pp. 56–72.

[22] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2550–2558.

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

[24] Y. Su, Z. Chen, Y. Du, Z. Ji, K. Hu, J. Bai, and X. Gao, "Explicit relational reasoning network for scene text detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, no. 7, 2025, pp. 7069–7077.

[25] 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," *IJCV*, pp. 1–20, 2024.

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

[27] Y. Su, Z. Chen, Z. Shao, Y. Du, Z. Ji, J. Bai, Y. Zhou, and Y.-G. Jiang, "Lranet: towards accurate and efficient scene text detection with low-rank approximation network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 5, 2024, pp. 4979–4987.

[28] C. Yang, X. Han, T. Han, H. Han, B. Zhao, and Q. Wang, "Edge approximation text detector," *IEEE Trans. Circuits Syst. Video Technol.y*, pp. 1–1, 2025.

[29] 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 Trans. Circuits Syst. Video Technol.y*, vol. 32, no. 9, pp. 6073–6085, 2022.

[30] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, 2023.

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

[32] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Reinforcement shrink-mask for text detection," *IEEE Trans. Multimedia*, vol. 25, pp. 6458–6470, 2023.

[33] T. Sheng, J. Chen, and Z. Lian, "Centripetaltext: An efficient text instance representation for scene text detection," *Proc. Adv. Neural Inf. Process. Syst*, vol. 34, pp. 335–346, 2021.

[34] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Focus entirety and perceive environment for arbitrary-shaped text detection," *IEEE Trans. Multimedia*, vol. 27, pp. 287–299, 2025.

[35] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[36] 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.

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

[38] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *IJGIG*, vol. 10, no. 2, pp. 112–122, 1973.
[39] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. CVPR*, June 2016.
[40] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
[41] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE CVPR*. IEEE, 2012, pp. 1083–1090.
[42] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, 2014.
[43] 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.
[44] 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 *Proc. ICDAR*. IEEE, 2015, pp. 1156–1160.
[45] C. Ch'ng and C. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *ICDAR*, vol. 1. IEEE, 2017, pp. 935–942.
[46] 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 Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3026–3036, 2022.
[47] 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 Trans. Circuits Syst. Video Technol.y*, vol. 32, no. 9, pp. 6073–6085, 2022.
[48] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[50] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.
[51] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[52] 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 Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14 200–14 212, 2023.
[53] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Zoom text detector," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15 745–15 757, 2024.
[54] 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 Trans. Multimedia*, vol. 23, pp. 454–467, 2021.
[55] Y. Cai, Y. Liu, C. Shen, L. Jin, Y. Li, and D. Ergu, "Arbitrarily shaped scene text detection with dynamic convolution," *PR*, vol. 127, p. 108608, 2022.
[56] B. Du, J. Ye, J. Zhang, J. Liu, and D. Tao, "I3cl: intra-and inter-instance collaborative learning for arbitrary-shaped scene text detection," *IJCV*, vol. 130, no. 8, pp. 1961–1977, 2022.
[57] M. Zhao, W. Feng, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Mixed-supervised scene text detection with expectation-maximization algorithm," *IEEE Trans. Image Process.*, vol. 31, pp. 5513–5528, 2022.
[58] 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 Trans. Multimedia*, vol. 24, pp. 3129–3143, 2022.
[59] 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 Trans. Multimedia*, vol. 25, pp. 5030–5042, 2023.
[60] Q. Wang, B. Fu, M. Li, J. He, X. Peng, and Y. Qiao, "Region-aware arbitrary-shaped text detection with progressive fusion," *IEEE Trans. Multimedia*, vol. 25, pp. 4718–4729, 2023.
[61] S. Zhang, X. Zhu, J. Hou, C. Yang, and X. Yin, "Kernel proposal network for arbitrary shape text detection," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2022.
[62] C. Xu, W. Jia, R. Wang, X. Luo, and X. He, "Morphtext: Deep morphology regularized accurate arbitrary-shape scene text detection," *IEEE Trans. Multimedia*, vol. 25, pp. 4199–4212, 2023.
[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 Trans. Intell. Transp. Syst.*, pp. 1–13, 2024.
[64] 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 Trans. Circuits Syst. Video Technol.y*, vol. 34, no. 1, pp. 286–298, 2024.
[65] 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 Trans. Circuits Syst. Video Technol.y*, vol. 34, no. 6, pp. 4407–4420, 2024.
[66] 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 Trans. Intell. Transp. Syst.*, pp. 1–15, 2024.
[67] X. Han, J. Gao, C. Yang, Y. Yuan, and Q. Wang, "Spotlight text detector: Spotlight on candidate regions like a camera," *IEEE Trans. Multimedia*, vol. 27, pp. 1937–1949, 2025.
[68] R. Wang, H. Chen, Y. Zhu, J. Xu, X. Cao, Z. Zhu, S. Qian, C. Gao, L. Liu, and N. Sang, "S3inet: Semantic-information space sharing interaction network for arbitrary shape text detection," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2025.
[69] Z. Chen, "Arbitrary shape text detection with discrete cosine transform and clip for urban scene perception in its," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–9, 2025.
[70] 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 *Proc. ICDAR*, vol. 1. IEEE, 2017, pp. 1454–1459.

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.

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/).