Text Growing on Leaf

Chuang Yang, Mulin Chen, Yuan Yuan, *Senior Member, IEEE,* and Qi Wang, *Senior Member, IEEE*

***Abstract*—Irregular-shaped texts bring challenges to Scene Text Detection (STD). Although existing contour point sequence-based approaches achieve comparable performances, they fail to cover some highly curved ribbon-like text lines. It leads to limited text fitting ability and STD technique application. Considering the above problem, we combine text geometric characteristics and bionics to design a natural leaf vein-based text representation method (LVT). Concretely, it is found that leaf vein is a generally directed graph, which can easily cover various geometries. Inspired by it, we treat text contour as leaf margin and represent it through main, lateral, and thin veins. We further construct a detection framework based on LVT, namely LeafText. In the text reconstruction stage, LeafText simulates the leaf growth process to rebuild text contour. It grows main vein in Cartesian coordinates to locate text roughly at first. Then, lateral and thin veins are generated along the main vein growth direction in polar coordinates. They are responsible for generating coarse contour and refining it, respectively. Considering the deep dependency of lateral and thin veins on main vein, the Multi-Oriented Smoother (MOS) is proposed to enhance the robustness of main vein to ensure a reliable detection result. Additionally, we propose a global incentive loss to accelerate the predictions of lateral and thin veins. Ablation experiments demonstrate LVT is able to depict arbitrary-shaped texts precisely and verify the effectiveness of MOS and global incentive loss. Comparisons show that LeafText is superior to existing state-of-the-art (SOTA) methods on MSRA-TD500, CTW1500, Total-Text, and ICDAR2015 datasets.**

***Index Terms*—Scene text detection, irregular-shaped text, leaf vein, text representation method**

I. Introduction

READING scene text helps intelligent devices are able to accomplish many applications (such as unmanned systems, intelligent transport, express system, and so on), which has dramatically improved production efficiency and people's quality of life. Scene Text Detection (STD) [1] is the key technique for intelligent devices to simulate humans reading scene text, which has attracted a growing number of researchers and becomes a hot topic in computer vision. In the past decade, deep learning has greatly promoted the development of many computer technologies. It helps to extract strong expressive image features for many tasks (e.g. recognition, tracking, and regression). Benefiting from the advantages of deep learning, the performance of STD technique achieves excellent improvements in the aspect of the regular-shaped text detection [2], [3]. However, there are many irregular-shaped texts in real scenarios, which brings challenges to traditional approaches. To fit arbitrary-shaped text instances effectively, an increasing number of novel methods are proposed, which can be categorized into segmentation-based methods [4], [5], [6] and regression-based methods [7], [8], [9] roughly.

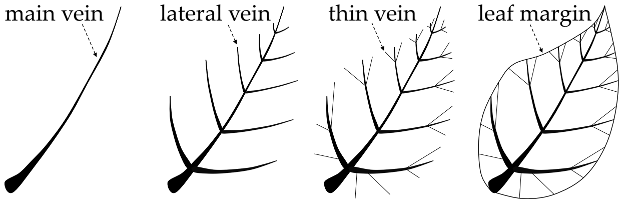


Fig. 1. Illustration of the proposed leaf vein-based text representation method. We aim to treat text contour as leaf margin and construct it through main, lateral, and thin veins. Main vein is used for locating text instance roughly. Lateral vein is responsible for determining coarse contour. Accurate text contour is fined through thin vein.

The former adopts mask representation, which segments text regions directly and can detect irregular-shaped text instances naturally. However, these methods frequently require large training data and less supervision information aggravates this phenomenon. The latter represents text instances by contour point sequences. They try to sample point sequences by regressing the offsets between center point or quadrilateral and irregular-shaped contour, which have clear drawbacks. Specifically, the one-stage regression-based methods fail to fit highly curved ribbon-like text lines because multiple contour points may reside in the same direction. For multiple-stages methods, the intrinsically computationally expensive post-processing leads to low detection efficiency and limited practical applications. Therefore, how to design an efficient and effective text representation method is under explored.

Considering the limitations above, we combine text geometric characteristics and bionics to design a natural text representation method, which can fit text instances with any shapes accurately, even for highly curved ones. As shown in Figure 1, it is found that leaf margins always enjoy irregular shapes, which is similar to scene texts. Importantly, the leaf margin can be covered precisely by a directed graph that is composed of main, lateral, and thin veins. Inspired by the leaf vein structure, we propose to represent text contour by the combination of main, lateral, and thin veins. We further construct a one-stage text detection framework (called LeafText) based on leaf vein. It rebuilds text contours by simulating the leaf growth process, which is an elegant and effective design. Concretely, for one text instance, LeafText first grows the corresponding main vein from the predicted kernel mask in Cartesian coordinates to locate the text roughly. Then, lateral and thin veins sprout along both sides of the main vein growth direction in polar coordinates. In the end, the text contour is drawn by connecting endpoints of lateral and thin

Chuang Yang is with the School of Computer Science, and with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China.

Mulin Chen, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence, OPtics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China.

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

Qi Wang is the corresponding author.

veins in a clockwise direction. Particularly, the lateral veins are used for determining coarse contour, and the thin veins are responsible for refining the contour to obtain an accurate detection result. Considering the deep dependencies of lateral and thin vein endpoints on the main vein, it is important to ensure a reliable main vein for rebuilding contour. However, the main vein extracted from the predicted kernel mask by the existing middle sampling method is always unreliable, which leads to a bad contour point sequence. Therefore, we propose a Multi-Oriented Smoother (MOS) to ensure the main vein robustness even when encountering unstable kernel masks. Additionally, text instances enjoy a large aspect ratio range compared with common objects, which brings challenges to predicting lateral and thin veins of text instances. Therefore, global incentive loss is proposed to force our model to balance the importance of text instances with different scales and focus on the prediction of lateral and thin veins. The main contributions of this paper are as follows:

1) By combining the text geometric characteristics and bionics, a leaf vein-based text representation method (LVT) is proposed. It explores a natural and effective way to fit arbitrary-shaped text instances, which enhances the model's fitting ability.
2) Thin vein is designed for fining text contours. It supports fitting texts accurately with lower model complexity, which accelerates the convergence of the training process effectively. Remarkably, the thin vein length is half of the lateral vein, which eases the learning of contour point sequence and ensures accurate detection results.
3) A Multi-Oriented Smoother (MOS) is designed to ensure the robustness of the main vein extracted from the predicted kernel mask. It provides the correct growth directions to the lateral and thin veins, which ensures a reliable contour point sequence.
4) Global incentive loss $\mathcal{L}_g$ is proposed to help balance the importance of text instances with different scales and force our method to focus on the predictions of lateral and thin veins. Particularly, it can be integrated into other regression-based detectors seamlessly.

The rest of the paper is organized as follows. Section II introduces the related works on text detection. Section III describes the architecture, training process, and inference process details of LeafText. The experimental results are discussed in Section IV. Section V concludes the paper.

II. Related Work

Recently, deep learning has promoted the development of the text detection technique greatly. According to the text representation method, previous text detectors can be classified into segmentation-based methods and regression-based methods roughly. In this section, a review of the existing text detection methods will be introduced.

A. Segmentation-Based Methods

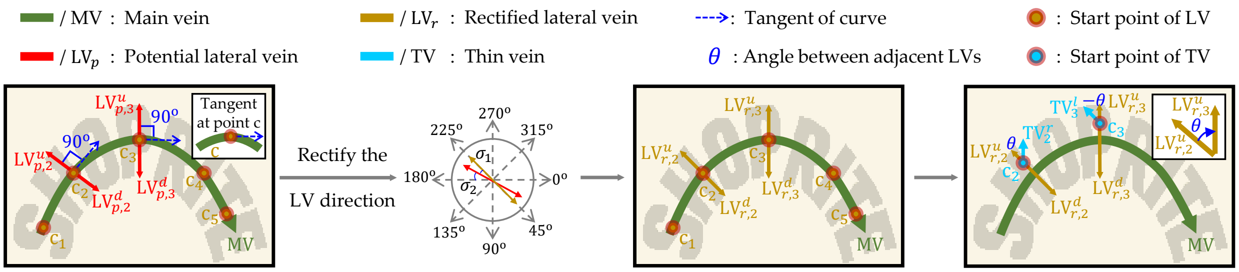
Segmentation technology [10] executes pixel-level classification on images, which provides an effective solution for text detection. Zhang *et al.* [11] segmented rough text regions at first. Then, they extracted character components within text blocks by MSER [12]. In the end, the authors suppressed false hypotheses by the intensity and geometric criteria of character components to obtain the final detection results. Lyu *et al.* [13] proposed to detect long text lines via a corner localization detection strategy. They generated candidate boxes by sampling and grouping corner points and filtered false-positive samples by the score of segmentation maps.

Deng *et al.* [14] found that it would lead to text adhesion problems if extracting text contours from segmentation maps directly. To alleviate the above problem, link heat maps in eight directions were predicted for separating adhesive text instances. The works [15], [16] designed similar strategies as [14] to provide solutions for the phenomenon that many texts are very close to each other. Different from the above works, Wang *et al.* [17], [18], [19] and Liao *et al.* [20], [21] proposed expansion strategies to generate text regions from shrink regions, which avoided detecting multiple adhesive texts as one either. The differences between them were that the former expanded shrink to text regions at pixel-level and the latter executed the expansion process at instance-level. The works [22], [23], [24] considered that a small amount of pixel-level annotated data limits the model performance and proposed a two-stage detection framework to make full use of a large amount of data annotated with rectangles. In the inference process, the authors located texts roughly by quadrilaterals and extracted text contours precisely from the corresponding segmented text regions within quadrilaterals. Zhang *et al.* [25] considered stack-omnidirectional text dilemma brings much challenges for text detection. They designed LSTM-based module to help generates omnidirectional text mask proposals from vertical and horizontal directions simultaneously to solve the stack-omnidirectional text dilemma.

Except for predicting the whole text instances directly, some approaches [26], [27], [28] detect texts in character-level. Baek *et al.* [26] proposed a weakly-supervised framework to generate character-level labels to promote the training process. In the rebuilding process, the approach first predicted character regions and then linked them by affinities to obtain the final detection results. Zhang *et al.* [27] adopted similar strategy as [26] to represent text instances. Moreover, they introduced Graph Convolutional Network (GCN) to predict the affinities between different character regions to improve the reliability of linked components. Different from them, Long *et al.* [28] segments center line firstly and then predict the each part bound along with the center line.

