

# Edge Approximation Text Detector

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

**Abstract**—Pursuing efficient text shape representations helps scene text detection models focus on compact foreground regions and optimize the contour reconstruction steps to simplify the whole detection pipeline. Current approaches either represent irregular shapes via box-to-polygon strategy or decomposing a contour into pieces for fitting gradually, the deficiency of coarse contours or complex pipelines always exists in these models. Considering the above issues, we introduce *EdgeText* to fit text contours compactly while alleviating excessive contour rebuilding processes. Concretely, it is observed that the two long edges of texts can be regarded as smooth curves. It allows us to build contours via continuous and smooth edges that cover text regions tightly instead of fitting piecewise, which helps avoid the two limitations in current models. Inspired by this observation, *EdgeText* formulates the text representation as the edge approximation problem via parameterized curve fitting functions. In the inference stage, our model starts with locating text centers, and then creating curve functions for approximating text edges relying on the points. Meanwhile, truncation points are determined based on the location features. In the end, extracting curve segments from curve functions by using the pixel coordinate information brought by truncation points to reconstruct text contours. Furthermore, considering the deep dependency of *EdgeText* on text edges, a bilateral enhanced perception (BEP) module is designed. It encourages our model to pay attention to the recognition of edge features. Additionally, to accelerate the learning of the curve function parameters, we introduce a proportional integral loss (PI-loss) to force the proposed model to focus on the curve distribution and avoid being disturbed by text scales. Ablation experiments demonstrate that *EdgeText* can fit scene texts compactly and naturally. Comparisons show that *EdgeText* is superior to existing methods on multiple public datasets. Code is available at <https://github.com/omtcyang/EdgeTD>

**Index Terms**—Scene text detection, text shape representation, edge approximation, parameterized modeling

## I. INTRODUCTION

SCENE text detector [1], [2], [3], a front component of the text spotting pipeline, is responsible for determining text locations and feeding for the following components to recognize texts. The design of an efficient text representation helps formulate compact instance contours to irregular scene texts with fewer procedures, which encourages the detection module to avoid bringing the noise into the recognition stage and optimizes the detection pipeline from complex contour

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

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

Tao Han is with Shanghai Artificial Intelligence Laboratory, Longwen Road 129, Xuhui District, 200232 Shanghai, China.

E-mail: omcyang@gmail.com, hxu04100@gmail.com, hanhao10200@gmail.com, 1356376210@qq.com, bxuanzhao202@gmail.com, crabwq@gmail.com.

Qi Wang is the corresponding author.

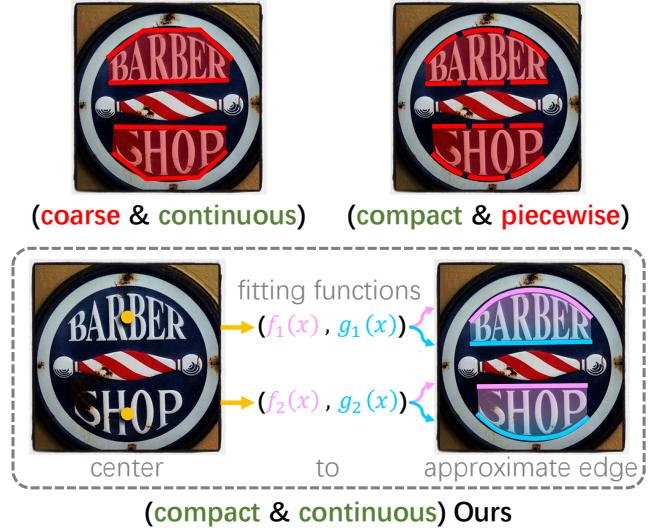


Fig. 1. Visualization of the differences between our edge approximation text representation method (bottom sub-figure) and the current popular representations of box-to-polygon strategy (left top sub-figure) and piecewise fitting process (right top sub-figure).

rebuilding processes. Therefore, **how to represent irregular scene texts efficiently** is important for text detection models.

Current popular practices are to represent irregular text shapes via box-to-polygon strategy [4] or decompose a contour into pieces for fitting gradually [5], [6], [7]. Specifically, the former strategy starts from rectangular box detection. It then samples a few points along two long sides of boxes and maps these points to text region boundaries. This strategy ensures intuitive polygon generation processes of irregular texts, but hard to fit compactly (such as the left top of Fig. 1), which may pass noises to the following components and make the recognition task difficult. Different from the box-to-polygon strategy, existing piecewise fitting methods are able to represent scene texts with compactness boundaries (the sub-figure at the right top of Fig. 1), but the gradual contour reconstruction process complicates the whole inference pipeline and the framework training optimization.

Following the above issues, we aim to represent scene texts compactly and continuously to avoid the limitations existing in the box-to-polygon and piecewise fitting methods. Concretely, it is observed that the text is a string with a series of characters (such as English letters, Arabic numerals, and Chinese, etc), which makes the scene text contour always can be seen as a ribbon-like shape that consists of two curves along the text's long edges. Therefore, the scene text contour can be reconstructed easily once the curves are determined. Based on the above observation, we formulate the irregular text representation problem as an approximation process of

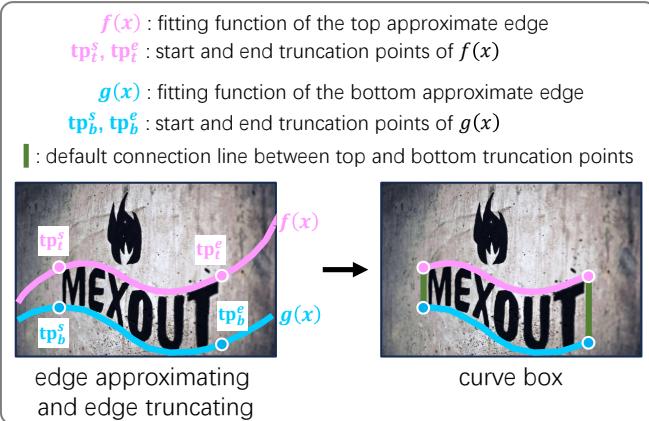


Fig. 2. Detail visualization of the proposed edge approximation text representation method. It fits texts with the curve box that is constructed based on the approximate edge and truncation points. The same as the traditional bounding box representation, our **curve box** enjoys a simple reconstruction process. Especially, it can fit irregular texts accurately instead of covering rectangular shapes only like the traditional bounding box.

parameterized curves toward text edges, and an end-to-end text detector framework, namely *EdgeText*, is constructed following the design of edge approximation text representation. In the inference stage (shown in Fig. 2), *EdgeText* starts with locating text centers based on the pixel-level foreground classification confidence map, then creating text edge approximation curves predefined as mathematical formula parameters by using the center's corresponding visual information. Meanwhile, determining truncation points defined in the image pixel coordinate system from text location features. Next, truncating the approximation curves with the points for extracting curve segments to reconstruct irregular text contours. As described before, the designed curve box relies on the curve fitting functions of approximate edges and truncation points deeply. To pursue accurate curve boxes in the reconstruction process of the inference stage, a bilateral enhanced perception (BEP) module is introduced. The module encourages *EdgeText* to focus on the two long truncated edges that are created from approximate edges, which helps our model improve the recognition of the corresponding edge features. Additionally, a proportional integral loss (PI-loss) is proposed. It treats the integral of the scaled difference between two curves as the optimization objective, which avoids the preference problem toward large-scale instances and helps improve the parameter learning of the curve fitting functions. The main contributions of this paper are as follows:

- 1) An edge approximation representation method is proposed following the scene text characteristics of ribbon-like shapes. It represents arbitrary-shaped texts via parameterized curve fitting functions and truncation points, which ensures compact and continuous rebuilt contours to achieve accurate text representation efficiently.
- 2) A bilateral enhanced perception (BEP) module is designed to improve the feature recognition ability toward text edges. It helps the model accurately perceive text edge features for generating reliable curve boxes.
- 3) A proportional integral loss (PI-loss) is introduced. It

treats the integral of the scaled difference between two curves as the optimization objective, which helps the model focus on the curve distribution and avoid being disturbed by text scales.

- 4) An efficient text detection framework (*EdgeText*) is constructed using the proposed edge approximation representation method, BEP, and PI-loss. It can fit arbitrary-shaped scene texts compactly with a simple and intuitive practice, which alleviates the limitations that exist in current popular models.

