Computers and Electronics in Agriculture 203 (2022) 107352

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Leaf vein segmentation with self-supervision

Lei Li [a], Wenzheng Hu [b,d], Jiang Lu [c], Changshui Zhang [a,*]

[a] Institute for Artificial Intelligence, Tsinghua University (THUAI), Beijing National Research Center for Information Science and Technology (BNRist), State Key Lab of Intelligent Technologies and Systems, Department of Automation, Tsinghua University, Beijing, PR China
[b] The State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing, PR China
[c] China Marine Development and Research Center (CMDRC), Beijing, PR China
[d] Kuaishou Technology Co., Ltd., PR China

ARTICLE INFO

Keywords:
Leaf vein segmentation
Computer vision
Self-supervision
Encoder–Decoder

ABSTRACT

The leaf vein, often viewed as the fingerprint of the leaf, is an important characteristic used to identify plant species. Leaf vein segmentation aims to extract the vein features and obtain the vein architectures from leaf images. Unlike the general semantic or instance segmentation focusing on the block and object level, the leaf vein segmentation is fine and it focuses on the internal details inside the mesophyll, whose color is indistinguishable, making this task difficult. To tackle this problem, we utilize the particularities of leaf veins, namely continuity and branching, and propose a **Co**nfidence **R**efining **Ve**in Network (CoRE-Net) to segment leaf veins by handling the intersections, breakpoints, and blurred boundaries of veins to enhance segment prediction. Moreover, the proposed network only needs a few labeled samples to warm start, and then the whole network can converge without any annotated labels. Meanwhile, we collect and release the first leaf vein dataset, *Leaf Vein Dataset 2021* (LVD2021). The proposed framework achieves mean Intersection over Union at 71.02% and mean Dice at 79.76% on LVD2021, which outperforms its counterparts in different settings, demonstrating the effectiveness of our framework.

1. Introduction

The veins are responsible for the mechanical support of leaves and the long-distance transport of water and nutrients (Carvalho et al., 2018; Lalonde et al., 2003). Meanwhile, they are also the main morphological characteristics of leaves, containing plants' intrinsic attributes and crucial genetic information. Many botanists have analyzed and evaluated traits of veins or leaf venation networks, including total vein length, vein density, piecewise vein lengths and widths, areole area, and skeleton graph statistics. In botany, precise measurements of the vein architecture play an important role in plant phenotyping for ecological and genetic research (Sack and Frole, 2006; Boyce et al., 2009), help recognize the plant species, help learn leaf carbon, water fluxes, and help monitor the environment e.t.c. Therefore, it is significant to extract the veins.

The traditional approaches to obtain the vein architecture are through manually chemical reagent processing, high-resolution scanner, or X-ray (Larese et al., 2014). These processes are not convenient and sometimes time-consuming. Then many image processing techniques are involved in extracting the features from the plants. However, there are few valuable works on particular leaf veins. Chakkaravarthy et al. (2016) proposed a system using Hough lines to yield vein features from the leaf images. Radha and Jeyalakshmi (2014) proposed an algorithm using the Canny edge detection method to extract leaf vein. Selda et al. (2017) and Larese and Granitto (2016) used scale-invariant feature transform (SIFT) for describing and detecting the local features of the leaf vein images. These algorithms are effective in specific environments, but many factors affect leaf images, e.g., light, plant types, and colors. Thus, there is a large gap between the extraction results and the expected output. Leaf vein feature extraction algorithms (Jeon and Rhee, 2017; Tan et al., 2018) based on neural networks avoid using handcrafted feature extractors but fell into the dilemma of rigidity and annotation burden. Therefore, most image-based methods rely on specialized handcrafted extractors and only work under an ideal experimental environment. With the recent development of deep learning, neural network models were applied to image segmentation and have shown promising potential to solve this problem.

Researchers have explored image segmentation in many different fields, such as medical image (Fu et al., 2018), traffic scene (Yan et al., 2020), facial recognition (Lin et al., 2020), e.t.c. Different from

* Corresponding author.
E-mail addresses: lilei17@mails.tsinghua.edu.cn (L. Li), huwenzheng@kuaishou.com (W. Hu), luj13@tsinghua.org.cn (J. Lu), zcs@mail.tsinghua.edu.cn (C. Zhang).

https://doi.org/10.1016/j.compag.2022.107352
Received 11 September 2021; Received in revised form 30 July 2022; Accepted 25 August 2022
Available online 2 November 2022

these semantic (Chen et al., 2014) or instance segmentation (Liu et al., 2018) focusing on block/object level, the leaf vein segmentation is fine and focuses on the internal details. Semantic segmentation is the task of associating each pixel of an image with a semantic class label. Sometimes, it is also regarded as combining the semantic feature extraction task and the pixel-wise classification task. Fueled by recent advances in the research of deep learning, deep convolutional neural networks (CNNs) (He et al., 2016) demonstrate the efficiency in both feature extraction and image classification. Fully Convolutional Networks (FCNs) (Long et al., 2015) which often employ a CNN pretrained as the backbone is proposed in many semantic segmentation tasks (Chen et al., 2014, 2017). With the emerging of the end-to-end FCN Ronneberger et al. (2015) proposed U-shape Net (U-Net) framework for biomedical image segmentation. U-Net has shown promising results on neuronal structures segmentation in electron microscopic recordings and cell segmentation in light microscopic images. It becomes a popular architecture for biomedical image segmentation tasks (Fu et al., 2018). DeepLabV3 (Chen et al., 2017) adopts several parallel atrous convolution with different rates to control the receptive fields of the feature maps. EncNet (Zhang et al., 2018) introduces a channel attention mechanism to capture the global context and utilizes global pooling to collect image-level context information. DANet (Fu et al., 2019) utilizes Non-local module to capture the contextual information, while GCNet (Cao et al., 2019) uses a global context block to efficiently model the global context. UperNet (Xiao et al., 2018) also employs the encoder–decoder structure and HRNet (Sun et al., 2019) is able to learn high-resolution representations throughout feature extraction. Gu et al. (2019) proposed a context encoder network (CE-Net), integrating a dense atrous convolution (DAC) block and a residual multi-kernel pooling (RMP) block with the backbone U-Net structure to capture more high-level features and preserve more spatial information. However, similar to other data-driven deep learning methods, these semantic segmentation models usually require large amounts of annotated data. Moreover, leaf veins are often located in the mesophyll, while the color-indistinguishable mesophyll makes vein segmentation challenging. Xu et al. (2020) is the only one that moves one step on the vein segmentation, while it only explores local images and requires large amounts of laboriously annotated images for supervised learning.