B. Regression-Based Methods

Object detection methods [29], [30], [31] adopt contour point sequence-based representation method to rebuild object contour or box, which brings great inspiration for the research of text detection. Liao *et al.* [32] inherited the framework of [30] directly to detect horizontal text. To improve the performance of the multi-oriented texts detection, they proposed to predict rotation angles of texts in [33]. Different from the above anchor-based detection framework, Zhou *et al.* [34] introduced the detection strategy proposed in [35] into text



(a) Growth of potential LV at n start point (n=5)

(b) Polar coordinate system with m directions (m=8)

(c) Rectified LV at n start point (n=5)

(d) Growth of thin vein

Fig. 2. Illustration of the vein growth process. It contains the following three stages: 1) growing potential lateral vein, which is responsible for determining potential growth directions according to the start points and the corresponding tangent slopes; 2) rectifying potential growth lateral vein directions; 3) growing thin vein based on the determined lateral vein.

detection, which predicted corner points of multi-oriented texts and connected them to obtain the text boxes. Liao *et al.* [36] focused on how to extract strong expressive features for multi-oriented texts. They proposed to rotate the convolutional filters to encourage the model to extract rotation-sensitive features. He *et al.* [37] extracted the text features with strong representation capacities through a hierarchical inception module.

Though the above works achieve comparable performance in detecting multi-oriented text instances, they are hard to detect curved texts effectively. To improve the model's ability to detect arbitrary-shaped text instances, Some researchers [38], [39], [40] separated word-level text blocks into multiple character-level regions. They regressed character boxes and linked those components to rebuild text blocks. The same as [38], Ma *et al.* [41] and Zhang *et al.* [42] adopted character-based detection strategy. Importantly, the authors utilized GCN to evaluate the linkages of adjacent characters to improve the stability of rebuilt text regions. Zhang *et al.* [43] and Wang *et al.* [44] designed two-stage contour point sequence representation method. They extracted text quadrangles and further predicted contour points based on features within the quadrangles through regression way. The former generated the text center line (TCL) region at first. Then, they regressed the offset between TCL and text contour to sample the contour point. The latter predicted the distance between quadrangle and text contour directly to extract the contour point. Wang *et al.* [45] proposed a more intuitive way to obtain contour points. The authors segmented those points directly in both vertical and horizontal directions and combined them to filter unreliable results. Inspired by [46], Wang *et al.* [47] modeled text instances into the polar coordinate system and emitted multiple rays from text center to contour. The ray endpoints were sampled as contour points and connected to obtained final detection results.

Some works [48], [49], [50] proposed novel regression strategies to detect text instances and achieved state-of-the-art performance. Specifically, Liu *et al.* [48] introduced Bezier-curve to represent text contours. They explored the probability to fit texts except for standard bounding box detection. Su *et al.* [49] encode the text regions into compact vectors through discrete cosine transform. Zhu *et al.* [50] modeled texts into Fourier domain and regressed contour point sequence by Fourier signature vectors.

III. Methodology

In this section, the leaf vein-based text representation method (LVT) is presented firstly. Then, we introduce the overall architecture of the proposed LeafText. Next, the details of Multi-Oriented Smoother (MOS) and Growth Process of Vein (GPV) are described. In the end, the optimization functions of network are given.

A. Leaf Vein-Based Text Representation Method

The proposed text representation method (LVT) treats the text contour as leaf margin (as shown in Fig 1) and represents it through main, lateral, and thin veins, which can fit text instances with any shapes effectively even for highly curved ones. This section describes the growth process of veins mathematically combined with Fig 2.

For **main vein** (as we can see from Fig 2), it is used for locating texts roughly. The corresponding growth process is modeled as polynomial $f(x)$ by MOS (described in Section III-C), which can be formulated as:

$$f\left(x\right)=\sum_{k=0}^{K}\omega_k x^k, \left(K\geqslant 1, x>0\right), \tag{1}$$

where K is the degree of $f(x)$. ω_k is the coefficient of x^k.

Given a main vein $f(x)$, n start points of lateral veins (n is set to 5 for better visualization) are sampled equidistantly along the growth direction of main vein (sampling process can be referred to Algorithm III-B). At each start point (x_{lv}, y_{lv}) on $f(x)$, the corresponding tangent angle φ_{lv} (Fig. 2 blue dotted arrow) can be computed by:

$$\varphi_{lv}=\arctan\left(\frac{df\left(x\right)}{dx}|_{x=x_{lv}}\right). \tag{2}$$

After obtaining φ_{lv}, we can determine the growth directions and lengths of lateral veins, which are responsible for generating coarse text contours.

For **the growth directions of lateral veins**, there are two lateral veins (LV^u and LV^d) along the growth direction of main vein at each start point (as we can see from Fig. 2 (a)). The corresponding potential growth directions (α_p^u and α_p^d) of

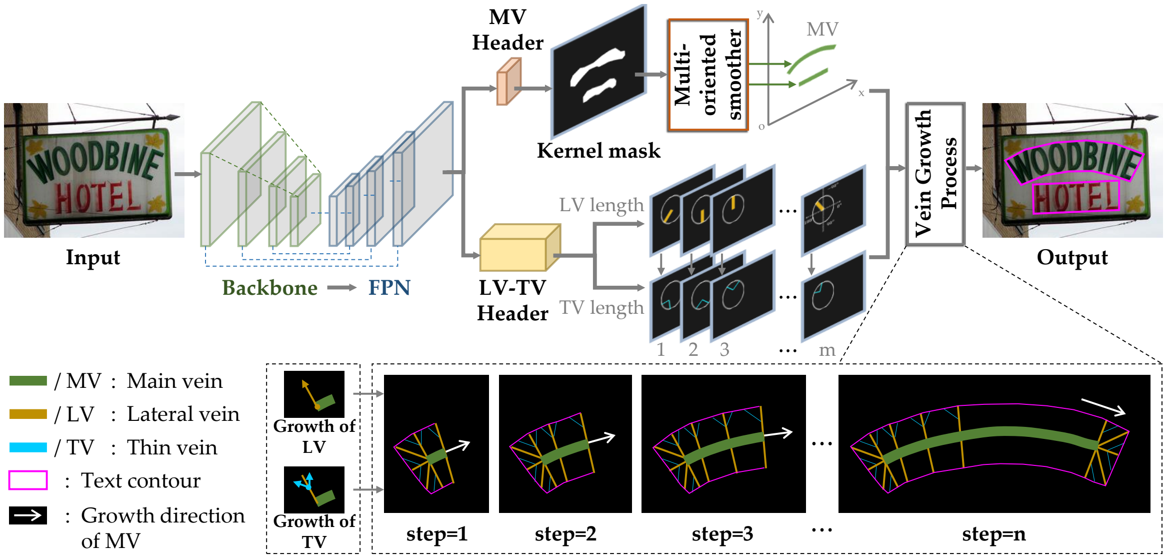


Fig. 3. Overall architecture of the proposed LeafText. It is composed of Backbone, FPN, MV Header, LV-TV header, Multi-Oriented Smoother (MOS), and Vein Growth Process. MV, LV, and TV denote main vein, lateral vein, and thin vein respectively. MOS extracts main vein from kernel mask in Cartesian coordinate system. Vein Growth Process is the text contours reconstructing process in Fig. 2.

them are defined as:

$$\alpha_p^u = \begin{cases} \varphi_{lv} - 90^o, & \text{if } \varphi_{lv} > 90^o \\ \varphi_{lv} - 90^o + 360^o, & \text{else} \end{cases}, \tag{3}$$

$$\alpha_p^d = \begin{cases} \varphi_{lv} + 90^o, & \text{if } \varphi_{lv} \leqslant 270^o \\ \varphi_{lv} + 90^o - 360^o, & \text{else} \end{cases}. \tag{4}$$

Since the coverage of all 360 directions would lead to expensive computational costs, we predefined a polar coordinate system with m directions ($m \ll 360$ and m is set to 8 for better visualization) to rectify the potential direction of lateral vein, which avoids a highly complicated neural network and ensures the strong fitting ability to text contours (verified in Section IV-C). Concretely, as shown in Fig. 2 (b), supposing $\alpha_1, \alpha_2, ..., \alpha_M$ are the all directions in the predefined polar coordinate system and $\alpha_m \leqslant \alpha_p^u \leqslant \alpha_{m+1}$ $(1 \leqslant m \leqslant M, m+1 = 1 | m = M)$, the rectified direction α_{rec}^u in Fig. 2 (c) can be calculated as:

$$\begin{aligned} \sigma_1 &= |\alpha_{m+1} - \alpha_p^u|, \\ \sigma_2 &= |\alpha_p^u - \alpha_m|, \\ \alpha_{rec}^u &= \begin{cases} \alpha_{m+1}, & \text{if } \sigma_1 < \sigma_2 \\ \alpha_m, & \text{else} \end{cases}, \end{aligned} \tag{5}$$

where σ_1 and σ_2 are the angles between the potential direction and the corresponding two adjacent directions in the predefined polar coordinate system. $|\cdot|$ denotes the operator for absolute value.

For **the lengths of lateral veins**, it can be constructed as the distances between the start points of lateral veins and text contours along the growth directions of lateral veins (refered to III-D).

With the determined lateral veins, the growth directions and lengths of thin veins can be formulated. They are used for refining the coarse contours generated by lateral veins to reconstruct accurate detection results. As we can see from Fig. 2 (d), the middle points of lateral veins are sampled as the start points of thin veins, and there are two thin veins (TV^l and TV^r) along the growth direction of lateral vein at one start point. For **the growth directions of thin veins**, they are determined according to the rectified directions of lateral veins. Specifically, given two adjacent lateral veins (TV_2^u and TV_3^u) and the corresponding rectified growth directions ($\alpha_{rec,2}^u$ and $\alpha_{rec,3}^u$) that computed by Equation 5, we can obtain the growth directions (α_2^r and α_3^l) of TV_2^r and TV_3^l by the Equation 6 and Equation 7, respectively:

$$\alpha_2^r = \alpha_{rec,3}^u, \tag{6}$$

$$\alpha_3^l = \alpha_{rec,2}^u. \tag{7}$$

For **the lengths of thin veins**, it is evaluated by the distances between the start points of thin veins and text contours along the growth directions of thin veins (refered to III-D). With the determined main veins, lateral veins, and thin veins, the text contour can be drawn by connected the endpoints of lateral veins and thin veins in a clockwise direction.