## II. RELATED WORK

The strong feature expression ability of deep learning networks progresses scene text detection models in the aspect of feature understanding of the complex background and text. To fit varied shapes of scene texts, existing methods focus on fitting text contours as accurately as possible. They can be divided into regular and irregular methods according to text shapes. In this section, we will introduce these methods briefly.

### A. Methods toward the Regular Text Shape

Regular-shaped texts denote the instances that can be fitted by rectangle and quadrilateral boxes accurately. Early text detectors [8], [9] are inspired by traditional two-stage object detection framework [10] to represent texts via anchor-to-box strategy. These methods start with extracting the text region of interest (RoI) based on predefined anchors and then refining the region vertex locations for generating regular text boxes. Especially, Liu *et al.* [8] and Ma *et al.* [9] introduced rotation anchors for fitting multi-oriented texts. Though they achieve comparable performance, the two-stage framework makes the pipeline complex. With the proposing of Single Shot MultiBox Detector (SSD) [11], one-stage text detection methods [12], [13], [14] inherit the corresponding design for alleviating the problem of the complex inference process. Different from the previous methods, the authors [15], [16], [17] abandoned the RoI refining process in the two-stage framework and regressed the vertex offsets between anchors and text boxes directly, which optimizes the overall detection pipeline effectively. Considering the anchor mechanism results in model performance deeply relies on prior knowledge of dataset statistics, researchers followed the anchor-free works [18], [19], [20] for avoiding the prior dependency problem that exists in those methods while further simplifying the whole detection process. Specifically, the anchor-free text detector [21] executes dense prediction toward the offset between the current pixel location and box vertices, which enjoys better generalization performance compared with previous methods. The above models achieve comparable detection results on public regular text detection datasets. However, with the appearance of irregular-shaped scene texts, these methods are hard to fit accurately.

### B. Methods toward the Irregular Text Shape

A valid representation design for irregular-shaped texts can help avoid feeding interference information to the text recognizer for improving the accuracy of recognition results. Initially, related methods [22], [23], [24] fit irregular

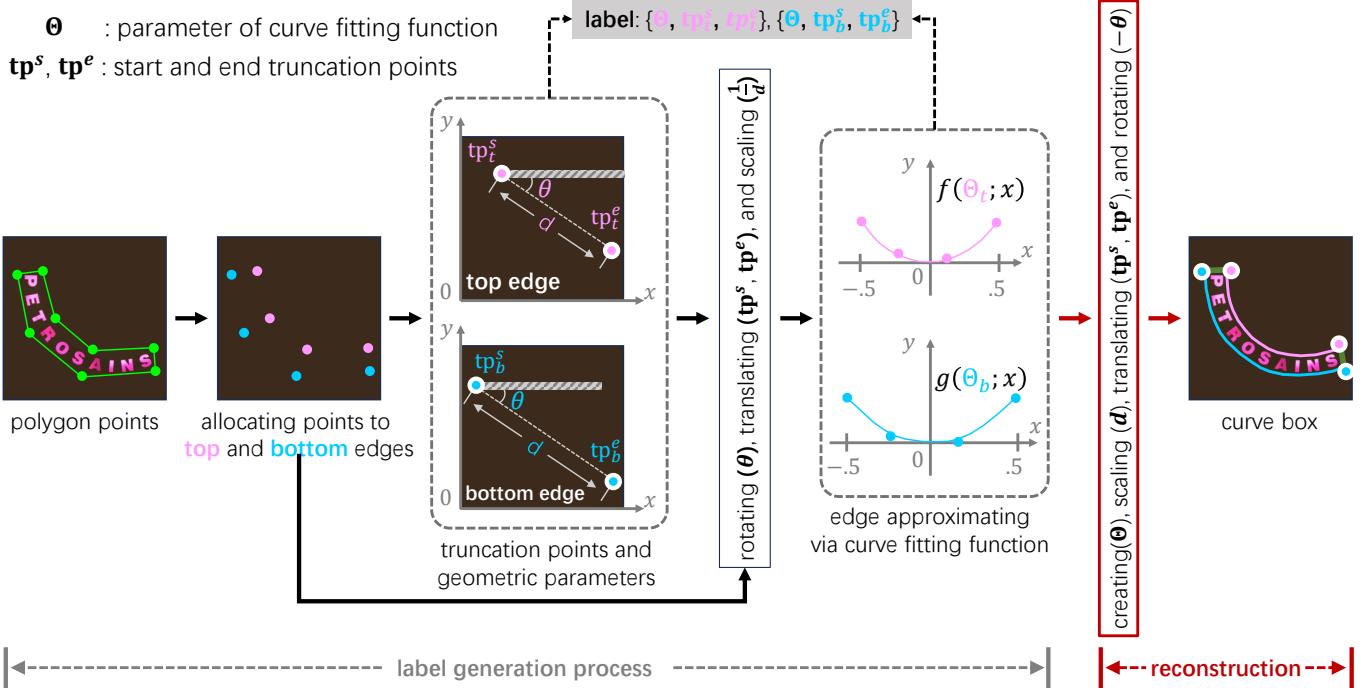


Fig. 3. Visualization of the processes of the label generation and curve box reconstruction.  $f(\Theta_t; x)$  and  $g(\Theta_b; x)$  are the same curve-fitting function except for different parameters. Polynomial is adopted as the curve fitting function in this paper.

texts based on the bounding box representation. Concretely, Zhang *et al.* [22] decomposed the text into a series of characters and detected them via traditional object detection methods. The final text contours can be obtained by linking character boxes. Some other authors [25], [26], [27] started with detecting the whole text box. Then, they sampled a few key points along the long sides of the box. Next, those points were mapped to text boundaries for fitting. In the end, text contours can be reconstructed by connecting mapped key points in a clockwise or counterclockwise direction. Besides the above detection-based representation design, an intuitive idea is to segment text regions [28], [29], [30] by using the strong pixel-level detection ability of the image segmentation techniques [31], [32]. However, the text adhesion or occlusion phenomenon always exists in natural scenes, it will lead to detect multiple instances as one if segmenting text regions directly. Considering the above issues, researchers reconstructed the segmentation-based detection pipeline [33], [34], [35], [36], [37]. Specially, Baek [38] adopted a character-connection strategy. The author segmented every char and grouped them together that belong to the same text for rebuilding text contours. Different from Baek [38], Zhu [39] and Liu [40] represented text contour via mathematical fitting functions (such as Fourier series and Bessel curve). However, the complex transform process from the corresponding fitting function to the text contour makes it hard to optimize the model and achieve text rebuilding via fewer steps. Current methods either are hard to cover irregular text regions compactly or rely on complex fitting processes, how to represent scene texts in an efficient manner is still under exploration.

### III. METHODOLOGY

In this section, we introduce the designed edge approximation-based text representation method first. Then, the overall constructed detection framework (EdgeText) is described combined with the pipeline figure. Next, the bilateral enhanced perception (BEP) module is visualized to illustrate the corresponding structure in detail. In the end, loss functions, especially the proposed proportional integral loss, are given to explain the model optimization process.

#### A. Edge Approximation Representation

Designing an efficient representation method for irregular-shaped scene texts can help cover text regions compactly with fewer procedures, which ensures reliable location information feeding for the following recognition task while keeping a simple inference pipeline. Considering current popular practices either are hard to cover texts compactly or rely on complex fitting processes, we proposed an edge approximation-based text representation method in this paper.