A promising method to dramatically reduce the cost of annotations is pseudo-labeling. Pseudo-labeling (Lee, 2013; Iscen et al., 2019; Shi et al., 2018; Han et al., 2019) belongs to the self-supervised learning scenario (Nguyen et al., 2019; Patro et al., 2021), and it is often used in semi-supervised learning. At the beginning of the self-supervised learning procedure, a supervised model is trained on only a few labeled data, and the resulting model is used to attain pseudo labels for the large number of unlabeled data. Then the model is re-trained on both the original labeled data and the newly attained pseudo-labeled data. Lee (2013) attempted to use the current network's predictions as pseudo labels of the unlabeled samples. Shi et al. (2018) proposed to add contrastive loss to consistency loss so as to force the network to have more confident predictions, and use the class predictions as the pseudo labels of the unlabeled samples. Finally, Iscen et al. (2019) proposed a framework for pseudo-labeling with graph-based method and refined hard pseudo-labels by label propagation. They deal with the unequal confidence in predictions and class-imbalance by adding an uncertainty score for every sample and class, respectively.

The main objective of this research was to develop a model which is able to get the precise leaf vein segmentation results from the input images and trained by a few labeled samples. The model needs to have the ability to learn from unlabeled data and considers the leaf veins' particularities. To our best knowledge, no one explicitly explores the veins' particularities in leaf vein segmentation, and no one utilize self-supervised learning in this task. Therefore, the objectives of this research were to (i) precisely predict the leaf veins directly from the images with a deep learning model; (ii) properly utilize the veins' particularities; (iii) train the model in a self-supervision manner with only a few labeled data.

In this paper, we use prior knowledge about the vein, take advantage of veins' particularities, namely continuity and branching, and propose a point refiner to handle the intersections, breakpoints, and blurred boundaries to enhance segment prediction. Then we propose a novel and specific two-phased self-supervised framework, Confidence Refining Vein Network (CoRE-Net), for the leaf vein segmentation. It contains supervised warm-start training and iteratively self-supervised training. Unlike most existing segmentations, CoRE-Net firstly takes advantage of veins' particularities, namely continuity and branching, and then can efficiently segment veins with only a few annotated images. To demonstrate the practicability and effectiveness of the CoRE-Net, we further collect and release the first pixel-wise annotation vein dataset, *Leaf Vein Dataset 2021* (LVD2021), which consists of 4977 images with 36 different classes. Experiments on LVD2021 demonstrate that the proposed CoRE-Net outperforms other state-of-the-art methods in different settings. We summarize our contributions as follows:

- To the best of our knowledge, the proposed CoRE-Net is the first to explore and utilize the vein's continuity and branching for the vein segmentation.
- The proposed framework is the first iteratively self-supervised segmentation framework that converges with only a few labeled samples.
- We collect and release the first pixel-wise annotation vein dataset, namely LVD2021 and build a benchmark for subsequent works.

2. Materials and methods

2.1. The Leaf Vein Dataset 2021

For now, there is no public dataset for vein segmentation. Therefore, we collect and release the first leaf vein dataset, the Leaf Vein Dataset 2021(LVD2021). LVD2021 contains 4977 high-resolution (2736 pix × 3648 pix) images with 36 kinds of leaves, and each type has more than 100 images with pixel-level annotations. Table 1 summarizes the details, and Fig. 1 displays each kind of leaves.

2.1.1. Data collection

Leaves are collected from healthy plants in the wild. They distribute in about six provinces in China, including Shanxi, Sichuan, Zhejiang, Shandong, Shanghai, and Hongkong. We collect images of these leaves on white paper indoor under natural light. For convenience and practicability, we utilize the smartphone's camera with automatic mode, e.g., iPhone 8 and Huawei Honor V30 pro. The camera is about 15 cm away from leaves.

2.1.2. Data annotation

Images are annotated manually by some annotators with basic botany knowledge, and all annotators use the same software Photoshop and pen tablets device (Wacom CTL-427). Each image has three annotations: contour edge annotation, coverage annotation, and the main vein annotation. The contour edge is the outline of the leaf, which is annotated via a quick detection tool in Photoshop and then careful manual calibration. The coverage annotation is the mask of the leaf based on the contour edge. The main vein annotation focus on the primary vein, the secondary vein, and the tertiary vein with more than 20 pixels. Fig. 2 displays two examples about the annotations.

Table 1
The composition of Leaf Vein Dataset 2021 (LVD2021). Each type of leaf is more than 100, and we offer the pixel-level annotation.