B. Overall Pipeline

The overall pipeline of LeafText is shown in Figure 3, which consists of backbone, FPN, MV header, LV-TV header, Multi-Oriented Smoother(MOS), and Vein Growth Process. ResNet [51] is adopted as the **backbone** to help extract basic input image $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$ features, where h, w are the height, width of input image with 3 channel. It outputs multiple coarse and fine feature maps $\mathbf{F}_{\frac{1}{s_1}}, \mathbf{F}_{\frac{1}{s_2}}, \mathbf{F}_{\frac{1}{s_3}}, \mathbf{F}_{\frac{1}{s_4}}$ simultaneously ($\mathbf{F}_{\frac{1}{s}} \in \mathbb{R}^{(h/s) \times (w/s) \times c}, s_1 = 4, s_2 = 8, s_3 = 16, s_4 = 32$), where s denotes the stride of network and c means feature maps channel. The coarse features bring a global correlation between texts and the fine ones focus on local details. To extract strong expressive features that are equipped with global and local information for the following detection headers, FPN [52] is used for combining multiple features from the backbone to generate a concatenated feature map $\mathbf{F}_{concat} \in \mathbb{R}^{(h/4)\times(w/4)\times(c\times4)}$. As described in Section III-A, text contour is represented by the combination of main vein, lateral vein, and thin vein. To extract main vein of text instance, LeafText inputs the $\mathbf{F}_{concat}$ into MV header the generation of kernel mask map $\mathbf{F}_k \in \mathbb{R}^{(h/4)\times(w/4)\times1}$ at first. Then, it extracts main vein from kernel mask $\mathbf{F}_k$ by MOS. For the growth of lateral and thin veins, LV-TV header conducts regression task on the $\mathbf{F}_{concat}$ to generate length mask map $\mathbf{F}_l \in \mathbb{R}^{(h/4)\times(w/4)\times m}$, where m is the number of directions in predefined polar coordinate system (as described in Section III-A). In $\mathbf{F}_l$, pixel values at the start points of lateral and thin veins are the vein lengths in m directions, respectively. With the determined main vein $f(x)$ (referred to Equation 1) and the lengths of lateral and thin veins in all m directions, text contour can be generated by the Vein Growth Process (described to Section III-A and Fig. 2).

C. Multi-Oriented Smoother

As shown in Fig. 2 and Fig. 3, extracting main vein $f(x)$ (refer to Equation 1) from the predicted kernel mask accurately is important for determining the lateral and thin veins, which is the key to rebuild text contour. However, generating main vein by the existing middle sampling method always results in a discrete jagged result, which leads to bad growth directions for lateral and thin veins and unreliable reconstructed text contours.

Considering the above issue, Multi-Oriented Smoother (MOS) is designed to improve the reliability of main vein. Specifically, as shown in Algorithm 1 **function** MULTI-ORIENTED SMOOTHER, MOS extracts kernel mask region from $\mathbf{F}_k$ at first. Meanwhile, considering rotating image leads to information loss of text instances at image borders, MOS pads kernel mask $kernel$ by 0 in top, bottom, left, and right directions with h (h is the height of input image). Then, initial center points $cpts$ are sampled by the **function** MIDDLE SAMPLE in Algorithm 1. Next, the angle ϕ between $kernel$ and X-axis is computed and $cpts$ are rotated ϕ with $cpts[0]$ as origin. In the end, main vein $f(x)$ is fitted by the rotated center points $cpts_r$. With a smooth main vein, reliable start points and growth directions of lateral veins can be determined by **function** MAIN in Algorithm 1, which improves the reliability of detection results significantly (verified in Section IV-C).

D. Label Generation

As described in Section III-A, text contour is represented by the combination of main, lateral, and thin veins. In Fig. 3, LeafText predicts kernel mask to extract main veins. Meanwhile, it regresses the lengths of lateral and thin veins in all directions of the predefined polar coordinate system. In this section, we illustrate the label generation process of kernel mask and vein lengths.

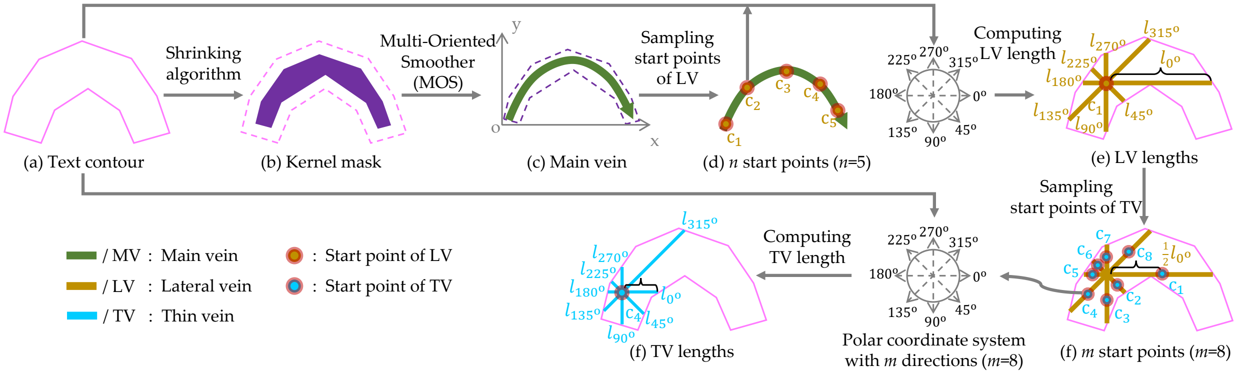
For **the label of kernel mask** (Fig. 4 (b)), the corresponding boundary is generated by shrinking text contour through the algorithm proposed in [53]. The inner region of the boundary is regarded as the kernel mask.

Algorithm 1 Growth of Lateral Vein

```
Require: The kernel mask map $\mathbf{F}_k$;
Ensure: The coordinates of lateral vein start points $cpts^e_r$ and corresponding tangent slopes $\varphi^e$, len($cpts^e_r$)=len($\varphi^e$)=$n, n \geqslant 2$;
function MAIN($\mathbf{F}_k$)
    $f(x)_r \leftarrow$ MULTI-ORIENTED SMOOTHER($\mathbf{F}_k$)
    $cpts^e_r \leftarrow$ equidistantSample($f(x)_r$) //$cpts^e_r$ means rotated equidistant start points of lateral veins sampled from $f(x)_r$, len($cpts^e_r$)=$n$;
    $\varphi^e_r \leftarrow$ tangentSlope($cpts^e_r, f(x)_r$) //$\varphi^e_r$ denote tangent slopes at start points, which can be computed by Equation 2, len($\varphi^e_r$)=$n$;
    $cpts^e \leftarrow$rotate($cpts^e_r, -\phi, cpts[0]$)
    $cpts^e \leftarrow (cpts^e_r - h)$
    $\varphi^e \leftarrow$rotate($\varphi^e_r, -\phi, cpts[0]$)
    return $cpts^e, \varphi^e$
end function

function MULTI-ORIENTED SMOOTHER($\mathbf{F}_k$)
    $h$ , $w \leftarrow$ size($\mathbf{F}_k$)
    $kernel \leftarrow$ padding($(\mathbf{F}_k > 0), h, 0$) //obtaining kernel mask $kernel$ from $\mathbf{F}_k$ and padding it by 0 in top, bottom, left, and right directions with $h$;
    $cpts \leftarrow$ MIDDLESAMPLE($kernel$) //$cpts$ denotes multiple center points of kernel mask, len($cpts$)=$n$;
    $\phi \leftarrow$ angle($\overrightarrow{cpts[0]cpts[-1]}, \overrightarrow{Ox}$) //$\phi$ is the angle between vector $\overrightarrow{cpts[0]cpts[-1]}$ and vector $\overrightarrow{Ox}$
    $cpts_r \leftarrow$ rotate($cpts, \phi, cpts[0]$) // rotating the $cpts$ $\phi$ with $cpts[0]$ as the origin;
    $f(x)_r \leftarrow$ polyFit($cpts_r$) //$f(x)_r$ is rotated $f(x)$;
    return $f(x)_r$
end function

function MIDDLESAMPLE($kernel$)
    initial $cpts \leftarrow [\varnothing]$
    $pts_k \leftarrow$coordinate($kernel$) //$pts_k$ are the point coordinates of kernel region;
    $x_{min} \leftarrow$min($pts^x_k$), $x_{max} \leftarrow$max($pts^x_k$) //$pts^x_k$ are the x coordinates of $pts_k$;
    $y_{min} \leftarrow$min($pts^y_k$), $y_{max} \leftarrow$max($pts^y_k$) // $pts^y_k$ are the y coordinates of $pts_k$;
    for $i = 1 \rightarrow n$ do
        if $x_{max} - x_{min} > y_{max} - y_{min}$ then
            $x_{sample} \leftarrow x_{min} + i \times \frac{x_{max}-x_{min}}{n+1}$
            $y_s \leftarrow pts_k[pts_k[:,0] == x_{sample}][:,1]$
            $y_s \leftarrow sort(y_s)$
            $y_{sample} \leftarrow y_s[len(y_s)//2]$
        else
            $y_{sample} \leftarrow y_{min} + i \times \frac{y_{max}-y_{min}}{n+1}$
            $x_s \leftarrow pts_k[pts_k[:,1] == y_{sample}][:,0]$
            $x_s \leftarrow sort(x_s)$
            $x_{sample} \leftarrow x_s[len(x_s)//2]$
        end if
        $cpts[i] \leftarrow (x_{sample}, y_{sample})$
    end for
    return $cpts$
end function
```



(a) Text contour (b) Kernel mask (c) Main vein (d) n start points (n=5) (e) LV lengths (f) m start points (m=8) (f) TV lengths

Fig. 4. Visualization of the label generation process. Kernel mask (b) is used for extracting main vein. It is the ground-truth of MV header in Fig. 3. For Lengths of lateral and thin veins (e) and (f) in all directions of predefined polar coordinate system, they are responsible for supervising LV-TV header of LeafText in the training process.

For **the label of lateral vein length**, main vein (Fig. 4 (c)) is extracted from kernel mask by the function MULTI-ORIENTED SMOOTHER in Algorithm 1 at first. Then, the start points and growth directions are determined (the details refer to function MAIN in Algorithm 1 and Section III-A, respectively). In the end, the lengths between start points and text contour in m directions of the predefined polar coordinate system are computed. For **the label of thin vein length**, start points are sampled according to the lateral vein, and the lengths are computed in the same way as lateral vein.