As mentioned in Fig. 2 before, the proposed representation method fits irregular texts via the curve box created through the two processes of edge approximating and edge truncating. As shown in Fig. 3, given a text instance labeled by a series of polygon points  $P = [p_1, p_2, \dots, p_{2k}]$ , where  $p_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ , which is the  $k$ -th point location in the horizontal pixel coordinate system.  $2k$  denotes the number of points. We first define the two long sides of the text as top and bottom edges respectively and allocate those polygon points to the two edges equally. The top edge points  $P_t = [p_1, p_2, \dots, p_k]$  and the bottom edge points  $P_b = [p_{k+1}, p_{k+2}, \dots, p_{2k}]$  can be obtained. Then,

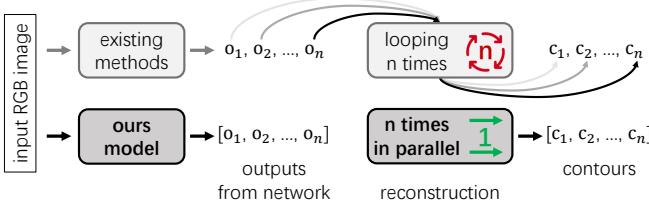


Fig. 4. Visualization of the contour reconstruction differences between existing methods and ours EdgeText. Compared with existing box-to-polygon strategy-based or piecewise fitting methods that have to reconstruct every single text one by one until all instance contours are generated, EdgeText rebuilds all text curve boxes in the input image in parallel, which enjoys a more intuitive contour rebuilding process with fewer procedures.

the parameter  $\Theta$  of the curve fitting function, and start and end truncation points ( $tp^s$  and  $tp^e$ ) are generated as the training labels, which are used for reconstructing text contours. **Here we take the top edge as an example to explain the generation process, and the label generation process of the bottom edge is the same as the top edge.** Concretely, the points  $p_1$  and  $p_k$  in  $P_t$  are picked as the truncation points ( $tp_t^s$  and  $tp_t^e$ ) of the top edge. And  $P_t$  are rotated, translated, and scaled according to truncation points for generating a series of transformed points  $P'_t$  of horizontal, symmetrical to the origin of the coordinate system, and the scaling to -0.5 to 0.5 on X-axis, which helps the model deviate from the interference of spatial distribution and focus on fitting curve shape of  $P_t$ . The rotating, translating, and scaling processes can be formulated by mathematical equations as follows:

$$\begin{aligned} P_t^R &= R_t(P_t - tp_t^s) = \begin{bmatrix} \cos \theta_t & -\sin \theta_t \\ \sin \theta_t & \cos \theta_t \end{bmatrix} (P_t - tp_t^s), \\ P_t^T &= P_t^R - \frac{1}{2}(P_{t,\frac{k-1}{2}}^R + P_{t,\frac{k}{2}}^R), \\ P'_t &= P_t^S = \frac{1}{d_t} P_t^T, \end{aligned} \quad (1)$$

where  $P_t^R$ ,  $P_t^T$ , and  $P_t^S$  indicate the points that are rotated, translated, and scaled based on  $P_t$ .  $R_t$  is the rotation matrix.  $\theta_t$  denotes the angle between the vector  $tp_t^s tp_t^e$  and X-axis.  $P_{t,\frac{k-1}{2}}^R$  and  $P_{t,\frac{k}{2}}^R$  are  $\frac{k-1}{2}$ -th and  $\frac{k}{2}$ -th points in  $P_t^R$ .  $d_t$  is the Euclidean distance between  $tp_t^s$  and  $tp_t^e$ . After transforming polygon points from  $P_t$  to  $P'_t$  through the three operations in Equation 1, the corresponding curve fitting function  $f(\Theta_t; x)$  that approximating text edge can be formulated, where  $\Theta_t$  is the parameter of curve fitting function that the model need to learn. For the text bottom edge, the corresponding truncation points  $tp_b^s$  and  $tp_b^e$ , transformed polygon points  $P_b$  and curve fitting function  $g(\Theta_b; x)$  can be obtained in the same way. **Notably**,  $f(\Theta_t; x)$  and  $g(\Theta_b; x)$  are the same curve-fitting function except for different parameters and the polynomial is adopted as the curve-fitting function in this paper:

$$\begin{aligned} f(\Theta_t; x) &= g(\Theta_b; x) = \sum_{i=1}^m \Theta_i x^i + c, \\ \Theta_t &= \{\Theta_{i,t}, c | i = 1, 2, \dots, m\}, \\ \Theta_b &= \{\Theta_{i,b}, c | i = 1, 2, \dots, m\}, \end{aligned} \quad (2)$$

where  $m$  denotes the highest degree of the polynomial and  $c$  represents the constant term.

In the inference stage, benefiting from the advantages brought by the edge approximation-based representation method and the above transformation operations in Equation 1, the reconstruction process of text contours with different rotated angles, spatial locations, and scales can be treated as a matrix manipulation about curve fitting function sampling points, which allows rebuilding all text contours in the input image in parallel. Compared with existing box-to-polygon strategy-based or piecewise fitting methods that have to reconstruct every single text one by one until all instance contours are generated, EdgeText enjoys a more intuitive contour rebuilding process with fewer procedures (as shown in Fig. 4). Specifically, with the predicted parameters  $\tilde{\Theta}_t$  and  $\tilde{\Theta}_b$  of the curve fitting functions  $f(\tilde{\Theta}_t; x)$  and  $g(\tilde{\Theta}_b; x)$ , and start and end truncation points, the transformed dense curve points  $\tilde{P}'$  can be obtained by following equations:

$$\tilde{P}' = \begin{bmatrix} \tilde{P}'_1 \\ \tilde{P}'_2 \\ \vdots \\ \tilde{P}'_n \end{bmatrix}, \quad \tilde{P}'_n = \begin{bmatrix} \tilde{P}'_{n,t} \\ \tilde{P}'_{n,b} \end{bmatrix} = \begin{bmatrix} X \\ f(\tilde{\Theta}_{n,t}; X) \\ X \\ g(\tilde{\Theta}_{n,b}; X) \end{bmatrix}, \quad (3)$$

$$X = [x_1, x_2, \dots, x_n], -0.5 \leq x \leq 0.5, n > k,$$

where  $n$  denotes the number of text instances in the input image,  $(\cdot)_{n,t}$  and  $(\cdot)_{n,b}$  are the predicted values of the top and bottom edges of the  $n$ -th text instance respectively.

With the transformed dense curve points  $\tilde{P}'$ , the final text contours can be generated by scaling, translating, and rotating operations, which enjoys inverse operation orders with label generation in Equation 1. The corresponding scaling process can be formulated as follows:

$$\tilde{P}'^S = \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \vdots \\ \tilde{d}_n \end{bmatrix} \odot \tilde{P}', \quad \tilde{d}_n = \begin{bmatrix} \tilde{d}_{n,t} \\ \tilde{d}_{n,b} \end{bmatrix}, \quad (4)$$

where  $\tilde{d}$  is the Euclidean distance computed by the predicted truncation points. With the scaling point matrix  $\tilde{P}'^S$ , translating point matrix  $\tilde{P}'^T$  can be obtained by:

$$\tilde{P}'^T = \tilde{P}'^S + (\begin{bmatrix} \tilde{tp}_1^s \\ \tilde{tp}_2^s \\ \vdots \\ \tilde{tp}_n^s \end{bmatrix} - \tilde{P}_{:,0}^S), \quad \tilde{tp}_n^s = \begin{bmatrix} \tilde{tp}_{n,t}^s \\ \tilde{tp}_{n,b}^s \end{bmatrix}, \quad (5)$$

where  $\tilde{tp}^s$  is the predicted start truncation point. Based on the above  $\tilde{P}'^T$ , the final point matrix  $\tilde{P}$  used for reconstructing text contours is computed by:

$$\tilde{P} = \tilde{P}'^R = \begin{bmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \vdots \\ \tilde{R}_n \end{bmatrix} \odot \tilde{P}'^T, \quad \tilde{R}_n = \begin{bmatrix} \tilde{R}_{n,t} \\ \tilde{R}_{n,b} \end{bmatrix}, \quad (6)$$

where  $\tilde{R}$  is a rotation matrix that can be referred the  $R_t$  in Equation 1, where the  $\theta$  is obtained by the predicted truncation points ( $\tilde{tp}^s$  and  $\tilde{tp}^e$ ) and X-axis.

