Text Growing on Leaf

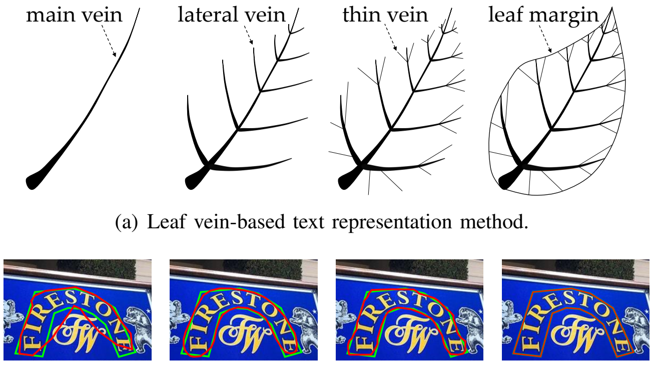
Chuang Yang, Mulin Chen, Yuan Yuan, *Senior Member, IEEE,* and Qi Wang, *Senior Member, IEEE*

***Abstract*—Irregular-shaped texts bring challenges to Scene Text Detection (STD). Although existing regression-based approaches achieve comparable performances, they fail to cover some highly curved ribbon-like text lines. Inspired by morphology, we found that the leaf vein can easily cover various geometries. Specifically, lateral and thin veins are emitted to margin along main vein gradually with the leaf growth. This process can decompose a concave object into consecutive convex regions, which are easier to fit. Hence, the leaf vein is suitable for representing highly curved texts. Considering the aforementioned advantage, we design a leaf vein-based text representation method (LVT), where text contour is treated as leaf margin and represented 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 contours. It grows main veins in Cartesian coordinates to locate texts roughly at first. Then, lateral and thin veins are generated along the main vein growth direction in polar coordinates. They are responsible for generating the coarse contour and refining it, respectively. Meanwhile, Multi-Oriented Smoother (MOS) is designed to smooth the main vein for ensuring reliable growth directions of lateral and thin veins. Additionally, a global incentive loss is proposed to enhance the predictions of lateral and thin veins. Ablation experiments demonstrate LVT can fit irregular-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], [2] is the key technique for intelligent devices to simulate humans recognizing scene text [3], [4], [5], 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 [6], [7]. However, there are many irregular-shaped texts in real scenarios, which brings challenges to traditional approaches. To fit arbitrary-shaped texts effectively, an increasing number of methods [8], [9] are proposed, which can be categorized into segmentation-based methods [10], [11] and regression-based methods [12], [13], [14] roughly.



(a) Leaf vein-based text representation method.

(b) TextRay [15] (c) FCENet [16] (d) TextDCT [2] (e) Ours

Fig. 1. Illustration of leaf vein-based text representation method. (a) shows leaf vein can fit ribbon-like irregular-shaped objects naturally and accurately, where the text contour is treated as leaf margin and represented through main, lateral, and thin veins. (b)–(e) compare the text fitting abilities of TextRay, FCENet, TextDCT, and oaur method, where the ground-truth contours are in green and the reconstructed ones are in red. TextRay fails to model highly-curved texts. FCENet and TextDCT are hard to fit them accurately.

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 and quadrilateral or irregular-shaped contour. These methods have clear drawbacks. Specifically, the one-stage regression-based methods fail to fit highly curved texts because multiple contour points may reside in the same direction or the predicted contour point sequence is disordered. For multiple-stages methods, the intrinsically expensive computational costs lead to low detection efficiency. Therefore, how to design an efficient and effective text representation method is under-explored.

As shown in Fig. 1(a), the leaf is a ribbon-like curved object and enjoys a similar geometry to the irregular-shaped text contour. Meanwhile, the leaf vein can easily cover various geometries. The two superiorities encourage us to explore the growth mode of leaf vein and utilize the leaf vein to represent scene texts. From Fig. 1(a), we observe that lateral

This work was supported by the National Natural Science Foundation of China under Grant U21B2041, 61825603, and 62106182, National Key R&D Program of China 2020YFB2103902.

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.

and thin veins are emitted to margin along the main vein direction gradually with the leaf growth. It splits the ribbon-like irregular-shaped object into a series of regular quadrilateral regions step by step, which makes highly curved texts easier to detect (shown in Fig. 1(d)). Considering the advantages of leaf vein, we combine text geometric characteristics and morphology to design a natural and effective leaf vein-based text representation method (LVT) to fit text instances. We further construct a one-stage text detection framework (called LeafText). It rebuilds text contours by simulating the leaf growth process. Concretely, for one text instance, LeafText first grows the main vein from the predicted kernel mask in Cartesian coordinates to locate the text roughly. Then, in polar coordinates, lateral veins sprout along both sides of the main vein growth direction and thin veins start from lateral veins to margin, where the former is used for determining coarse contour and the latter is responsible for refining the coarse one. In the end, the text contour is drawn by connecting endpoints of lateral and thin veins in a clockwise direction. Particularly, the thin vein sprout dynamically when refining contours, which helps gain accuracy for fitting texts with lower model complexity. Besides, the shorter lengths of thin veins make our model to learn them more easier.

Furthermore, 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 is extracted by the middle sampling method, which is sensitive to the accuracy of the predicted kernel masks. Therefore, we propose Multi-Oriented Smoother (MOS) to ensure the main vein robustness encountering unstable kernel masks. Additionally, text instances enjoy a large range of aspect ratios and scales compared with common objects, which leads to a larger range of lateral and thin vein scales and brings challenges to the training process. Therefore, global incentive loss is proposed to force our model to balance the importance of text instances with different aspect ratios and 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 morphology, a leaf vein-based text representation method (LVT) is proposed. It explores a natural and effective way to represent arbitrary-shaped text instances, which enhances the model's fitting ability effectively.
2) Thin vein is introduced to sprout dynamically to refine contours, which helps gain accuracy for fitting texts with lower model complexity. Meanwhile, the shorter lengths of thin veins make our model to learn them more easier.
3) A Multi-Oriented Smoother (MOS) is designed to smooth the main vein extracted from the predicted unstable kernel mask. It ensures reliable growth directions of lateral and thin veins, which helps generate accurate contour point sequences.
4) Global incentive loss $\mathcal{L}_g$ is proposed to help balance the importance of texts with different aspect ratios and scales, and to force LeafText 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 details of LVT, overall architecture of LeafText, MOS, label generation process and loss functions. The experimental results are discussed in Section IV. Section V concludes the paper.