E. Loss Function

LeafText determines main, lateral, and thin veins by MV and LV-TV Header (as shown in Fig. 3). In this paper, to optimize the proposed pipeline effectively, we propose a multi-task loss function $\mathcal{L}$ (referred to Equation 8). It consists of two loss functions of MV header loss $\mathcal{L}_{mv}$ and LV-TV header loss $\mathcal{L}_{lv-tv}$, which are responsible for supervising the corresponding headers in the training stage, respectively.

$$\mathcal{L} = \alpha\mathcal{L}_{mv} + \beta\mathcal{L}_{lv-tv}, \tag{8}$$

where α and β are the coefficients of $\mathcal{L}_{mv}$ and $\mathcal{L}_{lv-tv}$. They are set to 1 and 0.25 in following experiments.

Optimization of MV header. Dice loss [54] is designed for segmentation tasks where there is a strong imbalance between the positive and negative samples. Considering the regions of kernel mask are much smaller than background, Dice loss is adopted to evaluate the MV header loss $\mathcal{L}_{mv}$:

$$\mathcal{L}_{mv} = 1 - \frac{2 \times |\mathbf{K}_{pre} \cap \mathbf{K}_{gt}| + \varepsilon}{|\mathbf{K}_{pre}| + |\mathbf{K}_{gt}| + \varepsilon}, \tag{9}$$

where $\mathbf{K}_{pre}$ and $\mathbf{K}_{gt}$ denote the predicted kernel mask and the corresponding ground-truth. To avoid the situation that there may be no positive samples in $\mathbf{K}_{gt}$, we set ε as 1 to ensure that the denominator of $\mathcal{L}_{mv}$ is bigger than 0.

Optimization of LV-TV header. As we can see from Fig. 3, LV-TV header is responsible for regressing the lengths of lateral and thin veins. To facilitate the optimization of regression tasks, global incentive loss $\mathcal{L}_g$ is proposed to supervise LV-TV header in this paper:

$$\mathcal{L}_{lv-tv} = \mathcal{L}_g. \tag{10}$$

Global incentive loss $\mathcal{L}_g$ aims to force our model to balance the importance of text instances with different scales and focus on the prediction of lateral and thin veins.

Specifically, to keep the same sensitivity for multi-scale texts, $\mathcal{L}_g$ replaces L2-loss or Smooth-l_1 loss used in [31], [35] by negative logarithm loss $\mathcal{L}_{nl}$ (as shown in Equation 11). It scales the differences between predicted lengths and ground-truth into the range of 0–1, which ensures the effectiveness of our model to large and small text instances simultaneously.

$$\mathcal{L}_{nl} = -\log\left(\frac{\min(l_{pre}, l_{gt})}{\max(l_{pre}, l_{gt})}\right), \tag{11}$$

where l_{pre} and l_{gt} are the predicted lengths of lateral and thin veins and the corresponding ground-truth.

Moreover, as described in Section III-A and Section III-B, LeafText determines one lateral vein or thin vein from m directions. It leads to an overwhelming number of indirect samples and a small number of direct samples (lateral and thin veins). To make training more effective and efficient, we propose an incentive strategy for direct samples. It is found in Fig. 5 that the lengths of direct samples are smaller than indirect ones for a specific dataset. Therefore, a incentive coefficient λ is formulated as follow:

$$\lambda = \tanh(\rho(1 - (l_{gt}/l_s))), \tag{12}$$

where l_s denotes the shorter side size of resized input images in the training and testing stages. ρ is responsible for scaling λ in to the range of 0–1.

By combining the Equation 11 and 12, global incentive loss $\mathcal{L}_g$ can be formulated as:

$$\mathcal{L}_g = \frac{1}{T \times M}\sum_{t=1}^{T}\sum_{m=1}^{M}\lambda^{(t,m)}\mathcal{L}_{nl}^{(t,m)}, \tag{13}$$

where T is the sum of number of all lateral and thin vein start points. M denotes the direction number of the predefined polar coordinate system.

IV. Experiments

A. Datasets

To demonstrate the strong ability of LVT to fit arbitrary-shaped texts, we analyze the upper bound of the IoU between

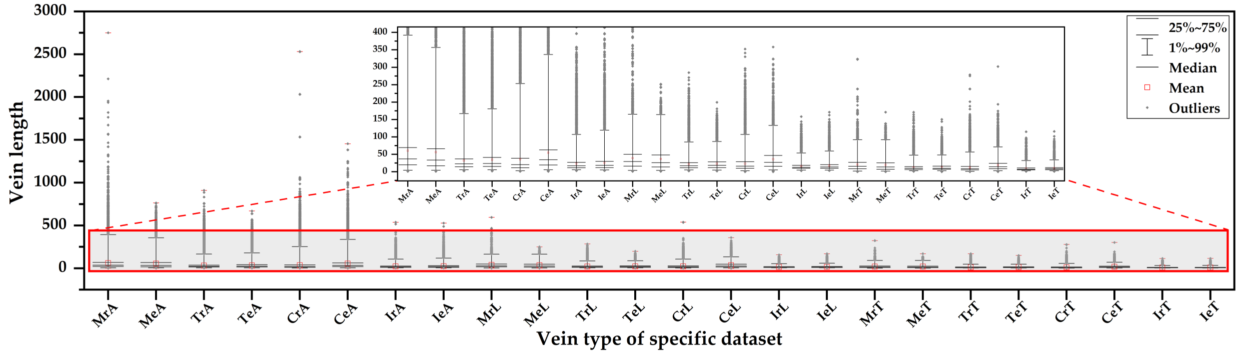


Fig. 5. Illustration of the distributions of lateral and thin vein lengths on different public benchmarks. ‘M’, ‘T’, ‘C’, and ‘I’ indicate MSRA-TD500, Total-Text, CTW1500, and ICDAR2015 datasets respectively. ‘r’ and ‘e’‘ are training and testing samples. ‘A’ denotes the lengths in all directions (referred to Fig. 4). ‘L’ and ‘T’ are lengths in lateral and thin vein directions.

generated label and ground-truth. Meanwhile, the effectiveness of MOS and global incentive loss $\mathcal{L}_g$ are verified. Moreover, the proposed LeafText is evaluated on multiple representative public benchmarks to show the superior performance.

SynthText [55] contains 800k composite training samples that are combined by synthetic varied text instances and scene RGB images. It is proposed to pre-train the model to improve the robustness of the proposed LeafText.

MSRA-TD500 [56] includes line-level Chinese and English text instances simultaneously. It is conposed of 300 training images and 200 testing images, respectively. To ensure a fair comparison environment, 400 images of HUST-TR400 [57] are extra introduced as training data.

Total-Text [58] consists of word-level arbitrary-shaped multilingual texts, which brings significant challenges for model generalization. There are 1255 images for training model and 300 images for evaluating performance.

CTW1500 [59] is composed of 1500 samples, where includes 1000 training images and 500 testing images. Particularly, CTW1500 mainly contains line-level arbitrary-shaped text instances, which requires model’s strong ability to deal with large-scale and ratio objects.

ICDAR2015 [60] is proposed in ICDAR 2015 Robust Reading Competition, which has 1000 training image and 500 testing images. Different from the above three public benchmarks, the background of ICDAR2015 images are more complicated. Meanwhile, the text instances enjoys similar basic features with background, which brings much difficulties for text detection.

B. Implementation Details

The overall pipeline of the proposed LeafText is depicted in Fig. 3. The backbone adopts ResNet [51] directly and the details of FPN can be referred to [52]. MV header and LV-TV header are composed of one 3×3 convolutional layer and m 3×3 convolutional layer, respectively.

In the pre-process stage, training samples can be obtained through data augmentation and label generation operators. For the former, it contains the following strategies: (1) random scaling (including image size and aspect); (2) random horizontal flipping; (3) random rotating in the range of (-10, 10); (4) random cropping and padding. For the latter, kernel mask and the lengths of lateral and thin veins in m directions are generated by the process in Fig. 5. Different from training samples, testing samples are produced by resizing input RGB images into specific sizes only. Particularly, the text instances labeled as DO NOT CARE are ignored during both the training and testing stages.

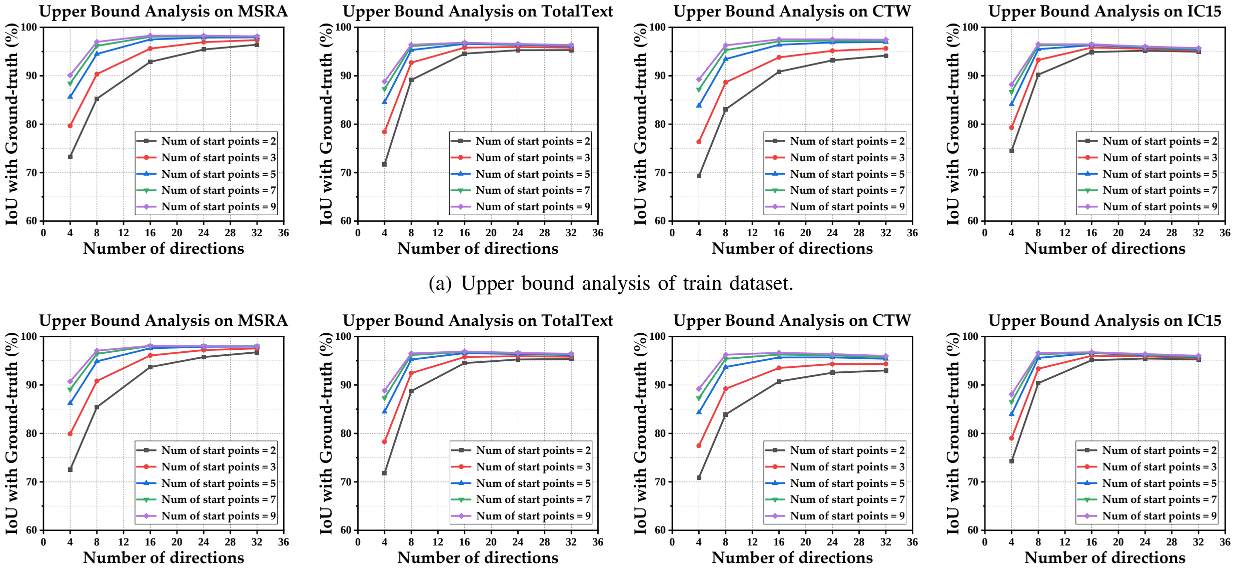
In the training stage, the weights of the CNN network are initialized first. Specifically, the backbone is pre-trained on the ImageNet [61]. For the FPN and headers, they are initialized by the strategy proposed in [62]. To ensure an efficient and effective converge process, Adam [63] is adopted as the optimizer. The learning rate is set as 0.001 and adjusted through ’polylr’ strategy with the model converging. In the comparison experiments, our model is trained on the SynthText dataset for 1 epoch at first. Then, it is finetuned on the official datasets (MSRA-TD500, Total-Text, CTW1500, and ICDAR2015) for 600 epochs with a batch size of 16. All the experiments in this paper are conducted on a workstation with RTX 1080Ti GPU.