Category	Image quantity	Category	Image quantity	Category	Image quantity
Walnut	165	Grape	102	Thistle	131
Smoke Tree	131	Hibiscus	143	Mirabilis Jalapa	170
Poplar	201	Morning Glory	120	Sycamores	101
Oriental Cherry	146	Apricot	113	Lilac	134
Chinese Redbud	125	Chenopodium Album	129	Persimmon	106
Crape Myrtle	150	Phlox Paniculata	133	Mulberry	123
Hackberry	147	Callistephus Chinensis	134	Sichuan Pepper	113
Crataegus Pinnatifida	152	Maple Tree	136	Vitex Negundo Var	135
Virginia Creeper	147	Amaranth	140	Magnolia Denudata	148
Forsythia Suspensa	153	Honeysuckle	107	Chinese Rose	119
Fructus Xanthii	140	Sweet Potato	134	Elm	122
Cynanchum	114	Cedar	121	Holly	292

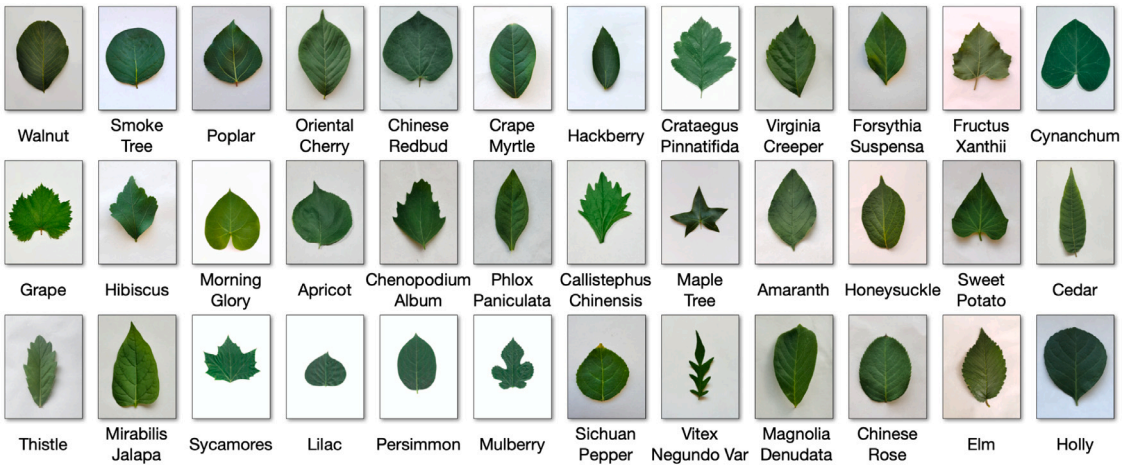

Fig. 1. Samples selected from our collected dataset LVD2021. Each type of leaf has a different outline and vein shape.

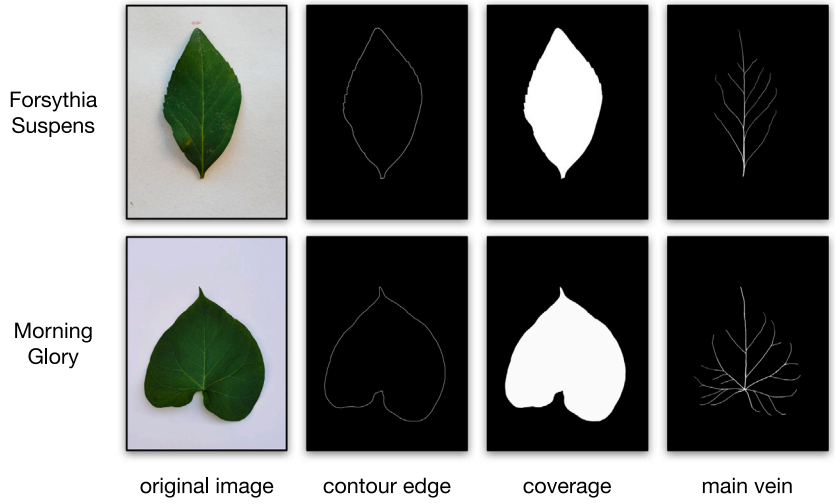

Fig. 2. Some examples of the LVD2021 dataset, taking *Forsythia Suspens* and *Morning Glory* for example. The images in first column are the original leaf images. The images in second column shows the contour edges of the leaves. The white area of the images in third column cover the whole leaves. The fourth column reveal the various characterization for main veins.

2.2. CoRE-Net

Most existing semantic or instance segmentations focus on the block and object level. However, the leaf vein segmentation is more refined. It focuses on the internal details inside the mesophyll, whose color is indistinguishable. Therefore, it is challenging to segment veins. Besides, most existing segmentation frameworks need thousands of labeled samples, leading to a time-consuming annotation. Here, we propose a **Co**nfidence **R**efining V**e**in Network (CoRE-Net) for the leaf vein segmentation, which utilizes the continuity and branching of veins and converges by self-supervised with only a few labeled samples and with lots of unlabeled samples. The CoRE-Net mainly contains three parts: encoder, decoder, and point refiner. Fig. 3 illustrates the overview of the CoRE-Net.