II. Related Work

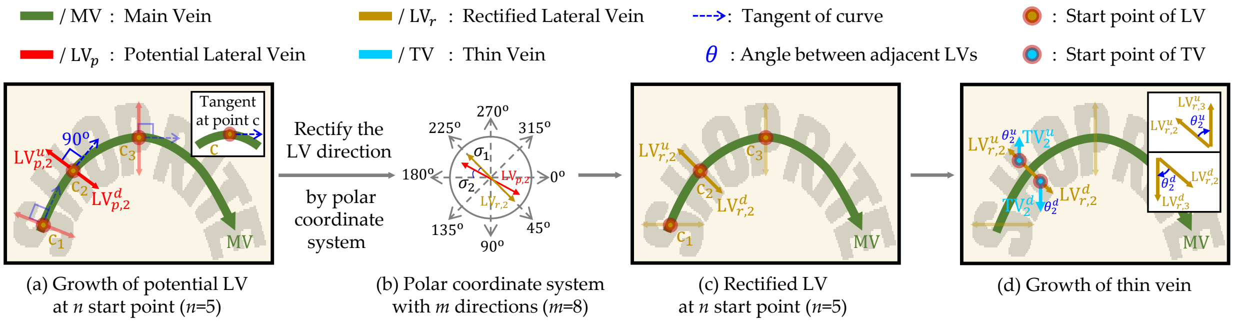
Recently, deep learning has promoted the development of the text detection technique greatly. According to the text representation method, previous text detectors can be classified into segmentation-based methods and regression-based methods roughly. In this section, a review of the existing text detection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

A. Segmentation-Based Methods

Segmentation technology [17] executes pixel-level classification on images, which provides an effective solution for text detection. Zhang *et al.* [18] segmented rough text regions at first. Then, they extracted character components within text blocks by MSER [19]. 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.* [20] 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.* [21] 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 [22], [23] designed similar strategies as [21] to provide solutions for the phenomenon that many texts are very close to each other. Different from the above works, Wang *et al.* [24], [25], [26] and Liao *et al.* [27], [28] proposed expansion strategies to generate text regions from shrink regions, which avoided detecting multiple adhesive texts as one either. The differences between them are that the former expanded shrink to text regions at pixel-level and the latter executed the expansion process at instance-level. The works [29], [30], [1] 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.* [31] considered stack-omnidirectional text dilemma brings much challenges for text detection. They designed LSTM-based module to help generate 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 [32], [33], [34] detected texts in character-level. Baek *et al.* [32] proposed a weakly-supervised framework to generate character-level labels to promote the training process. In the rebuilding process, the approach first predicted character



(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. (a) shows the determining of lateral vein potential direction. (b) and (c) depict the details for rectifying the potential growth direction. (d) visualizes the growth process of the thin vein.

regions and then linked them by affinities to obtain the final detection results. Zhang *et al.* [33] adopted similar strategy as [32] 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.* [34] segmented the center line firstly and then predicted multiple sub-boundaries along with the center line.

B. Regression-Based Methods

Object detection methods [35], [36], [37] adopt contour point sequence-based representation method to rebuild object boundaries, which brings inspiration for the research of text detection. Liao *et al.* [38] inherited the framework of [36] directly to detect horizontal texts. To improve the performance of the multi-oriented text detection, they proposed to predict rotation angles of texts in [39]. Different from the above anchor-based detection framework, Zhou *et al.* [40] introduced the detection strategy proposed in [41] into text detection, which predicted corner points of multi-oriented texts and connected them to obtain the text boxes. Liao *et al.* [42] 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.* [43] 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 [44], [45], [46] 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 [44], Ma *et al.* [47] and Zhang *et al.* [48] 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.* [49] and Wang *et al.* [50] designed two-stage contour point sequence text 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.* [51] 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 [52], Wang *et al.* [15] 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 obtain final detection results.

Some works [53], [2], [16] proposed novel regression strategies to detect text instances and achieved state-of-the-art performance. Specifically, Liu *et al.* [53] introduced Bezier-curve to represent text contours. They explored the probability to fit texts except for standard bounding box detection. Su *et al.* [2] encoded the text regions into compact vectors through discrete cosine transform. Zhu *et al.* [16] 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 at first. Then, we introduce the overall architecture of the proposed LeafText. Next, the details of Multi-Oriented Smoother (MOS) and label generation process 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(a)) 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 and is responsible for providing reliable growth directions of lateral and thin veins. However, the main vein is extracted by the middle sampling method generally, which is sensitive to the accuracy of the predicted kernel mask. Therefore, to ensure the robustness of main vein encountering

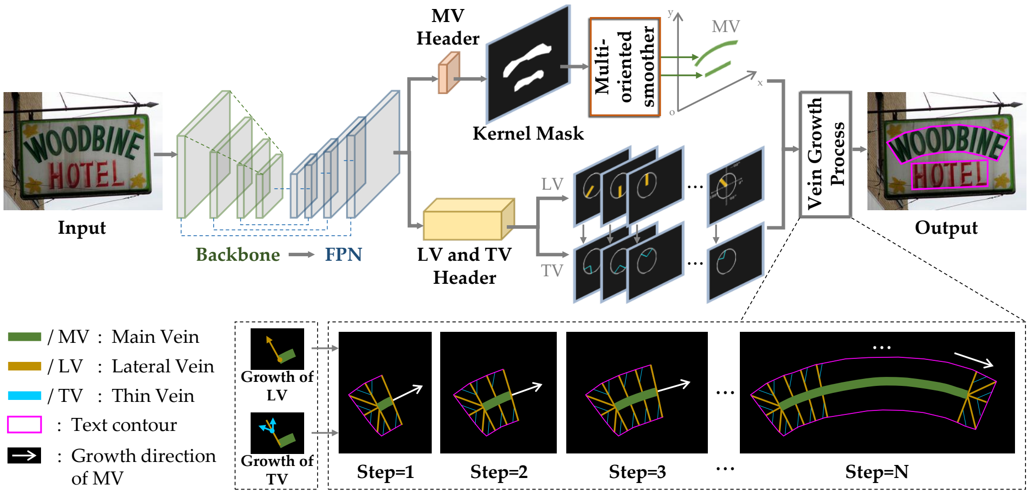


Fig. 3. Overall architecture of the proposed LeafText. It is composed of Backbone, FPN, MV Header, LV-TV header, Multi-Oriented Smoother (MOS), and Vein Growth Process. MV, LV, and TV denote main vein, lateral vein, and thin vein respectively. MOS extracts the main vein from the predicted kernel mask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

unstable kernel masks, it is modeled as polynomial $f(x)$ by MOS (described in Section III-C) and 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 x is the discrete X-axis coordinate of main vein extracted from the predicted kernel mask by middle sampling method. 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 to generate 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 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 text fitting ability (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 computing 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 (refer to III-D).