C. Ablation Study

To verify the effectiveness of LeafText, we conduct ablation experiments on multiple public benchmarks in this section. Specifically, to verify the strong fitting ability of LVT, we analyze the upper bound IoU between reconstructed text contour and ground-truth. Meanhiwle, the superiority of the proposed global incentive loss $\mathcal{L}_g$ is demonstrated by comparing it with exisiting loss functions. Furthermore, the importance of MOS for rebuilding text contours is verified. The details of experimental results are described in the following paragraphs.

Upper Bound Analysis of LVT. Considering existing approaches fail to fit irregular-shaped texts accurately, a leaf vein-based text representation method is proposed.

To verify the effectiveness of it, we analyze the upper bound of IoU that between rebuilt text contour based on generated label and ground-truth. Specifically, as shown in Fig. 6, the IoU can achieve 96% at least on both training and testing samples of four public benchmarks (MSRA-TD500, Total-Text, CTW1500, and ICDAR2015). For some highly curved text instances (as visualized in Fig 7), LVT still can



(a) Upper bound analysis of train dataset.

(b) Upper bound analysis of test dataset.

Fig. 6. Upper Bound Analysis. More start points and directions of lateral vein can model text contours with higher IoU with Ground Truth. 'Directions' are the vein directions in predefined polar coordinate system.

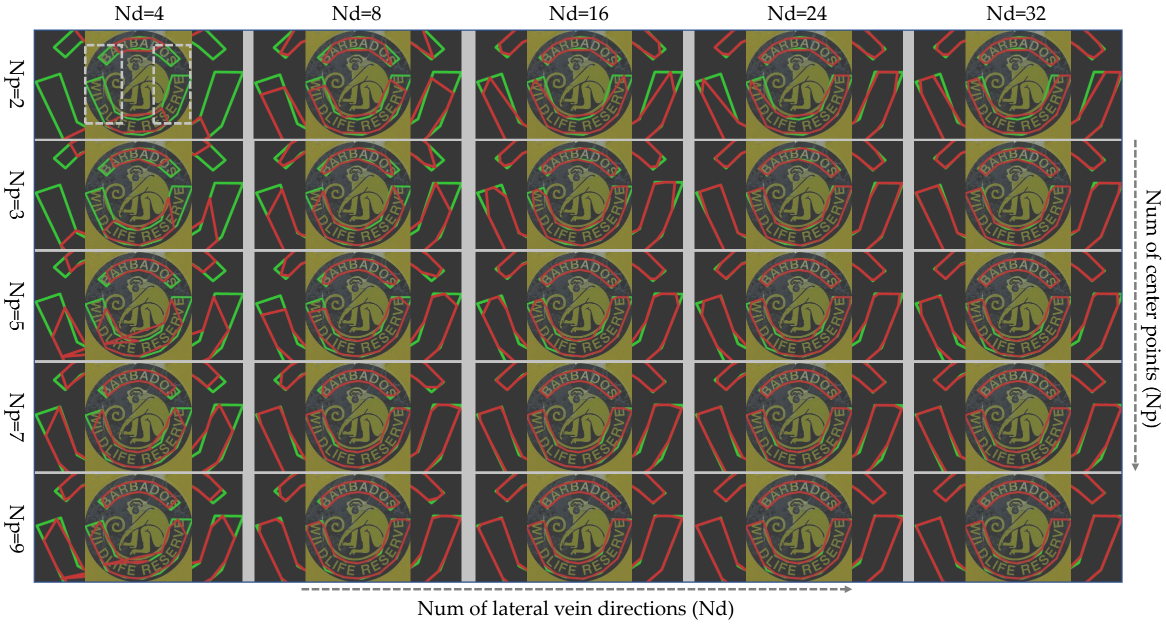

Nd=4
Nd=8
Nd=16
Nd=24
Nd=32
Np=2
Np=3
Np=5
Np=7
Np=9
Num of center points (Np)
Num of lateral vein directions (Nd)

Fig. 7. Visualization of the proposed leaf vein based text representation method. We aim to treat text contour as leaf margin and construct it through the combination of main vein, lateral vein, and thin vein.

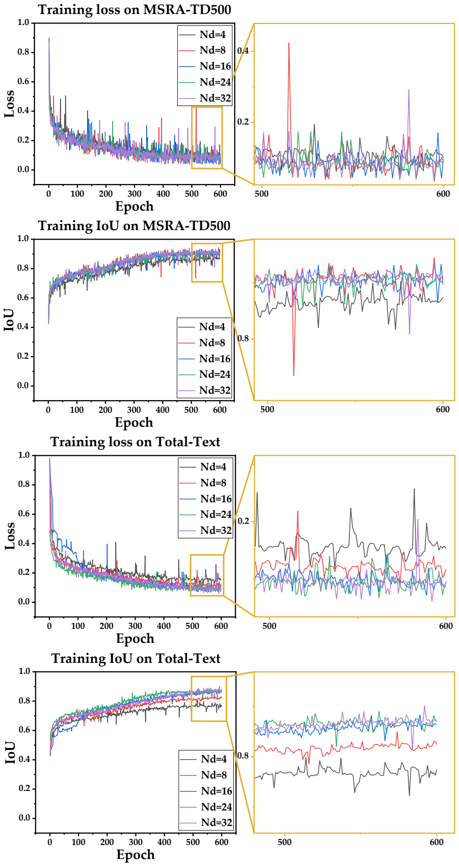
achieve superior performance. The results demonstrate the strong fitting ability of the proposed leaf vein-based text representation method for arbitrary-shaped texts.

Moreover, as described in Section III-A, the reconstruction process of LVT relies on the start points of lateral veins and the directions of predefined polar coordinate system. Therefore, we further explore the influences brought by the different numbers of the start points and directions ($\mathrm{N_p}$ and $\mathrm{N_d}$ in Fig. 7). Concretely, as we can see from Fig. 7, the IoU is evaluated when tuning $\mathrm{N_p}$ and $\mathrm{N_d}$, respectively. It is found that there is a significant increase for IoU with $\mathrm{N_p}$ being tuned from 2 to 5. The upper bound of IoU continues to slow-growing when $\mathrm{N_p}$ is set to 7 and 9, which shows the start points of lateral veins play an important role in representing texts. Furthermore, the relations between IoU and $\mathrm{N_d}$ are visualized in this section. Concretely, in Fig. 7, larger $\mathrm{N_d}$ brings improvements to the fitting ability of our method, which verifies the importance of $\mathrm{N_d}$ for leaf vein-based text representation method.

Performance Analysis under Different $\mathrm{N_d}$ and $\mathrm{N_p}$. We have analyzed the upper bound IoU of the proposed LVT on different kinds of text instances in Section IV-C. To further verify the model's performance, LeafText is trained and evaluated under different $\mathrm{N_d}$ and $\mathrm{N_p}$ on MSRA-TD500

N_d	N_p	MSRA-TD500			Total-Text		
		Precision	Recall	F-measure	Precision	Recall	F-measure
4	2	86.9	77.8	82.1	78.6	67.4	72.6
	3	88.3	78.9	83.3	85.5	73.4	79.0
	5	88.5	79.1	83.5	89.0	76.3	82.1
	7	88.5	79.1	83.5	88.6	75.9	81.8
	9	88.2	78.8	83.2	89.1	76.3	82.2
8	2	90.7	80.1	85.1	88.8	76.6	82.3
	3	91.5	80.4	85.6	90.5	78.2	83.9
	5	92.0	80.5	**85.9**	91.1	78.6	84.4
	7	91.9	80.4	**85.8**	91.2	78.7	84.5
	9	92.1	80.6	**86.0**	91.1	78.6	84.4
16	2	88.7	80.7	84.5	88.6	83.0	85.7
	3	89.1	81.0	84.9	88.9	83.4	**86.1**
	5	89.1	81.0	84.9	89.0	83.5	**86.2**
	7	88.9	80.8	84.7	89.0	83.5	**86.2**
	9	89.1	81.0	84.9	88.8	83.4	**86.0**
24	2	86.6	80.9	83.7	89.3	81.5	85.2
	3	86.6	80.9	83.7	89.6	81.8	85.5
	5	87.2	81.4	84.2	89.3	81.6	85.3
	7	87.0	81.2	84.0	89.4	81.7	85.4
	9	87.2	81.4	84.2	89.4	81.7	85.4
32	2	87.9	79.8	83.7	89.4	76.2	82.3
	3	88.5	80.3	84.2	89.6	76.5	82.5
	5	88.7	80.5	84.4	89.7	77.4	83.1
	7	88.7	80.5	84.4	89.1	76.3	82.2
	9	88.7	80.5	84.4	89.3	76.4	82.3

(a) Table of experimental results



(b) Curves of training loss and IoU.

Fig. 8. Ablation study for the impact of N_d and N_p on detection performance. N_d indicates the direction number predefined in a polar coordinate system. N_p means the number of center points sampled on the main vein for reconstructing text contours. **red**, **green**, and **blue** are the experimental results with three best groups of settings respectively on MSRA-TD500 and Total-Text datasets. 'IoU' in (b) indicates the Intersection of Union between predicted shrink-mask and the corresponding ground-truth.

TABLE I
THE DETECTION RESULTS OF THE MODELS EQUIPPED WITH MOS AND W/O MOS ON MSRA-TD500 AND TOTAL-TEXT DATASETS.

MOS	N_d	N_p	MSRA-TD500		
			Precision	Recall	F-measure
×	8	9	91.6	78.8	84.7
✓			92.1	80.6	86.0
MOS	N_d	N_p	Total-Text		
			Precision	Recall	F-measure
×	16	5	88.3	82.1	85.1
✓			89.0	83.5	86.2

and Total-Text text benchmarks.