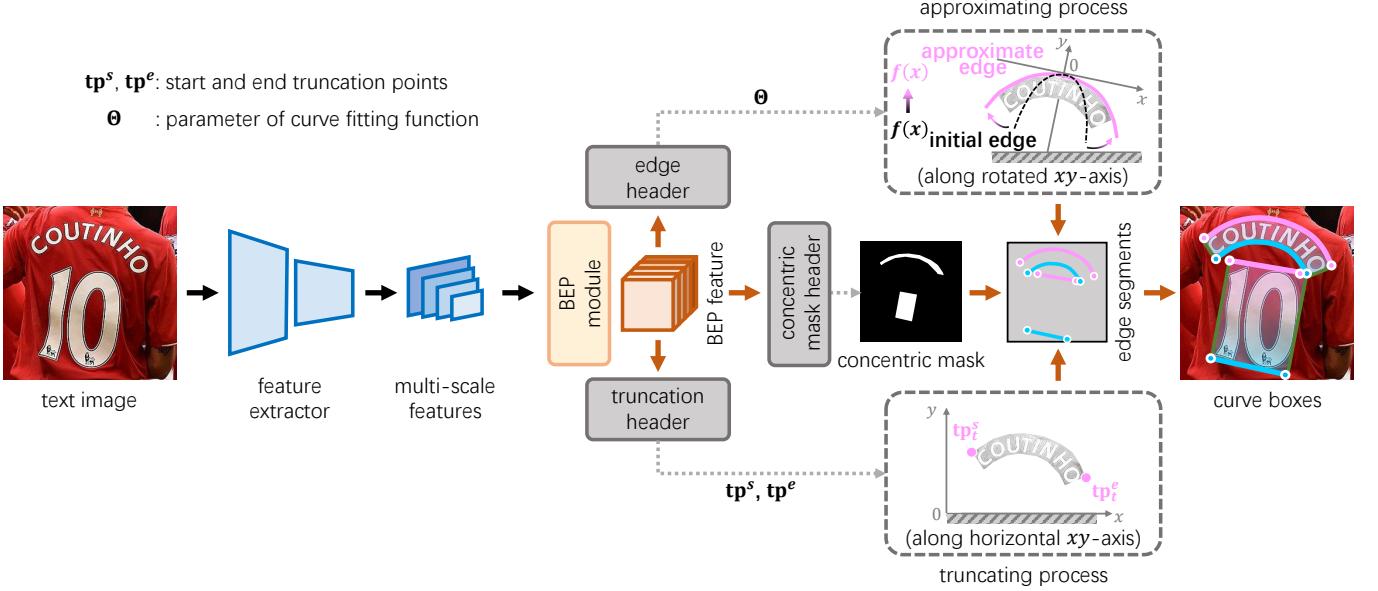


Fig. 5. Pipeline visualization of the constructed EdgeText, which is composed of a feature extractor, BEP module, and three headers of edge, truncation, and concentric mask. These headers are responsible for the parameter prediction of edge approximating curve fitting function, the determination of truncation points, and the location of text instances.

## B. Overall Framework

As introduced in the above section, the edge approximation-based representation method helps fit irregular texts compactly in an intuitive way, which encourages avoiding the limitations in current popular practices. Following the designed text representation method, we construct a novel detection framework (EdgeText), and the structure is described in this section.

The framework comprises a feature extractor, BEP module, and three headers that of edge, truncation, and concentric mask (as shown in Fig. 5). Feature extractor includes traditional backbone [41], [42], [43] and feature pyramid network (FPN) [44], which is responsible to provide multi-scale visual features  $F_1, F_2, F_3$ , and  $F_4$  corresponding to the image size of  $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ , and  $\frac{1}{32}$  respectively for the following prediction headers. Considering the recognition of text edge features is important for rebuilding reliable curve boxes in the inference stage, we design a bilateral enhanced perception (BEP) module. It takes the multi-scale visual features as input and optimizes the  $F_1$  with a Gaussian truncated edge segment, which encourages EdgeText to recognize edge shape distribution more accurately. The BEP final outputs a BEP feature  $F_{bep} \in \mathbb{R}^{\frac{H}{4}, \frac{W}{4}, ch}$  concatenated via enhanced multi-scale visual features, where  $H$  and  $W$  are the height and width of input image respectively.  $ch$  denotes the number of the feature channels. After the feature enhancing process, the concentric mask [45] (a shrink-mask used for locating text instances without the influence of occlusion or adhesion problem) header then segments the text concentric mask to determine the corresponding text location. In the end, edge and truncation headers then predict the parameter  $\Theta$  of the curve fitting function and start and end truncation points ( $tp^s$  and  $tp^e$ ) (referred to Section III-A) based on the BEP feature and text location information brought by concentric masks.

Specifically, The three headers ( $h_{conc}$ ,  $h_{edge}$ , and  $h_{trun}$ ) consists of a stack of fully convolution layers, the structure can be expressed as follows:

$$\begin{aligned} h_{conc} &= C_3^{ch_c}(C_3^{\frac{ch}{2}}(C_3^{ch}(C_1^{ch}(F_{bep})))), \\ h_{edge} &= C_3^{ch_e}(C_3^{\frac{ch}{2}}(C_3^{ch}(C_1^{ch}(F_{bep})))), \\ h_{trun} &= C_3^{ch_t}(C_3^{\frac{ch}{2}}(C_3^{ch}(C_1^{ch}(F_{bep})))), \end{aligned} \quad (7)$$

where  $C_1^{ch}$  denotes the convolution layer with  $1 \times 1$  kernel and outputs the features with  $ch$  channels.  $ch_c$ ,  $ch_e$ , and  $ch_t$  are the numbers of the output channels of the three headers respectively.  $ch_c$  is set to 1 for representing the text category.  $ch_e$  is the number of curve fitting function parameters (e.g., it is set as 2, 3, or 4 when the polynomial's degrees are 1, 2 or 3).  $ch_t$  is set to 8 for encoding the coordinates  $(x, y)$  of four truncation points ( $tp_t^s, tp_t^e, tp_b^s$ , and  $tp_b^e$ ).

For each text, the values on the concentric mask center locations of  $h_{edge}$  are picked as predicted curve fitting function parameters. Different from the parameters, considering the large offsets between center points and truncation points, we choose the lateral vein prediction strategy proposed in LeafText [46] to determine the truncation point locations. The strategy extracts text center lines from concentric masks at first. Then, the start and end points of the center lines are determined. Finally, the start truncation points ( $tp_t^s$  and  $tp_b^s$ ) of the text's top and bottom edges are obtained by the values from  $h_{trun}$  on the center line start point locations. For the end truncation points ( $tp_t^e$  and  $tp_b^e$ ), they are determined by the values from  $h_{trun}$  on the center line endpoint locations. The above strategy allows us to predict shorter offsets between the start and end points of the center lines and truncation points instead of regressing the larger offsets between center points and truncation points directly, which helps to locate truncation points more accurately. With the parameters and truncation

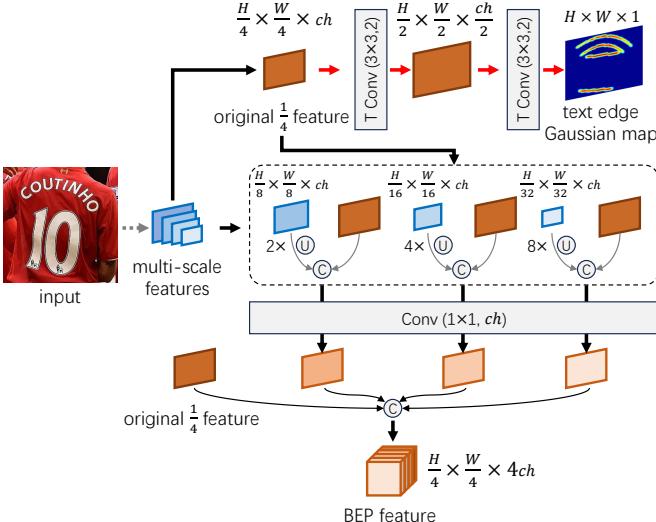


Fig. 6. Visualization of the proposed BEP module. It takes the multi-scale features from the feature extractor as input for generating a BEP feature to feed the following prediction headers.

points, the curve boxes can be reconstructed by the process introduced in the above section.

### C. Bilateral Enhanced Perception Module

To improve EdgeText's recognition ability of text edge features, we design a bilateral enhanced perception (BEP) module. It optimizes the network via Gaussian truncated edge segment supervision labels, encouraging EdgeText to recognize edge distribution more accurately. The module structure will be illustrated in this section.

As visualized in Fig. 6, the BEP module takes the multi-scale features from the feature extractor (referred to Fig. 5) as input for generating a bilateral enhanced perception (BEP) feature to feed the following prediction headers. Concretely, the module up-samples  $F_1$  to the size of  $H \times W$  via two transposed convolution layers to predict the text edge confidence map first. It enhances the fine-grained feature ( $F_1$ ) in multi-scale features to recognize the text edge shapes and the corresponding truncation point locations. Then,  $F_2$ ,  $F_3$ , and  $F_4$  are up-sampled to the size of  $F_1$  for generating three concatenate features. Next, the channels of these features are reduced to  $ch$  via a convolution layer with  $1 \times 1$  kernel. In the end, the above three concatenate features with  $ch$  channels are further concatenated with the enhanced  $F_1$  to generate the final BEP feature. The concatenate operation between fine-grained feature  $F_1$  and coarse-grained features  $F_2$ ,  $F_3$ , and  $F_4$  helps the BEP feature enjoys the comprehensive ability to recognize curve box related features.