With the determined lateral veins, the growth directions and lengths of thin veins can be formulated. The thin vein is introduced to refine contours, which grows along adjacent lateral vein directions generally, and will not sprout if the lateral vein parallels adjacent vein directions. It helps LeafText determine dynamically whether contours are needed to be refined by thin veins, which encourages our method to gain accuracy for fitting contours with lower model complexity. Meanwhile, the shorter lengths of thin veins make our model to learn them more easier.

For **the growth directions of thin veins**, as we can see from Fig. 2 (d), the middle points of lateral veins are sampled as the start points of thin veins. Given a lateral vein ($LV_{r,2}^u$) that needs to sprout thin vein (TV_2^u) and two adjacent lateral veins ($LV_{r,1}^u$ and $LV_{r,3}^u$), we can obtain two potential growth directions ($\alpha_{rec,1}^u$ and $\alpha_{rec,3}^u$) of TV_2^u. To ensure a clockwise contour point sequence, we have to determine the final thin vein growth direction $\alpha_{tv,2}^u$ from ($\alpha_{rec,1}^u$ and $\alpha_{rec,3}^u$):

$$\alpha_{tv,2}^u=\max(\alpha_{rec,1}^u, \alpha_{rec,3}^u). \tag{6}$$

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 (refer to III-D). With the determined main veins, lateral veins, and thin veins, the text contour can be drawn by connecting the endpoints of lateral veins and thin veins in the clockwise direction.

B. Overall Pipeline

The overall pipeline of LeafText is shown in Fig. 3, which consists of backbone, FPN, MV header, LV-TV header, Multi-Oriented Smoother (MOS), and Vein Growth Process (refer to Fig. 2). ResNet [54] 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 [55] is used for combining multiple features from the backbone to generate a concatenated feature map $\mathbf{F}_{concat} \in \mathbb{R}^{(h/4) \times (w/4) \times (c \times 4)}$. As described in Section III-A, text contour is represented by the combination of main, lateral, and thin veins. To extract the main vein of text instance, LeafText inputs the $\mathbf{F}_{concat}$ into MV header for generating kernel mask map $\mathbf{F}_k \in \mathbb{R}^{(h/4) \times (w/4) \times 1}$ at first. Then, it extracts the main vein from kernel mask $\mathbf{F}_k$ by MOS. For the growth of lateral and thin veins, LV-TV header conducts the regression task on $\mathbf{F}_{concat}$ 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 $f(x)$ (refer to Equation (1)) and the lengths of lateral and thin veins in all m directions, text contours can be generated by the Vein Growth Process (described in Section III-A and Fig. 2).

C. Multi-Oriented Smoother

As shown in Fig. 2 and Fig. 3, a reliable main vein is important for determining lateral and thin veins, which is the key to rebuilding text contours accurately. However, the main vein extracted by the existing middle sampling method is sensitive to the kernel mask, and the predicted kernel mask always deviates from the ground-truth. It leads to a discrete jagged main vein and bad growth directions for lateral and thin veins, which further results in unreliable rebuilt text contours (as shown in Fig. 7).

Considering the above issue, Multi-Oriented Smoother (MOS) is designed to improve the reliability of main vein encountering the kernel mask that deviates from the ground-truth. It rotates the text to be horizontal with the X-axis and fits the discrete jagged main vein by $f(x)$ (refer to Equation (1)). Specifically, as shown in Algorithm 1 **function** MULTI-ORIENTED SMOOTHER, MOS extracts the kernel mask region $kernel$ from $\mathbf{F}_k$ at first. Meanwhile, considering rotating image leads to information loss of text instances at image borders, MOS pads $kernel$ 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).

Algorithm 1 Growth of Lateral Vein

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

D. Label Generation

As described in Section III-A, the text contour is represented by the combination of main, lateral, and thin veins. In Fig. 3, LeafText predicts kernel masks 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 the kernel mask and lateral and thin vein lengths.

For **the label of kernel mask** (Fig. 4 (b)), the corresponding boundary is generated by shrinking the original text contour through the algorithm proposed in [56]. The inner region of the boundary is regarded as the kernel mask.

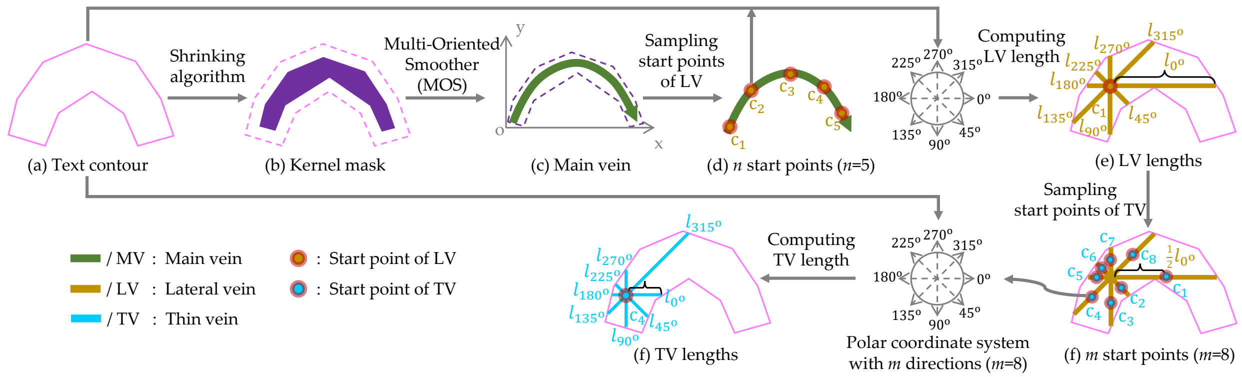

Shrinking algorithm
Multi-Oriented Smoother (MOS)
Sampling start points of LV
Computing LV length
(a) Text contour
(b) Kernel mask
(c) Main vein
(d) n start points (n=5)
(e) LV lengths
Sampling start points of TV
Computing TV length
Polar coordinate system with m directions (m=8)
(f) TV lengths
(f) m start points (m=8)
/ MV : Main vein
/ LV : Lateral vein
/ TV : Thin vein
: Start point of LV
: Start point of TV
c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8
0° 45° 90° 135° 180° 225° 270° 315°
l_{0^o} l_{45^o} l_{90^o} l_{135^o} l_{180^o} l_{225^o} l_{270^o} l_{315^o}
$\frac{1}{2}l_{0^o}$
x
y
o

Fig. 4. Visualization of the label generation process. The kernel mask (b) is used for extracting the main vein. It is the ground-truth of MV header in Fig. 3. For lateral and thin vein lengths (e) and (f) in all directions of the predefined polar coordinate system. The LV-TV header of LeafText is supervised by them in the training process.