Specifically, as we can see from the table (a) in Fig. 8, for multi-oriented texts (MSRA-TD500), LeafText achieves the best performance when N_d and N_p are set to 8 and 9, respectively. Meanwhile, the F-measure begins to decrease with the increase of N_d. For irregular-shaped text instances, our method achieves 86.2% in F-measure when N_d and N_p are set as 16 and 5, which outperforms the rest of the other models. The above results show the best settings of N_d and N_p to detect multi-oriented and irregular-shaped texts. Furthermore, we visualize the details of the training process in Fig. 8 (b). It can be found from the curves of the training IoU on MSRA-TD500 that the IoU is smaller than the model under other settings when N_d equals 8, which matches the results of Table (a) in Fig. 8. Meanwhile, the curves of the training IoU and loss on Total-Text show the unsatisfied convergence process when N_d is set to 4 and 8, which verifies the effectiveness of large N_d for irregular-shaped texts. The above experimental results provide appropriate model settings for the following comparison experiments on different kinds of text instances.

Effectiveness of MOS. As described in Section III-C, for improving the accuracy of reconstructed text instance, MOS is designed to ensure the reliability of the main vein extracted from the predicted unreliable kernel mask. To demonstrate the effectiveness of MOS, we analyze the improvements in detection performance brought by MOS and visualize some qualitative results. As shown in Table I, MOS can bring improvements in F-measure on both multi-oriented (MSRA-TD500) and irregular-shaped (Total-Text) datasets. Specifically, LeafText with MOS achieves 86.0% and 86.2% F-measure on the two benchmarks respectively, which surpasses LeafText without MOS 1.3% and 1.1%. These experimental

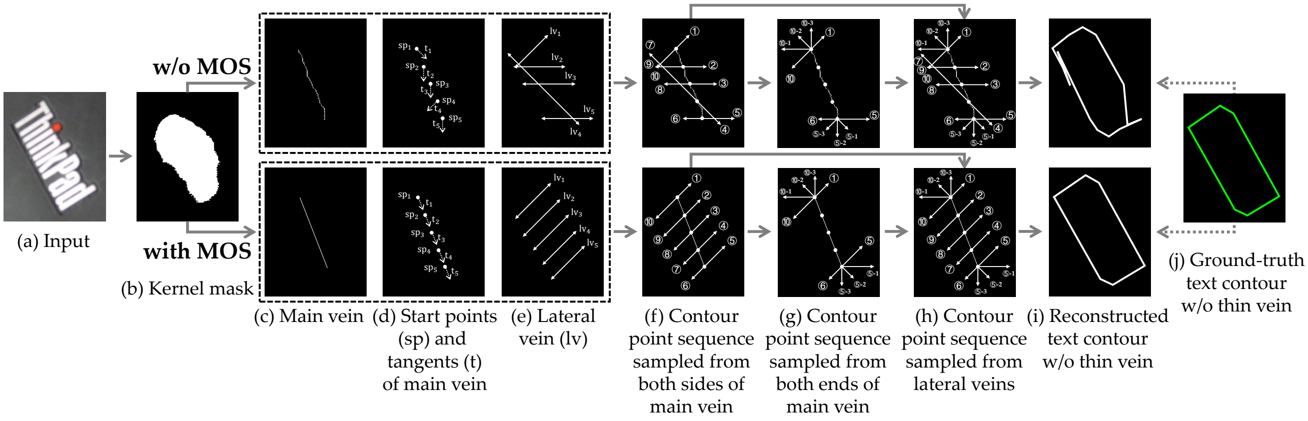

w/o MOS
with MOS
sp1 t1 sp2 t2 sp3 t3 sp4 t4 sp5 t5
lv1 lv2 lv3 lv4 lv5
(a) Input
(b) Kernel mask
(c) Main vein
(d) Start points (sp) and tangents (t) of main vein
(e) Lateral vein (lv)
(f) Contour point sequence sampled from both sides of main vein
(g) Contour point sequence sampled from both ends of main vein
(h) Contour point sequence sampled from lateral veins
(i) Reconstructed text contour w/o thin vein
(j) Ground-truth text contour w/o thin vein

Fig. 9. Visualization of the differences between the text contour reconstruction processes with MOS and w/o MOS. The sample is picked from MSRA-TD500 dataset and the $\mathrm{N_d}$ and $\mathrm{N_p}$ of model are set to 8 and 5, respectively.

TABLE II
THE DETECTION RESULTS OF THE MODELS TRAINED BY DIFFERENT LOSS FUNCTIONS ON MSRA-TD500 AND TOTAL-TEXT DATASETS.

Dataset	Loss	Precision	Recall	F-measure
MSRA-TD500	Smooth-L1	89.2	77.3	82.8
	L2	90.0	71.1	79.4
	Global incentive	92.1	80.6	86.0
Total-Text	Smooth-L1	88.5	80.1	84.1
	L2	88.7	74.5	81.0
	Global incentive	89.0	83.5	86.2

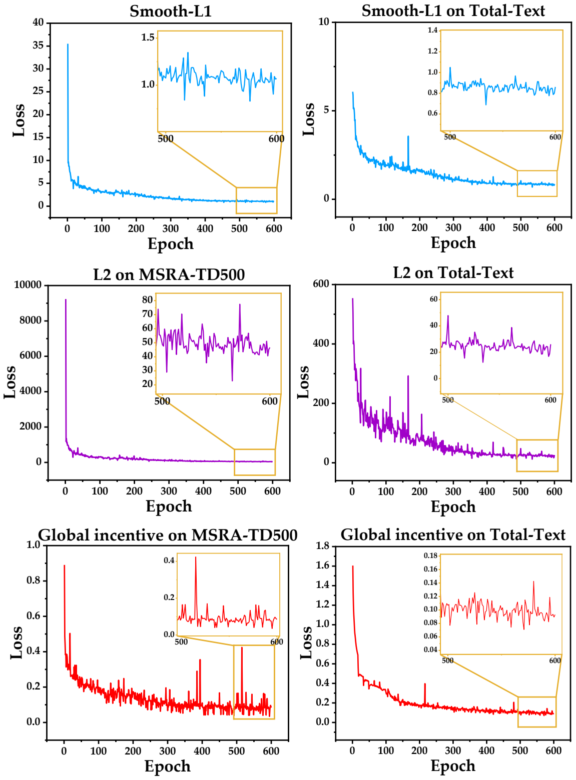


Fig. 10. Visualization of the training processes on the MSRA-TD500 and Total-Text with different loss functions.

results demonstrate the effectiveness of MOS for improving the quality of rebuilt contours. To further explain how MOS works for smoothing the main vein, we visualize the process details in Fig. 9. Concretely, given a predicted kernel mask (Fig. 9 (b)), MOS helps our method to determine correct tangent directions on each start point (Fig. 9 (d)), which helps avoid disordered contour point sequence (Fig. 9 (f)) and improve the reliability of reconstructed text contour (Fig. 9 (i)) effectively. The visualization demonstrates the effectiveness of MOS and depicts the differences between the text contour reconstruction processes with MOS and w/o MOS vividly.

Effectiveness of Global Incentive Loss. Considering existing L2-loss and Smooth-l_1 loss mainly focus on large samples, which leads to the ignorance of small objects, global incentive loss $\mathcal{L}_g$ is designed to force our model to balance the importance of texts with different scales and focus on the prediction of lateral and thin veins.

As shown in Table II, compared with existing L2-loss and Smooth-l_1 loss, training LeafText by the proposed $\mathcal{L}_g$ brings 3.2% and 2.1% improvements in F-measure on MSRA-TD500 and Total-Text at least, respectively. Considering there existing lots of large and small texts simultaneously in MSRA-TD500, the above experimental results demonstrate $\mathcal{L}_g$ can help the model improve the ability to deal with different sized text instances. Meanwhile, the results in Table II verify that our method can regress lateral and thin vein lengths more accurately when supervising the LV-TV prediction header by $\mathcal{L}_g$. Furthermore, we visualize the training processes of different losses in Fig. 10. It is found that global incentive loss function $\mathcal{L}_g$ fluctuates around 0.1 at the end of the convergence process on MSRA-TD500 and Total-Text datasets simultaneously. Compared with the loss functions of Smooth-L1 and L2, the proposed $\mathcal{L}_g$ can accelerate the model converging effectively and improve the model's ability to learn text features. The above results demonstrate the effectiveness of the proposed global incentive loss $\mathcal{L}_g$ for detecting multi-scaled texts.

Superiority of the Thin Vein. As described in Section III-A, the thin vein is designed for fining text contours, which supports accurately fitting texts with lower model complexity. Benefiting from the advantage that the thin vein length is half of the lateral vein, the thin vein eases the learning of

TABLE III
IMPACT OF TV FOR DETECTION RESULTS ON MSRA-TD500 AND TOTAL-TEXT DATASETS. 'LV' AND 'TV' DENOTE LATERAL VEIN AND THIN VEIN, RESPECTIVELY. 'MAE' MEANS MEAN ABSOLUTE ERROR.

TV	MAE		MSRA-TD500		
	LV	TV	Precision	Recall	F-measure
×	11.3	11.1	91.8	80.2	85.6
✓			92.1	80.6	86.0
TV	MAE		Total-Text		
	LV	TV	Precision	Recall	F-measure
×	5.8	5.3	88.1	82.7	85.3
✓			89.0	83.5	86.2

TABLE IV
PERFORMANCE COMPARISON ON MSRA-TD500 DATASET.