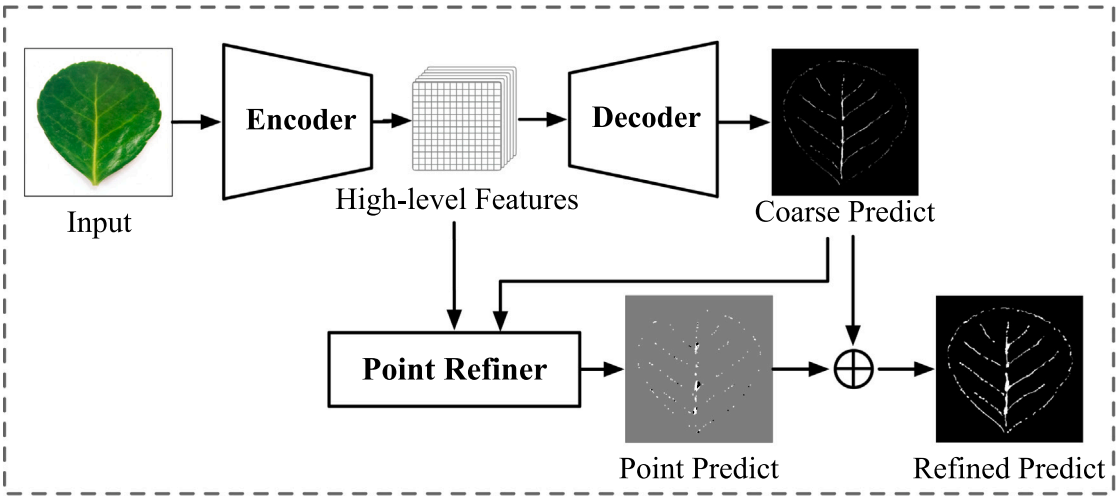

Fig. 3. Overview of the proposed iterative self-learning framework CoRE-Net. The current input image is fed into a encoder to get the high-level semantic feature maps, and the extracted features are fed into a decoder to obtain the coarse predict $\boldsymbol{Y}_{coarse}$. The point refiner select key points to get more accurate refined point predict $\boldsymbol{Y}_{refined}$. Finally, the two predicts are integrated to the final refined predict $\boldsymbol{Y}_u$. In the warm-start training phase, we train the model in a supervision with a few labeled samples to obtain a good initialization of the weights. In the self-supervision training phase, this refined predict will be used as pseudo label to train the model iteratively.

Notation. $\mathcal{D}_l = \{\boldsymbol{X}_l, \boldsymbol{Y}_l\} = \{(\boldsymbol{X}_{l_1}, \boldsymbol{Y}_{l_1}), \ldots, (\boldsymbol{X}_{l_N}, \boldsymbol{Y}_{l_N})\}$ is the supervised dataset with N annotated samples, and $\mathcal{D}_u = \{\boldsymbol{X}_u\} = \{\boldsymbol{X}_{u_1}, \ldots, \boldsymbol{X}_{u_M}\}$ is the unsupervised dataset with M unannotated samples. Here N is small, e.g. no more than 10, and $M \gg N$. Note that N is usually too small to do efficient supervised learning.

2.2.1. Encoder

Encoder aims to capture context and extract pyramid features. It contains two parts: backbone and context extractor module (Gu et al., 2019).

The backbone can be any common architecture with several scaled stages, extracting pyramid features. As shown in Fig. 4, we empirically use ResNet-34 as the backbone in our experiments. As shown in Fig. 4, the backbone extracts four different scaled pyramid feature maps, namely $112 \times 112, 56 \times 56, 28 \times 28, 14 \times 14$.

The context extractor module aims to capture context semantic information and generates more high-level feature maps. It consists of dense atrous convolution (DAC) block and residual multi-kernel pooling (RMP) block. As illustrated in Fig. 4, the DAC block has four cascade branches with an increment of the number of atrous convolution, which adopts different receptive fields to widen the architecture, and the RMP block uses multiple effective field-of-views to detect objects at a different size.

Let $\mathcal{F}_{en}(\boldsymbol{X}; \theta_{en}, s)$ be the backbone with parameter θ_{en} and with s scaled pyramid feature maps:

$$[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s] = \mathcal{F}_{en}(\boldsymbol{X}; \theta_{en}, s), \tag{1}$$

where $\boldsymbol{f}_i, i = 1, \ldots, s$ are s feature maps with different scales, and $\boldsymbol{f}_s$ is the feature maps with the largest scale, namely 14×14 in Fig. 4. Note that $\boldsymbol{f}_s$ denotes the output of RMP.

2.2.2. Decoder

Decoder aims to up-sample the feature maps with deconvolution and generate high-resolution semantic information for the segmentation mask. Note that the size of the mask is the same as the original input image. Here we take a u-shaped architecture, which is symmetric to the encoder and produces features of the same size as that in the encoder. To alleviate features with different scaled, skip connections bridging the encoder and decoder are needed. As shown in Fig. 4, there are three bridges between the decoder and the encoder. These bridges make the information of these two parts fused smoothly. We name the output of the decoder coarse-mask. Let $\mathcal{F}_{de}(\cdot; \theta_{de})$ be the decoder with parameter θ_{de}, then:

$$\boldsymbol{Y}^p_{coarse} = \mathcal{F}_{de}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s; \theta_{de}), \tag{2}$$

$$\boldsymbol{Y}_{coarse} = \mathbb{I}(\boldsymbol{Y}^p_{coarse} > 0.5), \tag{3}$$

where $\boldsymbol{Y}^p_{coarse}$ is a probability matrix for each pixel and $\boldsymbol{Y}_{coarse}$ is the segmentation shown in Fig. 4.

2.2.3. Point refiner

Point Refiner is specially designed for self-supervised learning of leaf veins, which aims to refine the uncertain points, branching points and fix the breakpoints. It consists of four parts: confidence mask module (CMM), point correction module (PCM), point feature extractor (PFE), and point head (PH), as shown in Figs. 5 and 7.

Confidence Mask Module. CMM is a module to attain a prediction label with high confidence and pick out uncertain points for further judgment. Empirically, the high confidence mask contains a small region at the beginning while expands during self-supervised training.