For **the label of lateral vein length**, the main vein (Fig. 4 (c)) is extracted from the kernel mask by the function MULTI-ORIENTED SMOOTHER in Algorithm 1 at first. Then, start points and growth directions are determined (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**, the start point is sampled according to the lateral vein, and the length can be computed in the same way as the 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 $\mathcal{L}$ (refer to Equation (7)). 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.

$$\mathcal{L} = \alpha\mathcal{L}_{mv} + \beta\mathcal{L}_{lv-tv}, \tag{7}$$

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 the following experiments.

Optimization of MV header. Dice loss [57] is designed for the segmentation task, where there is a strong imbalance between the positive and negative samples. Considering the regions of kernel mask are much smaller than the background, Dice loss is adopted to evaluate the MV header loss $\mathcal{L}_{mv}$:

$$\mathcal{L}_{mv} = 1 - \frac{2 \times |\mathbf{K}_{pre} \cap \mathbf{K}_{gt}| + \varepsilon}{|\mathbf{K}_{pre}| + |\mathbf{K}_{gt}| + \varepsilon}, \tag{8}$$

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 constructed 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:

$$\mathcal{L}_{lv-tv} = \mathcal{L}_g. \tag{9}$$

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 L1-loss, Smooth-l_1 loss, and L1-loss used in [37], [41] by negative logarithm loss $\mathcal{L}_{nl}$ (as shown in Equation (11)), which scales the differences between predicted lengths and ground-truth into the range of 0–1. It weakens the dominance of large scale texts in gradient optimization process to ensure the effectiveness of our model to both large and small text instances simultaneously.

$$\mathcal{L}_{nl} = -\log\left(\frac{\min(l_{pre}, l_{gt})}{\max(l_{pre}, l_{gt})}\right), \tag{10}$$

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 rectified lateral and thin veins from m directions. It leads to an overwhelming number of indirect samples and a small number of direct samples (rectified lateral and thin veins). To make training more effective and efficient, we propose an incentive strategy for direct samples. It is found that the lengths of direct samples are smaller than indirect ones for a specific dataset. Therefore, an incentive coefficient λ is formulated as follows:

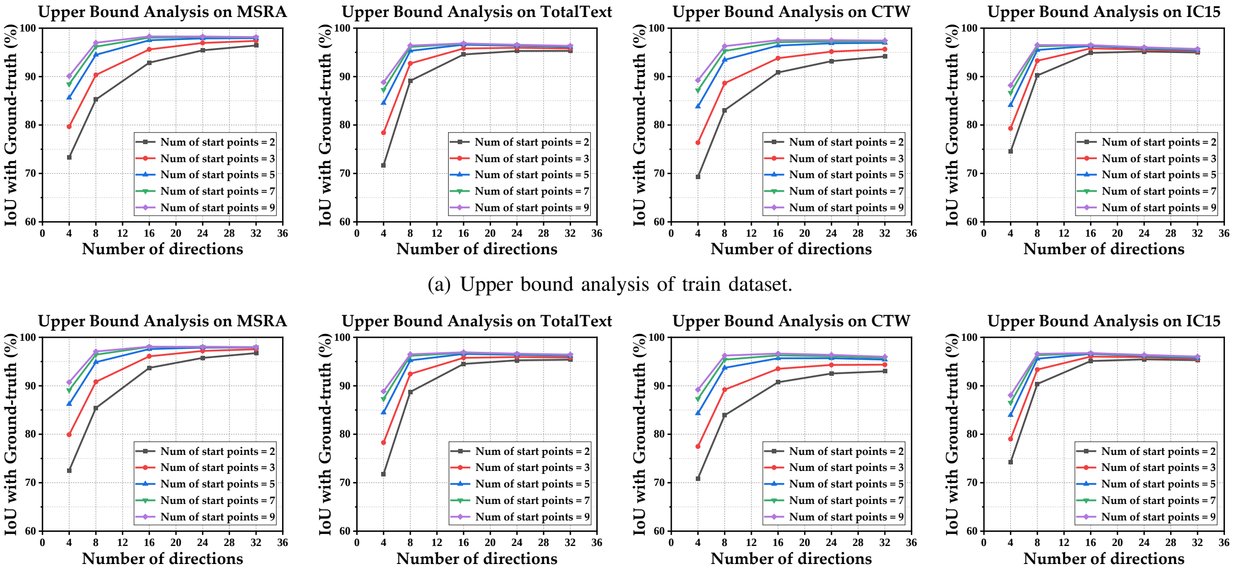
$$\lambda = \tanh(\rho(1 - (l_{gt}/l_s))), \tag{11}$$

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

By combining the Equation (10) and (11), global incentive loss $\mathcal{L}_g$ can be formulated as:

$$\mathcal{L}_g = \frac{1}{T \times M} \sum_{t=1}^{T} \sum_{m=1}^{M} \lambda^{(t,m)} \mathcal{L}_{nl}^{(t,m)}, \tag{12}$$

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



(a) Upper bound analysis of train dataset.

(b) Upper bound analysis of test dataset.

Fig. 5. Upper Bound Analysis. More start points and directions of the lateral vein can model text contours with higher IoU with Ground Truth. ‘Directions’ are the vein directions in the predefined polar coordinate system.

IV. Experiments

To demonstrate the strong ability of LVT to fit arbitrary-shaped texts, we analyze the upper bound of the IoU between the generated label and ground-truth. Meanwhile, the effectiveness of MOS and global incentive loss $\mathcal{L}_g$ are verified. Moreover, LeafText is evaluated on multiple representative public benchmarks to show the superior performance.

A. Datasets

SynthText [58] 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 [59] includes line-level Chinese and English text instances simultaneously. It is composed of 300 training images and 200 testing images, respectively. To ensure a fair comparison environment, 400 images of HUST-TR400 [60] are extra introduced as training data.

Total-Text [61] 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 [62] 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 the model’s strong ability to deal with large scale and ratio objects.

ICDAR2015 [63] is proposed in ICDAR 2015 Robust Reading Competition, which has 1000 training images and 500 testing images. Different from the above three public benchmarks, the background of ICDAR2015 images is more complicated. Meanwhile, the texts enjoy similar basic features with background, which brings challenges for text detection.

B. Implementation Details

The overall pipeline of the proposed LeafText is depicted in Fig. 3. The backbone adopts ResNet [54] directly and the details of FPN can be refer to [55]. MV header and LV-TV header are composed of one 3×3 convolutional layer and m 3×3 convolutional layers, respectively.