Methods	Precision	Recall	F-measure
MOTD [11] (CVPR 2016)	83.0	67.0	74.0
EAST [34] (CVPR 2017)	87.3	67.4	76.1
SegLink [2] (CVPR 2017)	86.0	70.0	77.0
PixelLink [14] (AAAI 2018)	83.0	73.2	77.8
TextSnake [28] (ECCV 2018)	83.2	73.9	78.3
RRD [36] (CVPR 2018)	87.0	73.0	79.0
CornerNet [13] (CVPR 2018)	87.6	76.2	81.5
CRAFT [26] (CVPR 2019)	88.2	78.2	82.9
TextField [16] (TIP 2019)	87.4	75.9	81.3
SAE [15] (CVPR 2019)	84.2	81.7	82.9
ATRR [9] (CVPR 2019)	85.2	82.1	83.6
PAN [18] (ICCV 2019)	84.4	83.8	84.1
DB [20] (AAAI 2020)	90.4	76.3	82.8
DRRG [27] (CVPR 2020)	88.1	82.3	85.1
OPMP [25] (TMM 2021)	86.0	83.4	84.7
PAN++ [19] (TPAMI 2021)	85.3	84.0	84.7
SAVTD [7] (CVPR 2021)	89.2	81.5	85.2
GV [8] (TPAMI 2021)	88.8	84.3	86.5
ReLaText [41] (PR 2021)	90.5	83.2	**86.7**
LPAP [5] (TOMM 2022)	87.9	77.7	82.5
DC [6] (PR 2022)	87.9	83.1	85.4
Res18-DB++ [21] (TPAMI 2022)	87.9	82.5	85.1
Res50-DB++ [21] (TPAMI 2022)	91.5	83.3	**87.2**
Res50-Pre-Ours (736)	92.1	83.8	**87.8**

contour point sequence and ensures accurate detection results. To verify the superiority of the thin vein, we evaluate the accuracy of the lateral and thin vein in table III. We first evaluate the Mean Absolute Error (MAE) of the lateral vein and the thin vein. It is found that the MAE of the lateral vein surpasses the thin vein 0.2 and 0.5 on MSRA-TD500 and Total-Text. It demonstrates the task of thin vein prediction is more accessible than the prediction of the lateral vein, which verifies the advantage of thin vein that can ease the learning of contour point sequence. Meanwhile, thin vein brings 0.4% and 0.9% in F-measure on MSRA-TD500 and Total-Text, respectively. The above experimental results prove thin vein can promote the model performance in the detection of text instances effectively.

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

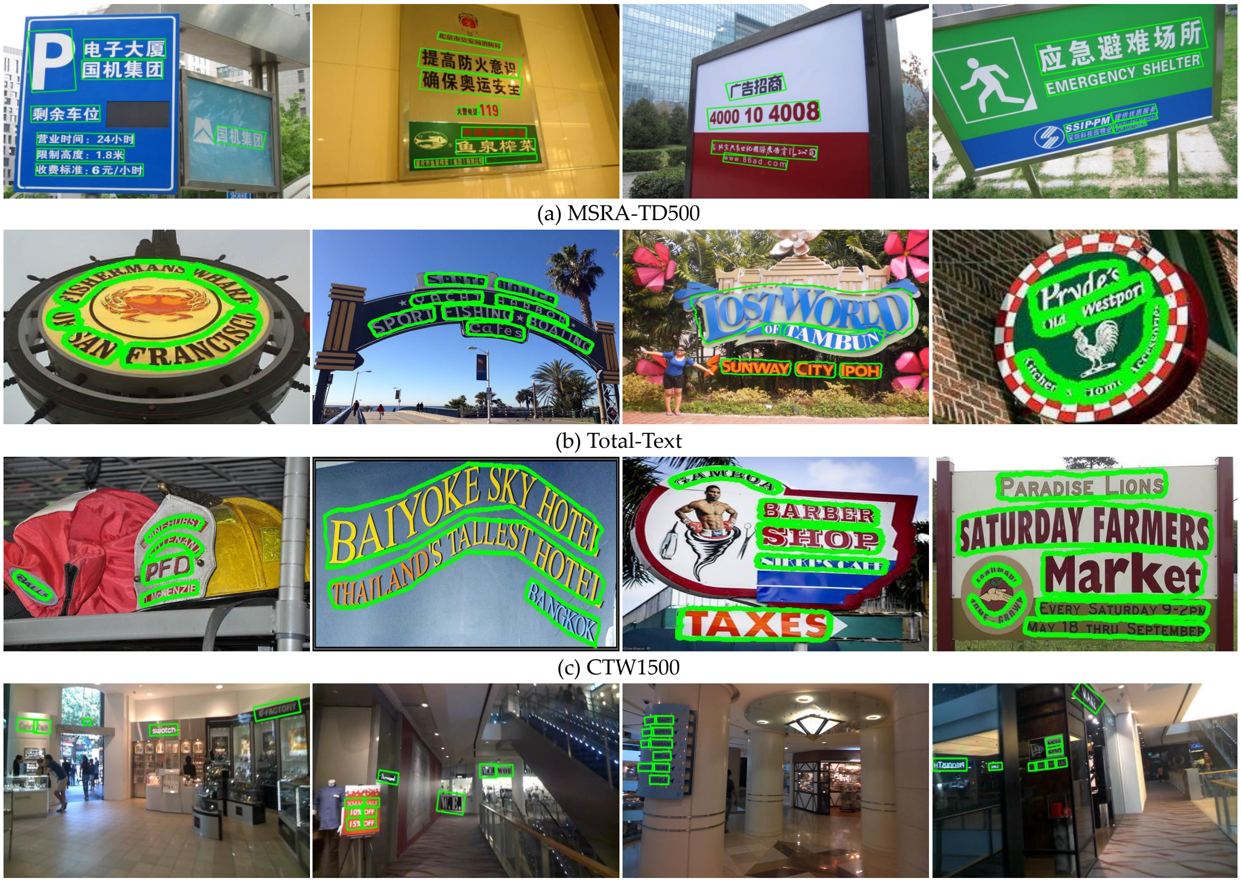
To demonstrate the superior performance of LeafText for detecting texts with arbitrary shapes, multi scales, and multilingual, we compare it with the existing state-of-the-art (SOTA) approaches on four representative public benchmarks (MSRA-TD500, Total-Text, CTW1500, and ICDAR2015) in this section. Meanwhile, the advantages of our method over previous methods are analyzed based on the comparisons and quality detection results.

Evaluation on MSRA-TD500. To verify the performance for detecting line-level multi-oriented text instances, we evaluate the proposed LeafText on the MSRA-TD500 dataset. As shown in Table IV, for existing state-of-the-art (SOTA) methods, ReLaText [41], GV [8], and DC [6] achieve 86.7%, 86.5%, and 85.4% in F-measure. Benefiting from decomposing long texts into multiple characters and the strong connection ability of Graph Convolutional Network (GCN), ReLaText surpasses GV and DC in F-measure 0.2% and 1.3% respectively. Unlike ReLaText, LeafText models the whole text directly, which effectively avoids the character ignorance problem and improves detection performance. Specifically, our method achieves 87.8% in F-measure on MSRA-TD500, which surpasses the best existing method ReLaText 1.1%. For DB++ [21], though it achieves significant improvement by embedding DConv [64] into the corresponding backbone, our method still outperforms it with basic network. We show some qualitative results on MSRA-TD500 in Fig. 11 (a). The above results demonstrate the superior ability of LeafText for detecting very long, multi-oriented, and multi-lingual texts.

Evaluation on Total-Text and CTW1500. Irregular-shaped texts bring challenges to existing text detection methods. LeafText represents contours through point sequences for improving the text fitting ability. To verify the effectiveness of our method for the detection of irregular-shaped texts, we make comparisons on the Total-Text and CTW1500 simultaneously. We first resize the short sizes of images into 640 while keeping the original ratio and evaluate the model performance with the backbones of ResNet-18 and ResNet-50, respectively.

As we can see from Table V, for the detection of word-level text instances in Total-Text, DC [6] and DB++ [21] achieve 86.4% and 86.0% in F-measure, they can surpass previous methods up to 7.5%. On this challenging dataset, LeafText achieves the SOTA performance of 87.3% in F-measure and exceeds DC [6] by 0.9%, which demonstrates the effectiveness of the proposed LVT and the superiority over the existing text representation methods. Meanwhile, the thin vein is helpful for detecting large-scaled instances, which further improves the model detection performance on the Total-Text.

Different from Total-Text, CTW1500 is composed of line-level text instances that contain large spaces between different characters or words, which brings challenges to existing methods. As shown in Table V, DB++ [21] and TextDCT [49] are latest SOTA methods on CTW1500 benchmark. They achieve 85.3% and 85.1% in F-measure, respectively. A similar conclusion on the CTW1500 dataset can be generated that our method is superior to previous methods. Specifically, our method achieves 85.5% in F-measure, which surpasses DB++ [21] 0.2% even it is equipped with DConv [64] and



(a) MSRA-TD500

(b) Total-Text

(c) CTW1500

(d) ICDAR2015

Fig. 11. Visualization of the differences between the text contour reconstruction processes with MOS and w/o MOS. The sample is picked from MSRA-TD500 dataset and the $\mathrm{N_d}$ and $\mathrm{N_p}$ of model are set to 8 and 5, respectively.

TABLE V
PERFORMANCE COMPARISON ON TOTAL-TEXT AND CTW1500 DATASETS.

Methods	Total-Text			CTW1500		
	Precision	Recall	F-measure	Precision	Recall	F-measure
TextSnake [28] (ECCV 2018)	82.7	74.5	78.4	67.9	85.3	75.6
ATTR [9] (CVPR 2019)	80.9	76.2	78.5	80.1	80.2	80.1
CRAFT [26] (CVPR 2019)	87.6	79.9	83.6	86.0	81.1	83.5
CTD [23] (ICDAR 2019)	80.6	82.3	81.4	79.9	77.0	78.5
LOMO [43] (CVPR 2019)	87.6	79.3	83.3	85.7	76.5	80.8
PSE [17] (CVPR 2019)	84.0	78.0	80.9	84.8	79.7	82.2
SegLink++ [38] (PR 2019)	82.1	80.9	81.5	82.8	79.8	81.3
TextDragon [40] (ICCV 2019)	85.6	75.7	80.3	84.5	82.8	83.6
Boundary [44] (AAAI 2020)	85.2	83.5	84.3	–	–	–
ContourNet [45] (CVPR 2020)	86.9	83.9	85.4	83.7	84.1	83.9
TextRay [47] (ACMMM 2020)	83.5	77.9	80.6	82.8	80.4	81.6
Spotter [22] (TPAMI 2021)	88.3	82.4	85.2	–	–	–
FCENet [50] (CVPR 2021)	87.4	79.8	83.4	85.4	80.7	83.1
PSE+STKM [1] (CVPR 2021)	86.3	78.4	82.2	85.1	78.2	81.5
OPMP [25] (TMM 2021)	88.5	82.9	85.6	85.1	80.8	82.9
ASTD [24] (TMM 2022)	85.4	81.2	83.2	86.2	80.4	83.2
TextDCT [49] (TMM 2022)	87.2	82.7	84.9	85.0	85.3	**85.1**
LPAP [5] (TOMM 2022)	87.3	79.8	83.4	84.6	80.3	82.4
DC [6] (PR 2022)	90.5	82.7	**86.4**	86.9	82.7	84.7
Res50-DB++ [21] (TPAMI 2022)	88.9	83.2	**86.0**	87.9	82.8	**85.3**
Res18-Pre-Ours (640)	90.8	84.0	**87.3**	87.1	83.9	**85.5**

TABLE VI
PERFORMANCE COMPARISON ON ICDAR2015 DATASET.