### D. Loss Function

EdgeText represents irregular texts via the curve box that consists of the curve fitting function and truncation points. As described in Fig. 5, the edge header and truncation header are responsible for predicting the parameters and points respectively, and the BEP module helps our model to recognize the

features of edge shape and truncation points. To optimize the proposed framework effectively, we construct a comprehensive loss function  $\mathcal{L}$ , which comprises edge header loss  $\mathcal{L}_{edge}$ , truncation header loss  $\mathcal{L}_{trun}$ , and BEP module loss  $\mathcal{L}_{bep}$ . The comprehensive loss function  $\mathcal{L}$  can be expressed as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{edge} + \beta \mathcal{L}_{trun} + \gamma \mathcal{L}_{bep}, \quad (8)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are the importance coefficients of  $\mathcal{L}_{edge}$ ,  $\mathcal{L}_{trun}$ , and  $\mathcal{L}_{bep}$ , respectively. They are set to 0.5, 0.5, and 1.0 in the following experiments.

For  $\mathcal{L}_{edge}$ , considering the varied scales of scene texts, we propose a proportional integral loss  $\mathcal{L}_{pi}$  to optimize the edge header.  $\mathcal{L}_{pi}$  creates curves according to the predicted parameter  $\tilde{\Theta}$  and the corresponding ground-truth  $\Theta$  in the range from -0.5 to 0.5 on X-axis. Then, the integral of the absolute difference between two curves in this range is taken as the optimization objective. Here, we take the top edge as an example to formulate the computation process:

$$\mathcal{L}_{edge} = \mathcal{L}_{pi} = \int_{-0.5}^{0.5} |f(\Theta; x) - f(\tilde{\Theta}; x)| dx, \quad (9)$$

where  $\Theta$  is the parameter obtained from the curve fitting function of transformed polygon points instead of the original label points (as explained in Fig. 3). Therefore, EdgeText can learn the overall shape features of the text edge in the range of -0.5 to 0.5 on the X-axis and not be affected by text scales, which ensures our model balances the importance of different instances with varied scales and accelerates the optimization compared with computing loss according to original points directly. Considering the image is pixel-level matrix and the independent variable  $x$  is not continuous, we transform the Equation 9 to the discrete form:

$$\mathcal{L}_{pi} = \lim_{N \rightarrow \infty} \sum_{i=1}^N |f(\Theta; x_i) - f(\tilde{\Theta}; x_i)| \Delta x, \quad (10)$$

$$\Delta x = \frac{1}{N}, \quad x \in [-0.5, 0.5],$$

where  $N$  denotes the number of sample points in the range of -0.5 to 0.5 on the X-axis and when  $N$  approaching infinity, the above discrete form is equal to continuous form. To compute the loss in the training stage,  $N$  is set as a finitude number in this paper empirically and the Equation 10 can be further simplified as following expression:

$$\mathcal{L}_{pi} = \sum_{i=1}^N |f(\Theta; x_i) - f(\tilde{\Theta}; x_i)|, \quad (11)$$

$$x_i = -0.5 + \frac{i}{N}.$$

For truncation header loss  $\mathcal{L}_{trun}$  and BEP module loss  $\mathcal{L}_{bep}$ , they are responsible for supervising the distance regression and semantic segmentation tasks respectively. Considering smooth- $l_1$  loss [10] is more robust to outliers and less susceptible to extreme values than  $l_2$  loss, thus we adopt smooth- $l_1$  loss as  $\mathcal{L}_{trun}$  for optimization of the truncation header in the training stage for ensuring model's stability and generalization ability. Furthermore, considering the sample imbalance between the background and the text edge region, Dice loss [47] is introduced to evaluate the BEP module loss  $\mathcal{L}_{bep}$ , which can be expressed as the following equation:

$$\mathcal{L}_{bep} = 1 - \frac{2 \times |\mathbf{M}_{pred} \cap \mathbf{M}_{gt}| + \varepsilon}{|\mathbf{M}_{pred}| + |\mathbf{M}_{gt}| + \varepsilon}, \quad (12)$$

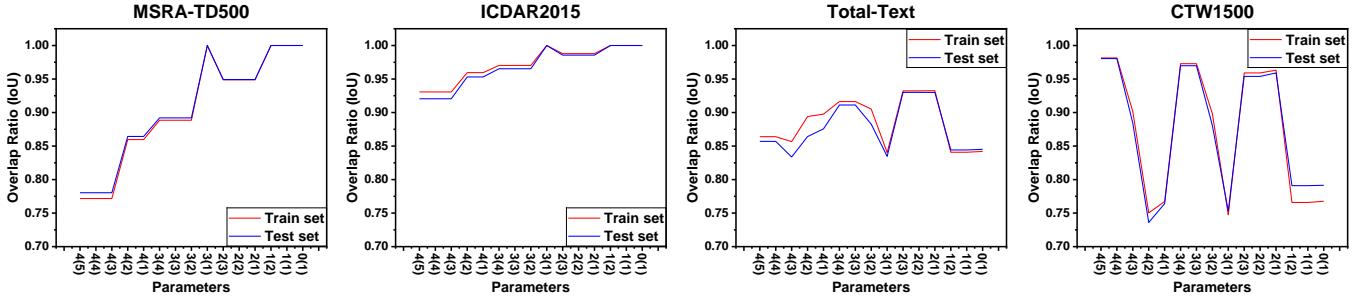


Fig. 7. Analysis of the overlap ratio between reconstructed curve boxes based on edge approximation representation and ground-truth contours via the metric of Intersection of Union (IoU) under different settings of fitting function parameters. The tick label on the Parameters-axis illustrates the specific parameter setting (e.g., 4(3) means the highest degree  $m$  of polynomial is set to 4). Only the parameters of  $\{\Theta_i, |i| = 2, 3, 4\}$  are adopted to fitting the text edge without the parameters of  $\{\Theta_i, c|i| = 1\}$ .

where  $M_{pred}$  and  $M_{gt}$  are the predicted text edge and the corresponding ground-truth.  $\varepsilon$  is used to avoid the situation where there may be no positive samples in  $M_{gt}$ , which is set to 1 in the following experiments empirically.

#### IV. EXPERIMENTS

In this section, we first introduce the representative text detection datasets and the corresponding evaluation metrics. Then, implementation details about experiments and model settings are illustrated. Next, we explore the proposed edge approximation-based text representation method's fitting ability. Furthermore, the effectiveness of the designed BEP module and PI-loss is demonstrated via ablation studies. Meanwhile, the model performance with different parameter settings of the curve fitting function is illustrated to explore suitable parameters. Ultimately, we show the superior performance of the constructed EdgeText by comparing it with existing state-of-the-art methods on multiple public datasets.

##### A. Datasets and Evaluation Metrics

**Datasets.** Scene texts can be divided into regular and irregular categories according to their shapes. In this paper, we introduce the datasets with multi-oriented rectangle shapes (MSRA-TD500 [48] and ICDAR2015 [49]) and curve shapes (Total-Text [50] and CTW1500 [51]) to evaluate the comprehensive detection performance of our EdgeText on arbitrary-shaped text instances.

**Evaluation Metrics.** To compare the proposed EdgeText with existing methods for showing superior performance, we follow current popular models to evaluate the performance through the three metrics (Precision, Recall, and H-mean) in object detection. These metrics can be computed as follows:

$$\begin{aligned} \text{Precision} &= \left( \frac{\text{TP}}{\text{TP} + \text{FP}} \right) \times 100\%, \\ \text{Recall} &= \left( \frac{\text{TP}}{\text{TP} + \text{FN}} \right) \times 100\%, \\ \text{H - mean} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned} \quad (13)$$

where TP, FP, and FN are the number of the predicted True Positive, False Positive, and False Negative samples.

##### B. Implementation Details

As introduced in Fig. 5 and Section III-B, the EdgeText framework extracts multi-scale features through the feature extractor that comprises a backbone and FPN. In the following experiments, we chose ResNet-50 [42] that pre-trained on ImageNet [52] as the backbone, and the FPN is the version implemented by PyTorch officially.