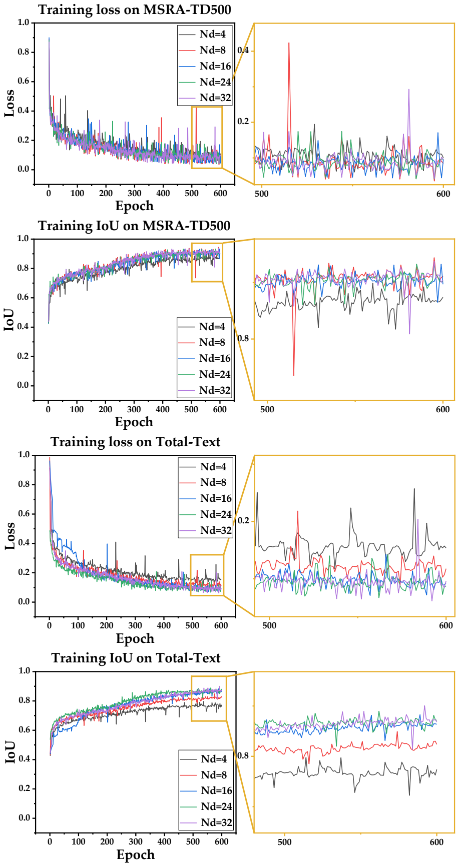
In the pre-processing stage, training samples can be obtained through data augmentation and label generation operators. For the former, it contains the following strategies: (1) random scaling (including image size and aspect); (2) random horizontal flipping; (3) random rotating in the range of (-10°, 10°); (4) random cropping and padding. For the latter, kernel masks and the lengths of lateral and thin veins in m directions are generated by the process in Fig. 4. Different from training samples, testing samples are produced by resizing input RGB images into specific sizes only in the pre-processing stage. Particularly, the texts 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 [64]. For the FPN and headers, they are initialized by the strategy proposed in [65]. To ensure an efficient and effective converge process, Adam [66] 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. In the ablation studies, LeafText is trained on the official datasets directly. All the experiments in this paper are conducted on a workstation with RTX 1080Ti GPU.

In the post-process stage, LeafText grows the main vein starting from the top or left for one text instance. At the start

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. 6. Ablation study for the impact of N_d and N_p on detection performance. N_d indicates the number of the predefined polar coordinate system. N_p means the number of start 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.

and end points of main vein, LeafText rebuilds text contours by the lateral veins in the range of $\{\alpha^d_{rec} - \alpha^u_{rec}\}$ and $\{\alpha^u_{rec} - \alpha^d_{rec}\}$ (refer to Equation (5)) along the clockwise direction, respectively. For the thin vein, it grows along adjacent lateral vein directions generally and will not sprout if the current lateral vein parallels adjacent vein directions.

C. Ablation Study

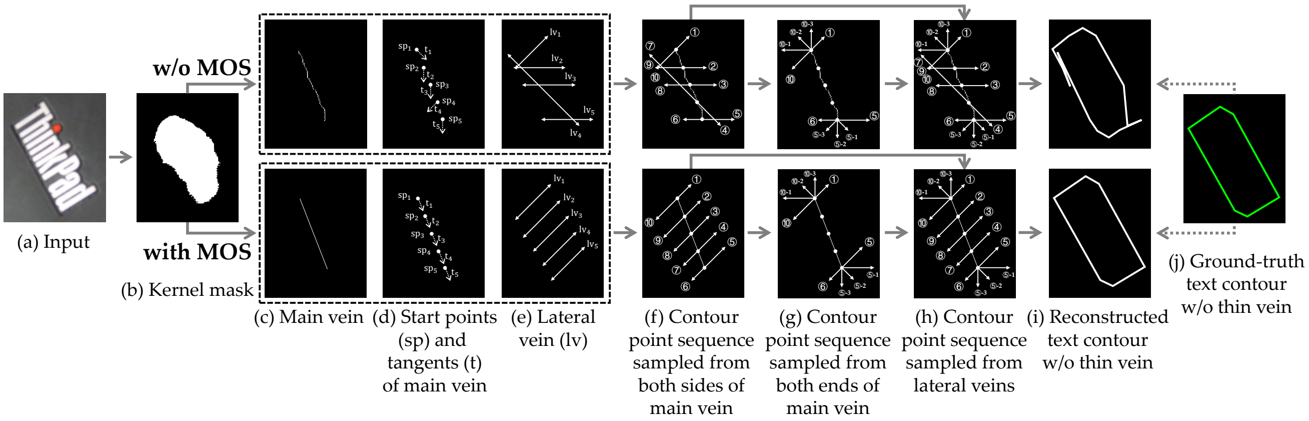
To verify the effectiveness of LeafText, we conduct ablation experiments on multiple public benchmarks in this section. Specifically, to verify the strong fitting ability of LVT, we analyze the upper bound IoU between reconstructed text contour and ground-truth. Meanhiwle, the superiority of the proposed global incentive loss $\mathcal{L}_g$ is demonstrated by comparing it with existing 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 between rebuilt text contour based on generated label and ground-truth. Specifically, as shown in Fig. 5, the IoU can achieve 96% at least on both training and testing samples of four public benchmarks (MSRA-TD500, Total-Text, CTW1500, and ICDAR2015). 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 (N_p and N_d). Concretely, as shown in Fig. 5, the IoU is evaluated when tuning N_p and N_d, respectively. It is found that there is a significant increase for IoU with N_p being tuned from 2 to 5. The upper bound of IoU continues to slow-growing when 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 N_d are visualized in this section. Specifically, larger N_d brings improvements to the fitting ability of our method, which verifies the importance of N_d for leaf vein-based text representation method.

Performance Analysis under Different N_d and N_p. We have analyzed the upper bound IoU of the proposed LVT on different kinds of text instances in Section IV-C. To



(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. 7. 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 I
Detection results of the models equipped with MOS and w/o MOS on MSRA-TD500 and Total-Text datasets.

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

TABLE II
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

further verify the model's performance, LeafText is trained and evaluated under different $\mathrm{N_d}$ and $\mathrm{N_p}$ on MSRA-TD500 and Total-Text text benchmarks.