Suppose $\boldsymbol{A}$ is the high confidence mask based on thresholds $\mathcal{T} = \{\mathcal{T}_l, \mathcal{T}_h\}$:

$$\boldsymbol{A}(i,j) = \begin{cases} 1, & \text{if } \boldsymbol{Y}^p_{coarse}(i,j) \geq \mathcal{T}_h \text{ or } \boldsymbol{Y}^p_{coarse}(i,j) \leq \mathcal{T}_l, \\ 0, & \text{others}, \end{cases} \tag{4}$$

where (i, j) are the locations in the image. The thresholds are not sensitive, and we empirically set them to $\mathcal{T} = \{0.90, 0.10\}$. Then, CMM selects the top K uncertain points. The uncertainty map is defined as follows:

$$\boldsymbol{A}_{un} = |(1 - \boldsymbol{A}) \odot \boldsymbol{Y}^p_{coarse} - 0.5|, \tag{5}$$

then the set of uncertain points Ω_K is the minimal K points as follows:

$$\Omega_K = \{(i,j) | Minimal(\{\boldsymbol{A}_{un}(i,j)\}^{i=H, j=W}_{i,j=0}, K)\}, \tag{6}$$

where (H, W) are the shape of output.

Point Correction Module. PCM is a module focusing on the breakpoints and branching points. Breakpoints are points that break out along the vein, and branching points are points where veins branch. Both of them are critical points in segmentation. This module picks out at most L breakpoints and I branching points at each iteration.

(i) Breakpoints. We then select L breakpoint candidates from the direction gradient map $\boldsymbol{G}$ of the decoder predict $\boldsymbol{Y}_{coarse}$. The direction gradient map can be obtained by the classic image gradient operator,

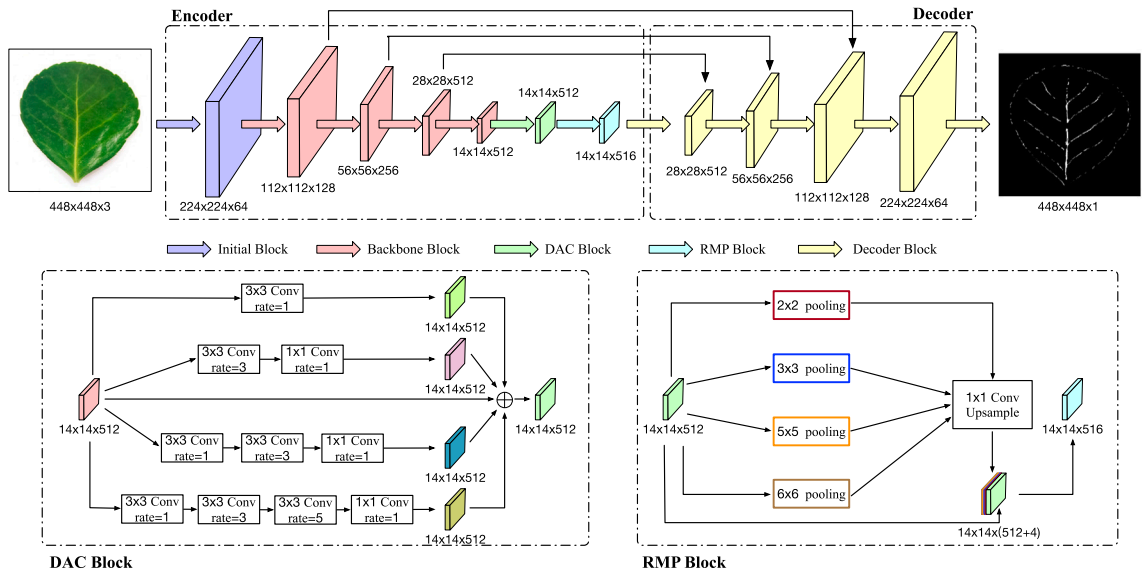

Fig. 4. Illustration of the pipeline of encoder and decoder structure. We use the CE-Net (Gu et al., 2019) as backbone which contains a dense atrous convolution (DAC) block and a residual multi-kernel pooling (RMP) block shown in bottom dotted boxes. The encoder adopts ResNet-34 as feature extractor without any pretraining and the decoder contains 1×1 convolution and 3×3 deconvolution operations, Besides, skip connections are built between symmetrical layers in encoder and decoder.