In the training stage, three kinds of the following data augmentation strategies are adopted to vary the training samples for improving EdgeText's generalization ability: 1) random cropping; 2) random rotating; 3) random scaling; 4) random color distortion. The model is optimized with 1200 epochs under the setting of 16 batch size and 1e-6 initialized learning rate. In the inference stage, the red color arrows in Fig. 6 are abandoned, which are responsible for enhancing the model's ability to recognize text edges and truncation point locations in the training stage only. During the training process, we used four 1080ti GPUs to train the model in parallel. For the testing process, a single 1080ti GPU is used to evaluate the model's performance.

##### C. Fitting Ability of Edge Approximation Representation

An edge approximation-based text representation is proposed to fit arbitrary-shaped scene texts compactly via a simple and intuitive process, which helps avoid the limitations exist in current popular methods. As introduced in Section III-A and Equation 2, polynomial is adopted as the curve fitting function for approximating text edges. Considering the varied shapes of scene texts, we explore the fitting ability of edge approximation representation on them and find out a suitable parameter settings to obtain a superior comprehensive performance for text instances with different shapes.

Concretely, we analyze the overlap ratio between reconstructed curve boxes based on edge approximation representation and ground-truth contours via the metric of Intersection of Union (IoU) under different settings of fitting function parameters. As shown in Fig. 7, the relationships between the overlap ratio and parameters on multiple public datasets are visualized, where the tick label on the Parameters-axis illustrates the specific parameter setting (e.g., 4(3) means the highest degree  $m$  of polynomial is set to 4). Only the parameters of  $\{\Theta_i, |i| = 2, 3, 4\}$  are adopted to fitting the text

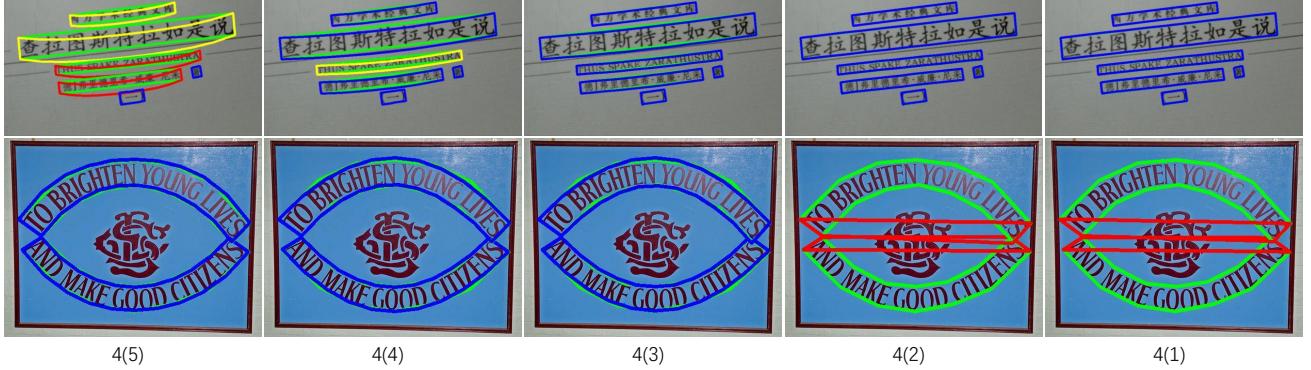


Fig. 8. Visualization of the fitting results generated by the proposed edge approximation representation method toward regular and irregular-shapes texts. Green contours are the ground-truth labels. Blue, yellow, and red contours are the generated high-quality, medium-quality, and low-quality curve boxes respectively, where different level qualities are defined according to the IoU between generated curve boxes and labels. Concretely, high-quality, medium-quality and low-quality curve boxes are distinguished by IoU ranges of  $\text{IoU} \leq 0.5$ ,  $0.5 \leq \text{IoU} \leq 0.7$ , and  $\text{IoU} \geq 0.7$ .

TABLE I  
EXPERIMENTAL RESULTS OF EDGE TEXT WITH DIFFERENT CURVE FITTING FUNCTION PARAMETER SETTINGS ON MSRA-TD500 AND TOTAL-TEXT DATASETS.

parameter	MSRA-TD500			Total-Text		
	Precision	Recall	H-mean	Precision	Recall	H-mean
2(1)	91.0	83.7	87.2	89.4	85.9	87.6
2(2)	93.3	87.1	90.1	88.8	85.1	86.9
2(3)	76.6	81.2	78.8	89.1	84.4	86.7

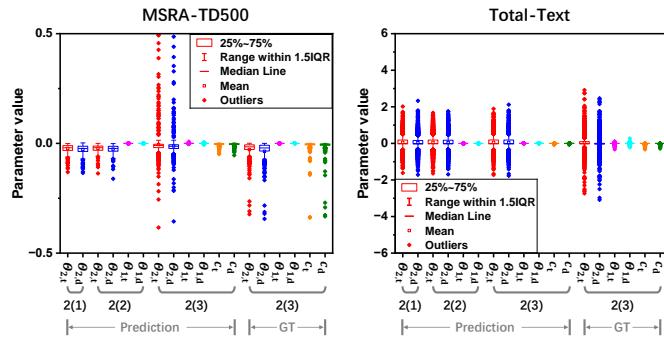


Fig. 9. Visualization of the predicted curve fitting function parameters with different settings and the corresponding ground-truth on MSRA-TD500 and Total-Text datasets.

edge without the parameters of  $\{\Theta_i, c|i = 1\}$ ). For rectangle and quadrilateral boxes labeled datasets (MSRA-TD500 [48] and ICDAR2015 [49]), it is found that the proposed edge approximation representation enjoys stronger fitting ability with the decrease of the polynomial highest degree (as shown in the Fig. 8 first row) and the fitting ability starts to fluctuate around the highest degree is set to 2, which mainly because the overfitting of the high degree polynomial toward edge with simple shapes (such as straight line). The same trend occurs when tuning the number of parameters smaller with a fixed degree (e.g., the ascending curve in the parameters range of 4(5) to 4(1) for MSRA-TD500 and ICDAR2015 datasets). Different from the above two datasets, as visualized in the Fig. 8 second row), the text in Total-Text [50] and CTW1500 [51] is composed of two curved edges that always

TABLE II  
EXPERIMENTAL RESULTS OF EDGE TEXT WITH DIFFERENT MODULE AND LOSS FUNCTION SETTINGS ON THE TOTAL-TEXT DATASET.

BEP module	PI-loss	Total-Text		
		Precision	Recall	H-mean
	✓	86.7	85.4	86.0
		88.2	84.9	86.5
	✓	89.4	85.9	87.6

TABLE III  
EXPERIMENTAL RESULTS OF EDGE TEXT WITH DIFFERENT TRUNCATION POINT SAMPLING STRATEGIES ON THE TOTAL-TEXT DATASET.

From center	From endpoint	Total-Text		
		Precision	Recall	H-mean
✓	✓	88.7	85.8	87.2
		89.4	85.9	87.6

can be approximated compactly by the polynomial with a higher degree ( $m \geq 2$ ,  $m$  denotes the highest degree of polynomial). It would lead to low-quality fitting results when  $m \leq 1$  because of the intrinsic limitation of straight lines to curves. Besides, there is an opposite trend compared with MSRA-TD500 and ICDAR2015 datasets in that the overlap ratio decreases when tuning the number of parameters smaller with a fixed degree. It is mainly because some text edge approximating curves are not Y-symmetric and the fitting function with fewer parameters fits these curves worse.

Overall, based on the above observation from the analysis in Fig. 7, we choose to set the polynomial highest degree  $m$  to 2 to ensure a strong comprehensive fitting ability of edge approximation representation toward arbitrary shapes of multiple public datasets.

#### D. Ablation Study

In the above Section IV-C, the strong fitting ability of the proposed edge approximation representation toward regular and irregular-shaped texts has been shown and suitable parameter settings ( $m = 2$ ) of the curve fitting function have been determined. We will further explore the performance

TABLE IV  
PERFORMANCE COMPARISON ON MULTIPLE PUBLIC DATASETS, INCLUDING MSRA-TD500, ICDAR2015, TOTAL-TEXT, AND CTW1500.