Specifically, as we can see from the Table (a) in Fig. 6, for multi-oriented texts (MSRA-TD500), LeafText achieves the best performance when $\mathrm{N_d}$ and $\mathrm{N_p}$ are set to 8 and 9, respectively. Meanwhile, the F-measure begins to decrease with the increase of $\mathrm{N_d}$. For irregular-shaped text instances, our method achieves 86.2% in F-measure when $\mathrm{N_d}$ and $\mathrm{N_p}$ are set as 16 and 5, which outperforms the rest of the other models. The above results show the best settings of $\mathrm{N_d}$ and $\mathrm{N_p}$ to detect multi-oriented and irregular-shaped texts. Furthermore, we visualize the details of the training process in Fig. 6 (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 $\mathrm{N_d}$ equals 8, which matches the results of Table (a) in Fig. 6. Meanwhile, the curves of the training IoU and loss on Total-Text show the unsatisfied convergence process when $\mathrm{N_d}$ is set to 4 and 8, which verifies the effectiveness of large $\mathrm{N_d}$ 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 the 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 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. 7. Concretely, given a predicted kernel mask (Fig. 7 (b)), MOS helps our method to determine correct tangent directions on each start point (Fig. 7 (d)), which helps avoid disordered contour point sequence (Fig. 7 (f)) and improve the reliability of reconstructed text contour (Fig. 7 (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

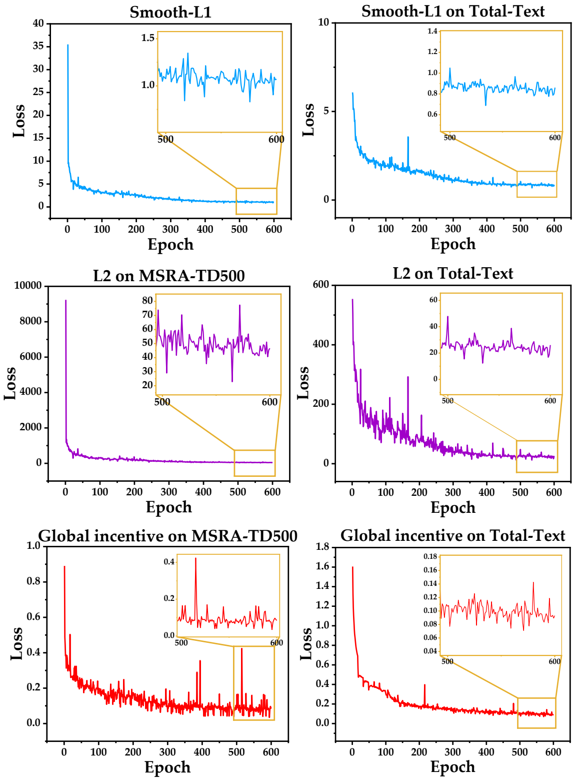


Fig. 8. Visualization of the training processes on the MSRA-TD500 and Total-Text with different loss functions.

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

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. 8. 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. Benefiting from the advantage that the thin vein length is shorter than the lateral vein, the thin vein eases the learning of 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.

TABLE IV
PERFORMANCE COMPARISON WITH RELATED METHODS ON MSRA-TD500 DATASET. **RED**, **GREEN**, AND **BLUE** ARE TOP THREE BEST RESULTS. "EXT." DENOTES EXTRA TRAINING DATA.

Methods	Backbone	Ext.	P	R	F
MOTD [18] (CVPR'16)	VGG16	✓	83.0	67.0	74.0
EAST [40] (CVPR'17)	PVANET2x	×	87.3	67.4	76.1
SegLink [6] (CVPR'17)	VGG16	✓	86.0	70.0	77.0
PixelLink [21] (AAAI'18)	VGG16	×	83.0	73.2	77.8
TextSnake [34] (ECCV'18)	VGG16	✓	83.2	73.9	78.3
RRD [42] (CVPR'18)	VGG16	✓	87.0	73.0	79.0
CornerNet [20] (CVPR'18)	VGG16	✓	87.6	76.2	81.5
CRAFT [32] (CVPR'19)	VGG16	✓	88.2	78.2	82.9
TextField [23] (TIP'19)	VGG16	✓	87.4	75.9	81.3
SAE [22] (CVPR'19)	ResNet50	✓	84.2	81.7	82.9
ATRR [14] (CVPR'19)	SE-VGG16	×	85.2	82.1	83.6
PAN [25] (ICCV'19)	ResNet18	✓	84.4	83.8	84.1
DB [27] (AAAI'20)	ResNet18	✓	90.4	76.3	82.8
DRRG [33] (CVPR'20)	VGG16	✓	88.1	82.3	85.1
OPMP [31] (TMM'21)	–	✓	86.0	83.4	84.7
PAN++ [26] (TPAMI'21)	ResNet18	✓	85.3	84.0	84.7
SAVTD [12] (CVPR'21)	ResNet50	×	89.2	81.5	85.2
GV [13] (TPAMI'21)	ResNet101	✓	88.8	84.3	86.5
ReLaText [47] (PR'21)	ResNet50	✓	90.5	83.2	**86.7**
LPAP [11] (TOMM'22)	ResNet50	✓	87.9	77.7	82.5
DC [9] (PR'22)	–	✓	87.9	83.1	85.4
DB++ [28] (TPAMI'22)	ResNet18	✓	87.9	82.5	85.1
DB++ [28] (TPAMI'22)	ResNet50	✓	91.5	83.3	**87.2**
Ours (736)	ResNet50	✓	92.1	83.8	**87.8**

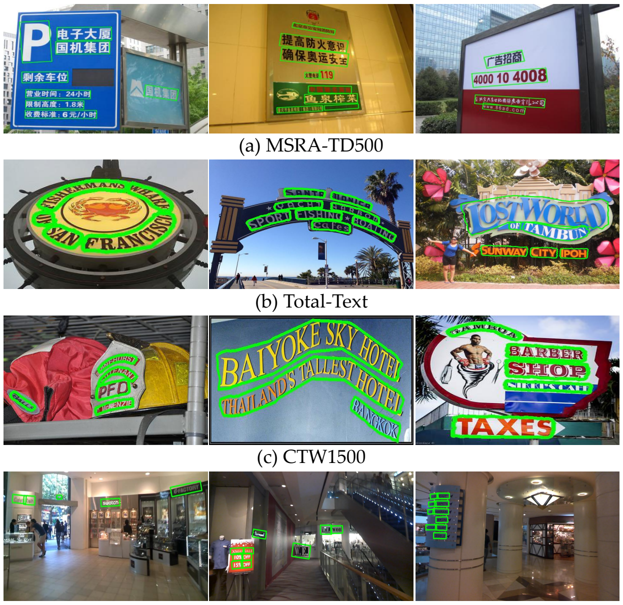
D. Comparison with State-of-the-Art Methods

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 [47], GV [13], and DC [9] achieve 86.7%, 86.5%, and 85.4% in F-measure. Benefiting from the strong connection ability of Graph Convolutional Network (GCN),

TABLE V
PERFORMANCE COMPARISON WITH RELATED METHODS ON TOTAL-TEXT AND CTW1500 DATASETS. RED, GREEN, AND BLUE ARE TOP THREE BEST RESULTS. "EXT." DENOTES EXTRA TRAINING DATA.