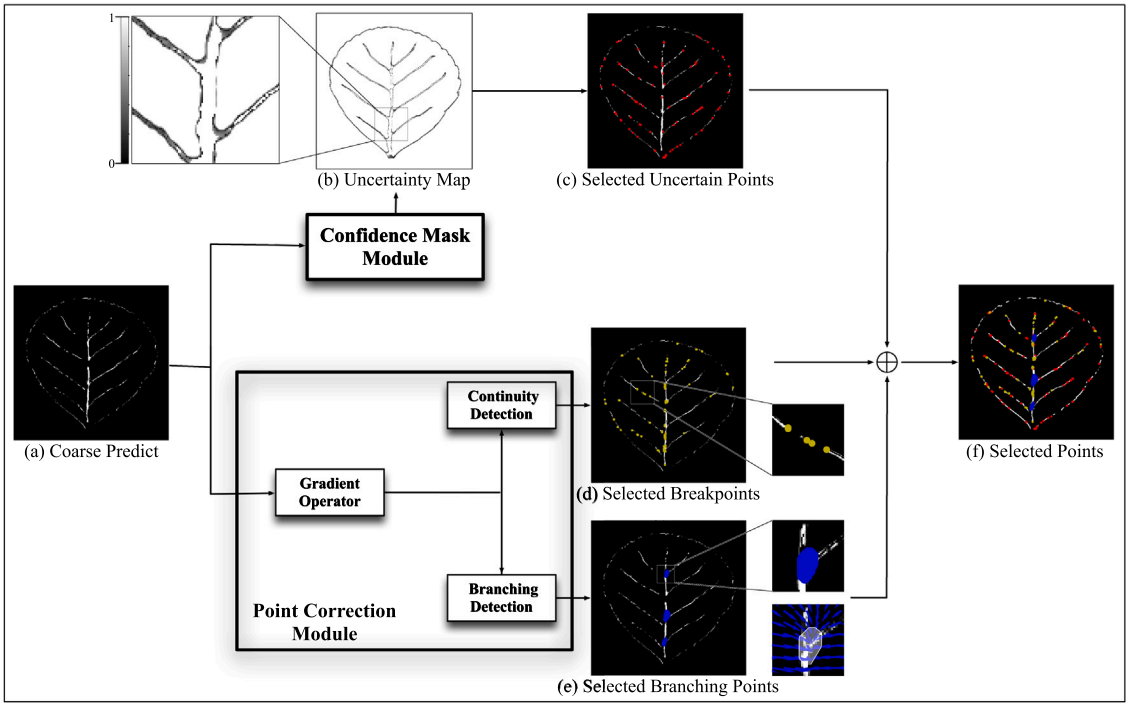

Fig. 5. Pipeline of two parts of point refiner: confidence mask module (CMM) and point correction module (PCM). The parts in top and bottom red dotted boxes are CMM and PCM respectively. CMM select unconfident points in uncertain areas based on uncertainty map while PCM select breakpoints and branching points according to continuity and branching characteristics of leaf veins. These points are further processed by point feature extractor and point head, shown in Fig. 7. (a) is a binary decoder predict, namely coarse mask. (b) shows the uncertainty map, from 0 to 1 indicates the points from the most uncertain to most certain. The images in (c), (d) and (e) demonstrates the three type of final selected points. (f) shows the all selected points. We also show the zoom-in images of breakpoints and branching points for better visualization. The lower right zoom-in image of (e) reveals divergence information of corresponding region in the box, see Fig. 6 for more details.

Sobel operator or Gradient Vector Flow (GVF) (Xu and Prince, 1998) for example,

$$\begin{aligned} \boldsymbol{G} &= [\boldsymbol{G}_x, \boldsymbol{G}_y] \\ &= [Sobel_x(\boldsymbol{Y}_{coarse}), Sobel_y(\boldsymbol{Y}_{coarse})], \end{aligned} \tag{7}$$

$$\boldsymbol{G}_A = \sqrt{\boldsymbol{G}_x^2 + \boldsymbol{G}_y^2}, \tag{8}$$

where $\boldsymbol{G}_x$ and $\boldsymbol{G}_y$ denote the horizontal and vertical gradient of the image respectively as shown in Fig. 6(a) and 6(b), $\boldsymbol{G}_A$ indicate the magnitude of the gradient as shown in Fig. 6(c), for better visualization we depict both magnitude and phase angle indicated by the length and direction of the arrow respectively. It is obvious that the direction of the gradient vector is perpendicular to the edges and veins of the leaf and the vectors gather at the branches of the vein. Intuitively, according to the continuity of leaf veins, when a pixel is surrounded by vein points

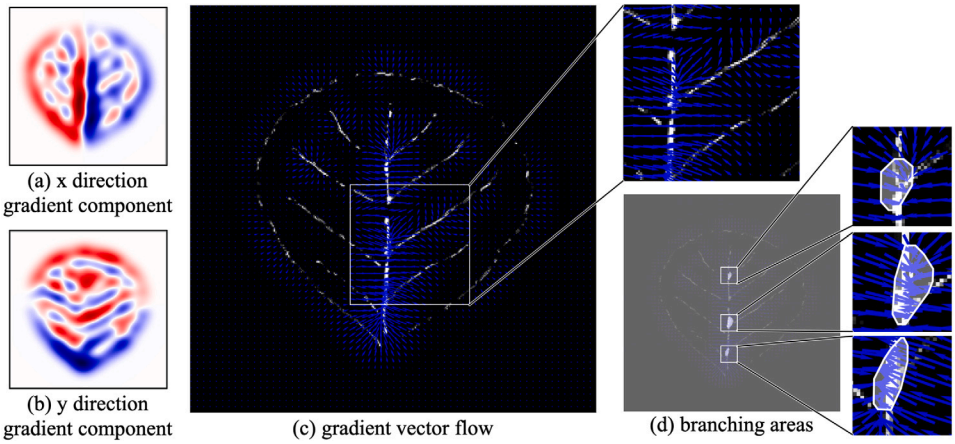

Fig. 6. Illustration of gradient and divergence information of the coarse mask. (a) and (b) show x direction gradient component $\boldsymbol{G}_x$ and y direction gradient component $\boldsymbol{G}_y$, respectively. (c) is a diagram of the gradient vector flow, the length and direction of the arrow indicate magnitude and phase angle of the gradient map, respectively. We add a mask of selected branching points on (c) to attain (d), revealing divergence characteristic that the place where arrows converge is the branching area.