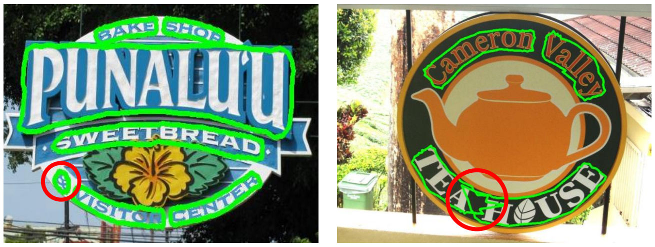
Methods	Precision	Recall	F-measure
WordSup [39] (ICCV 2017)	79.3	77.0	78.2
MCN [3] (CVPR 2018)	72.0	80.0	76.0
PixelLink [14] (AAAI 2018)	85.5	82.0	83.7
TextBoxes++ [33] (TIP 2018)	87.8	78.5	82.9
PSE [17] (CVPR 2019)	86.9	84.5	85.7
RRD [36] (CVPR 2019)	88.0	80.0	83.8
SegLink++ [38] (PR 2019)	83.7	80.3	82.0
Boundary [44] (AAAI 2020)	88.1	82.2	85.0
FCENet [50] (CVPR 2021)	85.1	84.2	84.6
Spotter [22] (TPAMI 2021)	85.8	81.2	83.4
PAN++ [19] (TPAMI 2021)	85.9	80.4	83.1
EAST+STKM [1] (CVPR 2021)	88.7	84.9	**86.8**
PSE+STKM [1] (CVPR 2021)	87.8	84.1	85.9
ASTD [24] (TMM 2022)	87.2	81.3	84.1
TextDCT [49] (TMM 2022)	88.9	84.8	**86.8**
LPAP [5] (TOMM 2022)	88.7	84.4	**86.5**
Res50-Pre-Ours (1152)	88.9	82.3	**86.1**

TABLE VII
CROSS-DATASET EVALUATIONS ON WORD-LEVEL (ICDAR2015 AND TOTAL-TEXT) AND LINE-LEVEL (MSRA-TD500 AND CTW1500) DATASETS.

Type	Methods	Training	Testing	P	R	F
word-level	TextField [16]	IC15	TT	61.5	65.2	63.3
	CM-Net [4]			75.8	64.5	69.7
	Res18-Pre-Ours			89.2	80.0	84.4
	TextField [16]	TT	IC15	77.1	66.0	71.1
	CM-Net [4]			76.5	68.1	72.1
	Res18-Pre-Ours			83.0	69.9	75.9
line-level	TextField [16]	MSRA	CTW	75.3	70.0	72.6
	CM-Net [4]			77.2	69.7	72.8
	Res18-Pre-Ours			83.8	75.0	79.2
	TextField [16]	CTW	MSRA	85.3	75.8	80.3
	CM-Net [4]			85.8	77.1	81.2
	Res18-Pre-Ours			82.9	82.0	82.4

complicated backbone (ResNet-50). The experimental results on Total-Text and CTW1500 prove the superiority of the proposed LVT for fitting irregular-shaped texts. Meanwhile, the strong ability to effectively detect word-level and line-level instances simultaneously is verified. Some qualitative results on Total-Text and CTW1500 are depicted in Fig. 11 (b) and (c) for further demonstrating the effectiveness of LeafText.

Evaluation on ICDAR2015. The images in International Conference on Document Analysis and Recognition (ICDAR) 2015 are sampled from the market, which leads to complicated backgrounds and brings challenges to distinguishing texts from interference regions. Moreover, multi-oriented and multi-scaled instance shapes aggravate the difficulty of text detection. To testify the model performance under a complex environment, we conduct comparison experiments on the ICDAR2015 benchmark. As exhibited in Table VI, our method achieves 86.1% F-measure. Although LeafText is a little lower (0.7% and 0.4%) than TextDCT [49] and LPAP [5] in F-measure, our method exceeds most existing SOTA methods (such as PSE [17], Boundary [44] and ASTD [24]). It is mainly because of LeafText's strong ability to fit various instance shapes and recognize text features. The results in Table VI and Fig. 11 (d) demonstrate our method can recognize the texts with various scales and multi-orientations from the complex background effectively.



(a) False-positive sample (b) Over emitting

Fig. 12. Illustration of some challenging samples. The green bounding boxes are the detection results from our method. The red ones are failed detection regions.

E. Cross Dataset Text Detection

To testify the LeafText's generalization performance on different datasets, we evaluate it through cross-train-test experiments. Specifically, the above four public benchmarks are composed of word-level (Total-Text and ICDAR2015) and line-level (MASRA-TD500 and CTW1500) texts. We conduct cross-train-test experiments on the two types of benchmarks in this section, respectively. As shown in Table VII, on the word-level datasets, our method achieves 84.4% and 75.9% in F-measure when it is trained on ICDAR2015 and Total-Text and is tested on each other. For line-level datasets, LeafText achieves 79.2% and 82.4% in F-measure when it is trained on MSRA-TD500 and CTW1500. The experiments show the LeafText's superior generalization performance.

F. Limitations of Our Algorithm

We have analyzed the upper bound performance of LeafText for fitting arbitrary-shaped text instances and verified the effectiveness of LVT, MOS, and thin vein by the ablation studies in Section IV-C. Meanwhile, the superior detection and generalization performance on multiple benchmarks of our method are demonstrated in Section IV-D and Section IV-E. In this section, we discuss the limitations of our method by visualizing some difficult samples. As depicted in Fig. 12, there are two typical cases. For the false-positive sample (Fig. 12(a)), the high similar vision features between texts and interference regions make it hard to distinguish them effectively. For the case shown in Fig 12(b), there are two adjacent texts and our method over emits into the inner of each other, which brings interference information into detection results and influences the following text recognition task. Therefore, solving the aforementioned limitations that exist in our method will be our future work.

V. Conclusion

In this paper, we explore the leaf vein geometric characteristic and relate it to text contour for designing an effective text representation method (LVT), which improves text fitting ability and avoids disordered point sequence problems naturally. Meanwhile, LVT fining text contour through the thin vein that enjoys half the length of the lateral vein, which reduces the model's complexity and eases the training convergence process while ensuring superior detection performance. Furthermore, considering the lateral and thin veins that are responsible for sampling contour point sequence deeply depending on the main vein, Multi-Oriented Smoother (MOS) enhances the robustness of the main vein, which ensures the correct growth directions of lateral and thin veins effectively. In the end, we successfully accelerate the supervision of lateral and thin vein predictions and balance the importance of texts with different scales through the proposed global incentive loss. Extensive experiments verify the effectiveness of the proposed LVT, MOS, and global incentive loss, and the superiority of the thin vein. Comparisons on the multiple public benchmarks demonstrate the superior detection performance of our approach.

References

[1] Q. Wan, H. Ji, and L. Shen, "Self-attention based text knowledge mining for text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5983–5992.

[2] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2550–2558.

[3] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh, "Learning markov clustering networks for scene text detection," in *CVPR*, 2018, pp. 6936–6944.

[4] 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. [Online]. Available: https://doi.org/10.1109/TIP.2022.3141844

[5] Z. Fu, H. Xie, S. Fang, Y. Wang, M. Xing, and Y. Zhang, "Learning pixel affinity pyramid for arbitrary-shaped text detection," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022.

[6] Y. Cai, Y. Liu, C. Shen, L. Jin, Y. Li, and D. Ergu, "Arbitrarily shaped scene text detection with dynamic convolution," *Pattern Recognition*, vol. 127, p. 108608, 2022.

[7] W. Feng, F. Yin, X. Zhang, and C. Liu, "Semantic-aware video text detection," in *CVPR*, 2021, pp. 1695–1705.

[8] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 4, pp. 1452–1459, 2020.

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6449–6458.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[11] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4159–4167.

[12] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3538–3545.

[13] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7553–7563.

[14] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *AAAI*, 2018, pp. 6773–6780.

[15] 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.

[16] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: Learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.

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

[18] 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 *ICCV*, 2019, pp. 8440–8449.

[19] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Y. Zhibo, T. Lu, and C. Shen, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] 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.

[21] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[22] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 532–548, 2021.

[23] X. Qin, Y. Zhou, D. Yang, and W. Wang, "Curved text detection in natural scene images with semi-and weakly-supervised learning," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 559–564.

[24] P. Dai, Y. Li, H. Zhang, J. Li, and X. Cao, "Accurate scene text detection via scale-aware data augmentation and shape similarity constraint," *IEEE Transactions on Multimedia*, vol. 24, pp. 1883–1895, 2021.

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

[26] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *CVPR*, 2019, pp. 9365–9374.

[27] 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. 9696–9705.

[28] 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.

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

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

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[32] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," *arXiv preprint arXiv:1611.06779*, 2016.

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

[34] 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.

[35] 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.

[36] 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.

[37] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3047–3055.

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

[39] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4940–4949.

[40] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9076–9085.

[41] C. Ma, L. Sun, Z. Zhong, and Q. Huo, "Relatext: exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks," *Pattern Recognition*, vol. 111, p. 107684, 2021.

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

[43] 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.

[44] H. Wang, P. Lu, H. Zhang, M. Yang, X. Bai, Y. Xu, M. He, Y. Wang, and W. Liu, "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 160–12 167.

[45] 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.

[46] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 193–12 202.

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

[48] 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.

[49] Y. Su, Z. Shao, Y. Zhou, F. Meng, H. Zhu, B. Liu, and R. Yao, "Textdct: Arbitrary-shaped text detection via discrete cosine transform mask," *IEEE Transactions on Multimedia*, 2022.

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

[51] 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.

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

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

[54] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision*, pp. 565–571.

[55] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[56] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1083–1090.

[57] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

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

[59] 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.

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

[61] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017, pp. 764–773.