Methods	Backbone	Ext.	Total-Text			CTW1500		
			Precision	Recall	F-measure	Precision	Recall	F-measure
TextSnake [34] (ECCV'18)	VGG16	✓	82.7	74.5	78.4	67.9	85.3	75.6
ATRR [14] (CVPR'19)	SE-VGG16	×	80.9	76.2	78.5	80.1	80.2	80.1
CRAFT [32] (CVPR'19)	VGG16	✓	87.6	79.9	83.6	86.0	81.1	83.5
CTD [30] (ICDAR'19)	ResNet50	×	80.6	82.3	81.4	79.9	77.0	78.5
LOMO [49] (CVPR'19)	ResNet50	✓	87.6	79.3	83.3	85.7	76.5	80.8
PSE [24] (CVPR'19)	ResNet50	✓	84.0	78.0	80.9	84.8	79.7	82.2
SegLink++ [44] (PR'19)	VGG16	✓	82.1	80.9	81.5	82.8	79.8	81.3
TextDragon [46] (ICCV'19)	VGG16	✓	85.6	75.7	80.3	84.5	82.8	83.6
Boundary [50] (AAAI'20)	ResNet50	✓	85.2	83.5	84.3	–	–	–
ContourNet [51] (CVPR'20)	ResNet50	×	86.9	83.9	85.4	83.7	84.1	83.9
TextRay [15] (ACMMM'20)	ResNet50	×	83.5	77.9	80.6	82.8	80.4	81.6
Spotter [29] (TPAMI'21)	ResNet50	✓	88.3	82.4	85.2	–	–	–
FCENet [16] (CVPR'21)	ResNet50	×	87.4	79.8	83.4	85.4	80.7	83.1
PSE+STKM [8] (CVPR'21)	ResNet18	✓	86.3	78.4	82.2	85.1	78.2	81.5
OPMP [31] (TMM'21)	–	✓	88.5	82.9	85.6	85.1	80.8	82.9
ASTD [1] (TMM'22)	ResNet101	✓	85.4	81.2	83.2	86.2	80.4	83.2
TextDCT [2] (TMM'22)	ResNet50	✓	87.2	82.7	84.9	85.0	85.3	**85.1**
LPAP [11] (TOMM'22)	ResNet50	✓	87.3	79.8	83.4	84.6	80.3	82.4
DC [9] (PR'22)	–	✓	90.5	82.7	**86.4**	86.9	82.7	84.7
DB++ [28] (TPAMI'22)	ResNet18	✓	88.9	83.2	**86.0**	87.9	82.8	**85.3**
Ours (640)	ResNet18	✓	90.8	84.0	**87.3**	87.1	83.9	**85.5**


(a) MSRA-TD500

(b) Total-Text

(c) CTW1500

(d) ICDAR2015

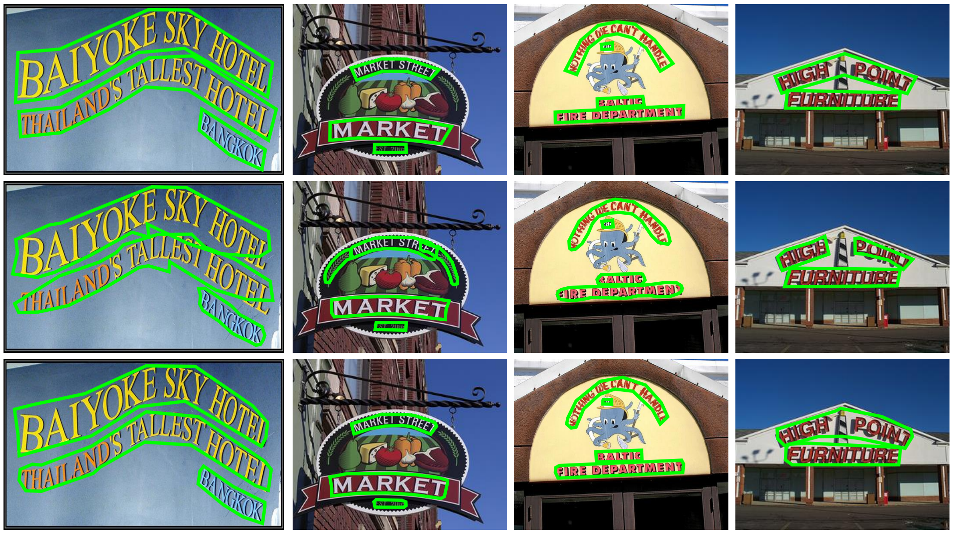
Fig. 9. Visualization of some qualitative results on MSRA-TD500, Total-Text, CTW1500, and ICDAR2015 datasets.

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++ [28], though it achieves significant improvement by embedding DConv [67] into the corresponding backbone, our method still outperforms it with basic network. We show some qualitative results on MSRA-TD500 in Fig. 9 (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. 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 to evaluate the model performance with ResNet-18.

As we can see from Table V, for the detection of word-level text instances in Total-Text, DC [9] and DB++ [28] 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 [9] 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++ [28] and TextDCT [2] 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++ [28] 0.2% even it is equipped with DConv [67] and 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



(a) vs. TextRay [15] (b) vs. FCENet [16] (c) vs. TextDCT [2] (d) vs. LPAP [11]

Fig. 10. Qualitative comparisons with TextRay [15], FCENet [16], TextDCT [2], and LPAP [11] on selected challenging samples. The first row visualizes ground-truth. The second row shows the detection results of TextRay, FCENet, TextDCT, and LPAP. The last row shows the detection results of ours.

TABLE VI
PERFORMANCE COMPARISON WITH RELATED METHODS ON ICDAR2015 DATASET. RED, GREEN, AND BLUE ARE TOP THREE BEST RESULTS. "EXT." DENOTES EXTRA TRAINING DATA.

Methods	Backbone	Ext.	P	R	F
WordSup [45] (ICCV'17)	VGG16	✓	79.3	77.0	78.2
MCN [7] (CVPR'18)	VGG16	✓	72.0	80.0	76.0
PixelLink [21] (AAAI'18)	VGG16	×	85.5	82.0	83.7
TextBoxes++ [39] (TIP'18)	VGG16	✓	87.8	78.5	82.9
PSE [24] (CVPR'19)	ResNet50	✓	86.9	84.5	85.7
RRD [42] (CVPR'19)	VGG16	✓	88.0	80.0	83.8
SegLink++ [44] (PR'19)	VGG16	✓	83.7	80.3	82.0
Boundary [50] (AAAI'20)	ResNet50	✓	88.1	82.2	85.0
FCENet [16] (CVPR'21)	DCN-ResNet50	×	85.1	84.2	84.6
Spotter [29] (TPAMI'21)	ResNet50	✓	85.8	81.2	83.4
PAN++ [26] (TPAMI'21)	ResNet18	✓	85.9	80.4	83.1
EAST+STKM [8] (CVPR'21)	ResNet18	✓	88.7	84.9	86.8
PSE+STKM [8] (CVPR'21)	ResNet18	✓	87.8	84.1	85.9
ASTD [1] (TMM'22)	ResNet101	✓	87.2	81.3	84.1
TextDCT [2] (TMM'22)	ResNet50	✓	88.9	84.8	86.8
LPAP [11] (TOMM'22)	ResNet50	✓	88.7	84.4	86.5
Ours (1152)	ResNet50	✓	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.