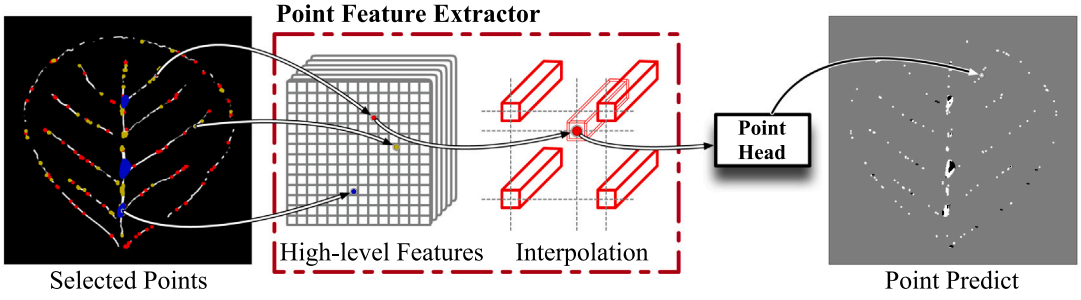

Fig. 7. Diagram of the point feature extractor and point head. To obtain the point features of selected points, we adopt RoIAligh (He et al., 2017) to tackle the coordinate misalignments between the coarse mask (i.e., 448 × 448) and the extracted features (i.e., 14 × 14). The point features are then fed into point head to get the refined labels. The points predicted as the foreground are presented in white, the points predicted as the background are presented in black, and the remaining areas are rendered in gray for better visualization.

along the same vein, it is also a vein point. Therefore, detecting the gradient magnitude of the neighborhoods pixel could find out whether the pixels in the uncertain regions are the breakpoints.

$$\hat{\boldsymbol{G}}_A(i,j) = \mathbb{I}(\boldsymbol{G}_A(i,j) > 0) = \begin{cases} 1, & \text{if} \quad \boldsymbol{G}_A(i,j) > 0, \\ 0, & \text{if} \quad \boldsymbol{G}_A(i,j) \le 0, \end{cases} \tag{9}$$

$$\mathcal{K} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -10 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{10}$$

$$\hat{\boldsymbol{G}}_A^{\mathcal{K}} = Conv(\hat{\boldsymbol{G}}_A, \mathcal{K}). \tag{11}$$

We select the points where $\hat{\boldsymbol{G}}_A^{\mathcal{K}} > \tau$ as breakpoint candidates, τ is set to 4 by default so as to filter the edge points along the veins, which only has a few vein neighbors, and focus on the truly breakpoints.

(ii) Branching points. Divergence characterizes the strength of the divergence of the vector field at each point in space. In physical, when divergence is positive, the point has a positive field source of flux; when divergence is negative, there is a negative field source absorbing flux; when divergence is zero, it is a field without sources. Here we take a similar concept to identify the branching points, which is defined as follows:

$$\boldsymbol{G}_{div} = |\frac{\partial \boldsymbol{G}_x}{\partial x} + \frac{\partial \boldsymbol{G}_y}{\partial y}|. \tag{12}$$

Thus, the divergences of branching points tend to be high, and the divergences of other points tend to be below, as shown in Fig. 6(c). According to this, we select the maximal I points in $\boldsymbol{G}_{div}$ for better judgments. The set of selected branching points Ω_I is defined as follows:

$$\Omega_I = \{(i,j) | Maximal(\{\boldsymbol{G}_{div}(i,j)\}_{i,j=0}^{i=H,j=W}, I)\} \tag{13}$$

Point Feature Extractor. The PFE is a module to extract point feature vectors from scaled feature maps. As shown in Fig. 7. As the size of coarse mask (i.e., 448 × 448) and the semantic feature maps (i.e., 14 × 14) are different, the coordinate projection from selected points on coarse mask to the corresponding position of feature map suffers a quantization loss, introducing misalignments between the selected points and the extracted features. Thus, we adopt RoIAligh (He et al., 2017) to address this problem, which avoids any quantization of the region of interest (RoI) boundaries, i.e., the position of selected points. Therefore, RoIAligh is able to properly align the extracted features with the selected points. Let $\boldsymbol{f}_{ij}$ represents the feature vector at point (i,j), it is a average on a given neighborhood on point (i,j):

$$\boldsymbol{f}_{ij} = Interpolate_{(p,q)\in\Omega_n}(\boldsymbol{f}_{pq}), \tag{14}$$

Ω_n is a neighborhood on point (i,j). In our experiment, Ω_n is four closest points of (i,j).

Point Head. The PH is a multi-layer perceptron (MLP) to predict the label of each point. Let $\mathcal{F}_p(\cdot;\theta_p)$ represent the PH with parameter θ_p, then :

$$\boldsymbol{Y}_{refined}^p = \mathcal{F}_p(\{\boldsymbol{f}_{ij} | (i,j) \in \{\Omega_K, \Omega_L, \Omega_I\}\}; \theta_p), \tag{15}$$

$$\boldsymbol{Y}_{refined} = \mathbb{I}(\boldsymbol{Y}_{refined}^p > 0.5). \tag{16}$$

In our experiment, the PH is a 5-layer MLP with 512, 256, 256, 64, 1 units. Finally, the CoRE-Net output is an integration of $\boldsymbol{Y}_{refined}$ and

$\boldsymbol{Y}_{coarse}$:

$$\boldsymbol{Y}_u = \begin{cases} \boldsymbol{A} \odot \boldsymbol{Y}_{coarse} + \boldsymbol{Y}_{refined}, & \text{self-supervision training,} \\ \boldsymbol{Y}_{coarse} + \boldsymbol{Y}_{refined}, & \text{testing.} \end{cases} \tag{17}$$

2.2.4. Iterative self-supervision

The whole training procedure contains two phases: warm-start training phase and self-supervision training phase. Let $\mathcal{F}_\theta(\cdot)$ represent the whole network with parameter $\theta = [\theta_{en}, \theta_{de}, \theta_p]$.Warm-start training phase is a supervised pretraining to minimize $\mathcal{L}$ on $\mathcal{D}_l$:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\boldsymbol{Y}_l, \mathcal{F}_\theta(\boldsymbol{X}_l; \theta)), \tag{18}$$

where $\mathcal{L}(\cdot)$ is the empirical risk function, i.e., cross-entropy loss.