Methods	MSRA-TD500			ICDAR2015			Total-Text			CTW1500		
	Precision	Recall	H-mean									
EAST'17 [21]	87.3	67.4	76.1	83.3	78.3	80.7	49.0	43.1	45.9	46.7	37.2	41.4
TextSnake'18 [7]	83.2	73.9	78.3	84.9	80.4	82.6	61.5	67.9	64.6	65.4	63.4	64.4
CARFT'19 [38]	88.2	78.2	82.9	89.8	84.3	86.9	87.6	79.9	83.6	86.0	81.1	83.5
TextField' 19 [33]	87.4	75.9	81.3	84.3	83.9	84.1	81.2	79.9	80.6	83.0	79.8	81.4
ContourNet' 20 [26]				87.6	86.1	86.9	86.9	83.9	85.4	83.7	84.1	83.9
DBNet'20 [34]	91.5	79.2	84.9	88.2	82.7	85.4	87.1	82.5	84.7	86.9	80.2	83.4
CAFA'20 [36]				86.2	82.7	84.4	84.6	78.6	81.5	85.7	85.1	85.4
FCENet'21 [39]				90.1	82.6	86.2	89.3	82.5	85.8	87.6	83.4	85.5
PCR'21 [4]	90.8	83.5	87.0				88.5	82.0	85.2	87.2	82.3	84.7
ABCNet'21 [6]	89.4	81.3	85.2	90.4	86.0	88.1	90.2	84.1	87.0	85.6	83.8	84.7
CM-Net'22 [45]	89.9	80.6	85.0	86.7	81.3	83.9	88.5	81.4	84.8	86.0	82.2	84.1
NASK'22 [3]				90.9	89.2	90.0	85.6	83.2	84.4	83.4	80.1	81.7
RFN'22 [2]	88.4	87.8	88.1									
ASTD'22 [37]	88.6	82.7	85.6	88.2	83.3	85.7	89.8	84.8	87.2	87.6	80.8	84.1
DeepSolo'23 [5]				92.8	87.4	<b>90.0</b>	93.9	82.1	87.3			
LeafText'23 [46]	92.1	83.8	87.8	88.9	82.3	86.1	90.8	84.0	87.3	87.1	83.9	85.5
CBNet'24 [35]	91.1	84.8	87.8	89.0	95.5	87.2	90.1	82.5	86.1	89.0	81.9	85.3
VTD'24 [1]	89.2	81.5	85.2	88.5	85.8	87.1						
<b>Ours</b>	<b>93.3</b>	<b>87.1</b>	<b>90.1</b>	<b>89.8</b>	<b>83.6</b>	<b>86.6</b>	<b>89.4</b>	<b>85.9</b>	<b>87.6</b>	<b>86.9</b>	<b>84.3</b>	<b>85.6</b>



Fig. 10. Visualization of some quality detection results on regular (MSRA-TD500 and ICDAR2015) and irregular-shaped (Total-Text and CTW1500) text detection datasets. It can be observed that the proposed method can fit arbitrary-shaped scene texts simultaneously and effectively.

influences brought by different numbers of parameters under the setting of  $m = 2$  and verify the effectiveness of the introduced BEP module and PI-loss.

**Influence of the number of parameters.** As shown in Table I, EdgeText achieves comparable performance in H-mean (87.2% and 90.1%) when building the curve fitting function without constant term on MSRA-TD500. Meanwhile, the model performance degrades significantly when reconstructing with constant terms in the inference stage. It is observed that the predicted  $\Theta_{2,t}$  and  $\Theta_{2,d}$  of the function deviates far from the corresponding ground truth when the constant term participates in the training process (as shown in the parameter analysis in left sub-figure of Fig. 9). Since  $\Theta_{2,t}$  and  $\Theta_{2,d}$  are more important for rebuilding contours than other lower-degree parameters and constant terms, the parameter deviation results in performance degradation. On

the Total-Text dataset, the performance shows a slight decrease trend with the number of parameters tuned larger. Especially, there isn't a significant performance decrease trend when the constant term is considered in model optimization. It is mainly because the corresponding label distributes around 0, which brings limited influence to the model to learn coefficients of a higher polynomial degree.

**Effectiveness of BEP module and PI-loss.** Considering the deep dependency of EdgeText on text edges, the BEP module and PI-loss are introduced to encourage our model to recognize edge features more accurately. As shown in Table II, compared with the baseline, the BEP module and PI-loss bring 0.5% and 1.1% improvements in H-mean. Specifically, the above two components help our model recognize the text edge features effectively in the optimization process, which enhances the precision of the rebuilt text contours in the inference stage

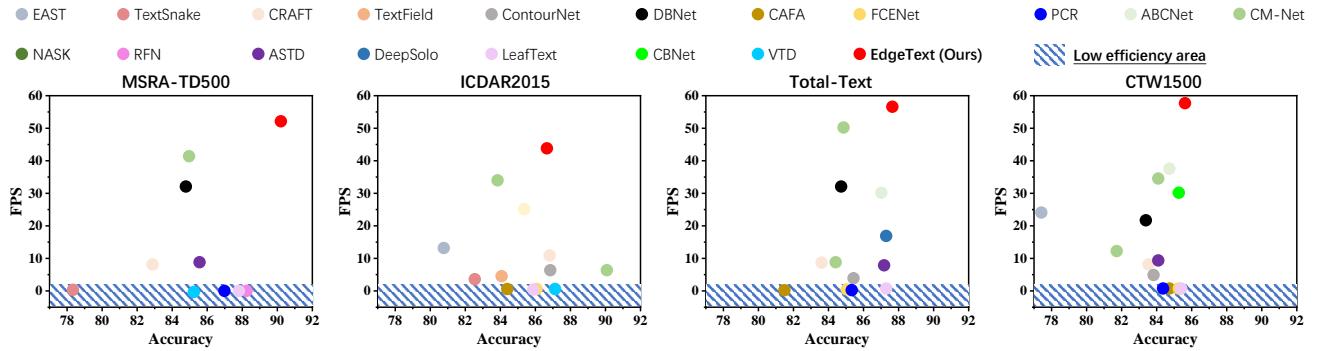


Fig. 11. Visualization of the comparison between the efficiency and accuracy balance with existing advanced methods.



Fig. 12. Qualitative comparisons with FCENet [39] and CM-Net [45] on some challenging samples. FCENet and CM-Net are the representative irregular-shaped text detection methods that adopt the piecewise fitting and shrink-mask expand strategies, respectively.

(1.5% and 1.2% improvements in Precision metric). The above experimental results demonstrate the effectiveness of the BEP module and PI-loss in improving EdgeText's ability to recognize text edges more accurately.

**Influence of different truncation point sampling strategies.** As described in Section III-B, considering the large offsets between center points and truncation points, the lateral vein prediction strategy proposed in LeafText [46] is adopted to determine the truncation points. It allows the proposed EdgeText to determine the truncation point locations by predicting shorter offsets between truncation points and center line endpoints of concentric masks instead of the large offsets between truncation points and centers, which helps predict truncation points more accurately. As shown in Table III, the strategy proposed in LeafText brings 0.4% performance improvements in H-mean compared with determining the truncation points by combining the text centers and the offsets between truncation points and centers.

### E. Comparison Experiments

According to the experimental analysis in Section IV-C and Section IV-D, the effectiveness of the proposed BEP module and PI-loss are verified and suitable model settings are found for obtaining a superior comprehensive performance on varied texts. Based on the above results, we compare EdgeText with existing approaches for showing the superiority of our model.

As shown in Table IV and Fig. 10, the detection results of our model on various public datasets are illustrated. For large-scale text instances in MSRA-TD500 dataset, EdgeText achieves 90.1% H-mean, which outperforms the existing state-of-the-art (SOTA) method (LeafText [46]) 2.3%. As described in Section III, we adopt the lateral vein prediction strategy proposed in LeafText to encourage determining the truncation points more accurately. Furthermore, compared with the piecewise fitting process of LeafText, our method represents the text contour as a whole, which avoids the cumulative error that may exist in a gradual rebuilding process and ensures the compactness of reconstructed contours. Benefiting from the above advantages, EdgeText outperforms existing methods a lot. For ICDAR2015, a small-scale regular text dataset compared with MSRA-TD500, our model behind of DeepSolo [5] slightly. It is mainly because the concentric mask-based text location strategy proposed in CM-Net [45] makes it hard to determine small instances because of the lesser pixel-level positive samples. Besides of the above regular-shaped text datasets, EdgeText enjoys superior performance on irregular datasets (Total-Text and CTW1500). Concretely, it surpasses LeafText [46] 0.3% in H-mean on Total-Text dataset and ASTD [37] 1.5% in H-mean on CTW1500. The above superiority of our model brought by the edge approximation representation method that fits texts compactly and continuously in an intuitive way, which helps EdgeText detect texts more accurately. Furthermore, in Fig. 12, we also compare the compactness of the detection results from EdgeText and existing representative methods that adopt piecewise fitting and shrink-mask expand strategies, respectively. Compared with previous methods, the compact contours generated by our method effectively avoid bringing unnecessary visual information into the subsequent text recognition process. This helps to reduce background interference and significantly improves overall recognition accuracy.