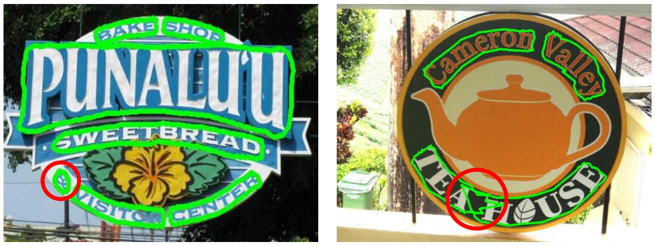
Type	Methods	Training	Testing	P	R	F
word-level	TextField [23]	IC15	TT	61.5	65.2	63.3
	CM-Net [10]			75.8	64.5	69.7
	Res18-Pre-Ours			89.2	80.0	84.4
	TextField [23]	TT	IC15	77.1	66.0	71.1
	CM-Net [10]			76.5	68.1	72.1
	Res18-Pre-Ours			83.0	69.9	75.9
line-level	TextField [23]	MSRA	CTW	75.3	70.0	72.6
	CM-Net [10]			77.2	69.7	72.8
	Res18-Pre-Ours			83.8	75.0	79.2
	TextField [23]	CTW	MSRA	85.3	75.8	80.3
	CM-Net [10]			85.8	77.1	81.2
	Res18-Pre-Ours			82.9	82.0	82.4

results on Total-Text and CTW1500 are depicted in Fig. 9 (b) and (c) to further demonstrate the effectiveness of LeafText. Furthermore, we visualize some comparison results in Fig. 10 to show the superiority of our method.

Evaluation on ICDAR2015. The images in this dataset are sampled from the market, which leads to complicated backgrounds and brings challenges for text detection. 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 [2] and LPAP [11] in F-measure, our method exceeds most existing SOTA methods (such as PSE [24], Boundary [50] and ASTD [1]). 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. 9 (d) demonstrate our method can recognize the texts with various scales and multi-orientations from the complex background effectively.

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



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

Fig. 11. Illustration of some challenging samples. The green bounding boxes are the results from our method. The red ones are failed detection regions.

is trained on MSRA-TD500 and CTW1500. The experiments show 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 performance on multiple benchmarks of our method is 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. 11, there are two typical cases. For the false-positive sample (Fig. 11(a)), the highly similar vision features between texts and interference regions make it hard to distinguish them effectively. For the case shown in Fig. 11(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 growth mode and relate it to text contour for designing an effective text representation method (LVT), which improves the model's ability to fit highly curved texts naturally and effectively. Particularly, the thin vein helps LeafText cover text contours with lower model complexity and the shorter length of it eases the training convergence process while ensuring superior detection performance. Furthermore, considering the deep dependencies of lateral and thin veins on the main vein, Multi-Oriented Smoother (MOS) is designed to ensure the reliability of the main vein encountering the unstable kernel mask. In the end, we successfully accelerate the lateral and thin vein predictions and balance the importance of texts with different scales through global incentive loss. Extensive experiments verify the effectiveness of LVT, MOS, and global incentive loss. Comparisons on the multiple public benchmarks demonstrate the superior detection performance of our approach.

References

[1] 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 Trans. Multimedia*, vol. 24, pp. 1883–1895, 2021.

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

[3] L. Wu, Y. Xu, J. Hou, C. P. Chen, and C.-L. Liu, "A two-level rectification attention network for scene text recognition," *IEEE Trans. Multimedia*, 2022.

[4] D. Peng, L. Jin, W. Ma, C. Xie, H. Zhang, S. Zhu, and J. Li, "Recognition of handwritten chinese text by segmentation: A segment-annotation-free approach," *IEEE Trans. Multimedia*, 2022.

[5] M. Li, B. Fu, Z. Zhang, and Y. Qiao, "Character-aware sampling and rectification for scene text recognition," *IEEE Trans. Multimedia*, 2021.

[6] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *CVPR*, 2017, pp. 2550–2558.

[7] 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.

[8] Q. Wan, H. Ji, and L. Shen, "Self-attention based text knowledge mining for text detection," in *CVPR*, 2021, pp. 5983–5992.

[9] 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.

[10] 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.

[11] Z. Fu, H. Xie, S. Fang, Y. Wang, M. Xing, and Y. Zhang, "Learning pixel affinity pyramid for arbitrary-shaped text detection," *TOMM*, 2022.

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

[13] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, 2020.

[14] 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.

[15] 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.

[16] 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.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[18] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *CVPR*, 2016, pp. 4159–4167.

[19] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *CVPR*, 2012, pp. 3538–3545.

[20] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *CVPR*, 2018, pp. 7553–7563.

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

[22] 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.

[23] 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.

[24] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *CVPR*, 2019, pp. 9336–9345.

[25] 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.

[26] 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 Trans. Pattern Anal. Mach. Intell.*, 2021.

[27] 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.

[28] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[29] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 532–548, 2021.

[30] X. Qin, Y. Zhou, D. Yang, and W. Wang, "Curved text detection in natural scene images with semi-and weakly-supervised learning," in *ICDAR*, 2019, pp. 559–564.

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

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

[33] 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.

[34] 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.

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

[36] 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.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *ICCV*, 2017, pp. 3047–3055.

[44] 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.

[45] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," in *ICCV*, 2017, pp. 4940–4949.

[46] 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 *ICCV*, 2019, pp. 9076–9085.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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 *CVPR*, 2020, pp. 12 193–12 202.

[53] 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.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[55] 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.

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

[57] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV*, pp. 565–571.

[58] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *CVPR*, 2016, pp. 2315–2324.

[59] 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.

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

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

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

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

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

Mulin Chen received the B.E. degree in software engineering and the Ph.D. degree in computer application technology from Northwestern Polytechnical University, Xi'an, China, in 2014 and 2019 respectively. He is currently a researcher with the School of Artificial Intelligence, OPtics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His current research interests include computer vision and machine learning.

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

Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Artificial Intelligence, OPtics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, pattern recognition and remote sensing.