### F. Efficiency Analysis

As described in Section III-A, our proposed edge approximation representation method addresses the arbitrary-shaped text detection problem as a curve box regression process. This approach enables the parallel rebuilding of text instances with varying scales and shapes, thereby simplifying post-processing procedures compared to existing advanced methods and re-

ducing inference time. As illustrated in Fig. 11, our method outperforms previous approaches in terms of speed on multiple representative public datasets while maintaining competitive accuracy. The efficiency of the detection process is largely attributed to the streamlined post-processing steps. Additionally, the designed Bilateral Enhanced Perception (BEP) module facilitates the extraction of strong, representative features with a lightweight network, ensuring high detection accuracy at a lower resource cost. These results underscore the superior efficiency of our model.

## V. CONCLUSION

In this paper, we design an edge approximation representation based on the observation of scene text shapes and construct a novel text detection framework, namely EdgeText. The proposed representation helps fit arbitrary-shaped texts compactly and continuously, which avoids the intrinsic limitations that exist in current popular methods. Meanwhile, the edge approximation representation allows our model to rebuild multiple text contours in parallel, which helps EdgeText enjoy a more intuitive contour rebuilding process with fewer procedures compared with existing box-to-polygon strategy-based or piecewise fitting methods. Furthermore, considering the deep dependencies of EdgeText on the text edges and truncation points, the bilateral enhanced perception (BEP) module is designed to improve the model's ability to recognize edge features. In the end, we introduce a proportional integral loss (PI-loss) for accelerating the parameters prediction of the edge approximating curve functions. Ablation studies demonstrate the effectiveness of the introduced BEP module and PI-loss. Comparison experiments on the multiple datasets show the superior performance of our EdgeText.

## REFERENCES

- [1] J.-B. Zhang, W. Feng, M.-B. Zhao, F. Yin, X.-Y. Zhang, and C.-L. Liu, “Video text detection with robust feature representation,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [2] 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.
- [3] 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, 2021.
- [4] P. Dai, S. Zhang, H. Zhang, and X. Cao, “Progressive contour regression for arbitrary-shape scene text detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7393–7402.
- [5] M. Ye, J. Zhang, S. Zhao, J. Liu, T. Liu, B. Du, and D. Tao, “Deep solo: Let transformer decoder with explicit points solo for text spotting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 348–19 357.
- [6] Y. Liu, C. Shen, L. Jin, T. He, P. Chen, C. Liu, and H. Chen, “Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8048–8064, 2021.
- [7] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, “Textsnake: A flexible representation for detecting text of arbitrary shapes,” in *ECCV*, 2018, pp. 20–36.
- [8] Y. Liu and L. Jin, “Deep matching prior network: Toward tighter multi-oriented text detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1962–1969.
- [9] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, “Arbitrary-oriented scene text detection via rotation proposals,” *IEEE transactions on multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *NeurIPS*, 2015, pp. 91–99.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, “Ssd: Single shot multibox detector,” in *ECCV*, 2016, pp. 21–37.
- [12] P. Wang, H. Li, and C. Shen, “Towards end-to-end text spotting in natural scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7266–7281, 2021.
- [13] 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.
- [14] S. Tian, S. Lu, and C. Li, “Wetext: Scene text detection under weak supervision,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1492–1500.
- [15] B. Shi, X. Bai, and S. Belongie, “Detecting oriented text in natural images by linking segments,” in *CVPR*, 2017, pp. 2550–2558.
- [16] 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*, vol. 31, no. 1, 2017.
- [17] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, “Rotation-sensitive regression for oriented scene text detection,” in *CVPR*, 2018, pp. 5909–5918.
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *CVPR*, 2016, pp. 779–788.
- [19] L. Huang, Y. Yang, Y. Deng, and Y. Yu, “Densebox: Unifying landmark localization with end to end object detection,” *arXiv preprint arXiv:1509.04874*, 2015.
- [20] Z. Tian, C. Shen, H. Chen, and T. He, “Fcose: Fully convolutional one-stage object detection,” in *ICCV*, 2019, pp. 9627–9636.
- [21] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “East: an efficient and accurate scene text detector,” in *CVPR*, 2017, pp. 5551–5560.
- [22] S. Zhang, X. Zhu, J. Hou, C. Liu, C. Yang, H. Wang, and X. Yin, “Deep relational reasoning graph network for arbitrary shape text detection,” in *CVPR*, 2020, pp. 9699–9708.
- [23] L. Deng, M. Huang, X. Xie, Y. Liu, L. Jin, and X. Bai, “Progressive evolution from single-point to polygon for scene text,” *arXiv preprint arXiv:2312.13778*, 2023.
- [24] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, “Arbitrary shape scene text detection with adaptive text region representation,” in *CVPR*, 2019, pp. 6449–6458.
- [25] 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 *CVPR*, 2019, pp. 10 552–10 561.
- [26] 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 *CVPR*, 2020, pp. 11 753–11 762.
- [27] 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 *AAAI*, vol. 34, no. 07, 2020, pp. 12 160–12 167.
- [28] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, “Mask textspotter v3: Segmentation proposal network for robust scene text spotting,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 706–722.
- [29] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, “Learning shape-aware embedding for scene text detection,” in *CVPR*, 2019, pp. 4234–4243.
- [30] X. Xu, Z. Zhang, Z. Wang, B. Price, Z. Wang, and H. Shi, “Rethinking text segmentation: A novel dataset and a text-specific refinement approach,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 045–12 055.
- [31] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015, pp. 234–241.
- [33] 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.
- [34] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, “Real-time scene text detection with differentiable binarization,” in *AAAI*, 2020, pp. 11 474–11 481.
- [35] X. Zhao, W. Feng, Z. Zhang, J. Lv, X. Zhu, Z. Lin, J. Hu, and J. Shao, “Cbnnet: A plug-and-play network for segmentation-based scene text detection,” *International Journal of Computer Vision*, pp. 1–20, 2024.

- [36] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Transactions on Multimedia*, vol. 22, no. 8, pp. 1969–1984, 2019.
- [37] 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.
- [38] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *CVPR*, 2019, pp. 9365–9374.
- [39] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *CVPR*, 2021, pp. 3123–3131.
- [40] 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 *CVPR*, 2020, pp. 9809–9818.
- [41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [42] 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.
- [43] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
- [44] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.
- [45] C. Yang, M. Chen, Z. Xiong, Y. Yuan, and Q. Wang, "Cm-net: Concentric mask based arbitrary-shaped text detection," *IEEE Trans. Image Process.*, vol. 31, pp. 2864–2877, 2022.
- [46] C. Yang, M. Chen, Y. Yuan, and Q. Wang, "Text growing on leaf," *IEEE Transactions on Multimedia*, 2023.
- [47] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV*, pp. 565–571.
- [48] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *CVPR*, 2012, pp. 1083–1090.
- [49] 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 *ICDAR*, 2015, pp. 1156–1160.
- [50] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *ICDAR*, vol. 1, 2017, pp. 935–942.
- [51] Y. Liu, L. Jin, S. Zhang, and S. Zhang, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.
- [52] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.



**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.



**Tao Han** received a B.E. degree in transportation equipment and control engineering and an M.S. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2019 and 2022. He is currently pursuing a Ph.D. degree in computer science and engineering at the Hong Kong University of Science and Technology. His research interests include computer vision, ai4science, and AIGC.



**Han Han** received the B.E. degree in computer science from Northeast Forestry University, Harbin, China in 2023. He is currently working toward the master degree at the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and deep learning.



**Bingxuan Zhao** received the B.Sc. degree in Information and Computing Science from Northwestern Polytechnical University, Xi'an, China, in 2024. 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.



**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, embodied AI, and intelligent transportation.



the link (<https://crabwq.github.io/>)

**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