Then, the self-supervised training phase is an iterative training to minimize the empirical risk loss on $\mathcal{D}_l \cup \mathcal{D}_u$ as follows:

$$\theta^{t+1} = \arg\min_{\theta=[\theta_{en},\theta_{de},\theta_p]} [\mathcal{L}(\mathcal{F}_{\theta^t}(\boldsymbol{X}), \mathcal{F}_{de}(\mathcal{F}_{en}(\boldsymbol{X}; \theta_{en}); \theta_{de})) + \mathcal{L}(\mathcal{F}_{\theta^t}(\boldsymbol{X}), \mathcal{F}_p(\mathcal{F}_{en}(\boldsymbol{X}; \theta_{en}); \theta_p))], \tag{19}$$

where θ^t is the parameters at iteration t, and θ^{t+1} is the parameters at iteration $t+1$. The whole procedure of the iterative self-supervision is shown in Algorithm 1.

Algorithm 1 Iterative Self-supervision

Initial networks parameter $\theta = \{\theta_{en}, \theta_{de}, \theta_p\}$.
for each *epoch* $\in$ [1, num_epochs] **do**
  if i < warm_start_epoch **then**
    // Warm-starting phase
    Sample $(\boldsymbol{X}_l, \boldsymbol{Y}_l)$ from labeled dataset $\mathcal{D}_l$.
    $\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \nabla \mathcal{L}(\boldsymbol{Y}_l, \mathcal{F}_\theta(\boldsymbol{X}_l; \theta))$ // α is the learning rate.
  else
    // Self-supervision phase
    for each *batch* $\in$ [1, num_batches] **do**
      Sample $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{batch\_size}\}$ from $\mathcal{D}_{all}$, $\mathcal{D}_{all} = \{\mathcal{D}_l, \mathcal{D}_u\}$.
      $[\boldsymbol{f}_1, \cdots, \boldsymbol{f}_s] = \mathcal{F}_{en}(\boldsymbol{X}; \theta_{en}, s)$. // Extract the features
      // Get the output probability from *decoder*
      $\boldsymbol{Y}^p_{coarse} = \mathcal{F}_{de}(\boldsymbol{f}_1, \cdots, \boldsymbol{f}_s; \theta_{de})$.
      $\boldsymbol{Y}_{coarse} = \mathbb{I}(\boldsymbol{Y}^p_{coarse} > 0.5)$. // Get the binary coarse mask.

      //CMM, selected uncertain points
      $\boldsymbol{A} = \mathbb{I}(\boldsymbol{Y}^p_{coarse} \geq \mathcal{T}_h \ or \ \boldsymbol{Y}^p_{coarse} \leq \mathcal{T}_l)$. // Calculate confidence mask
      $\boldsymbol{A}_{un} = |(1-\boldsymbol{A}) \odot \boldsymbol{Y}^p_{coarse} - 0.5|$. // Calculate uncertainty map
      // Select K most uncertain points
      $\Omega_K = \{(i,j) | Minimal(\{\boldsymbol{A}_{un}(i,j)\}^{i=H,j=W}_{i,j=0}, K)\}$.

      //PCM
      //(i) selected breakpoints
      // Calculate direction gradient map
      $\boldsymbol{G} = [\boldsymbol{G}_x, \boldsymbol{G}_y] = [Sobel_x(\boldsymbol{Y}_{coarse}), Sobel_y(\boldsymbol{Y}_{coarse})]$.
      $\boldsymbol{G}_A = \sqrt{\boldsymbol{G}_x^2 + \boldsymbol{G}_y^2}$.
      $\hat{\boldsymbol{G}}^{\mathcal{K}}_A = Conv(\mathbb{I}(\boldsymbol{G}_A > 0), \mathcal{K})$
      $\Omega_L = \{(i,j) | \hat{\boldsymbol{G}}^{\mathcal{K}}_A(i,j) > \tau\}$. // Select L breakpoints
      //(ii) selected branching points
      $\boldsymbol{G}_{div} = |\frac{\partial \boldsymbol{G}_x}{\partial x} + \frac{\partial \boldsymbol{G}_y}{\partial y}|$. // Get the divergence map
      // Select I branching points
      $\Omega_I = \{(i,j) | Maximal(\{\boldsymbol{G}_{div}(i,j)\}^{i=H,j=W}_{i,j=0}, I)\}$.

      // Obtain the feature representation for each point
      for each $(i,j) \in \{\Omega_K, \Omega_L, \Omega_I\}$ **do**
        $\boldsymbol{f}_{ij} = Interpolate_{(p,q)\in\Omega_n}(\boldsymbol{f}_{pq})$.
        // Ω_n are neighborhoods of the point (i,j)
      end for
      //get the refined labels from projection *point head*
      $\boldsymbol{Y}_{refined} = \mathcal{F}_p(\{\boldsymbol{f}_{ij} | (i,j) \in \{\Omega_K, \Omega_L, \Omega_I\}\}; \theta_p)$.
      // Integrate the coarse mask and fine mask to the whole one
      $\boldsymbol{Y}_u = (\boldsymbol{A} \odot \boldsymbol{Y}_{coarse}) + \boldsymbol{Y}_{refined}$.
      $\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \nabla \mathcal{L}(\boldsymbol{Y}_u, \mathcal{F}_\theta(\boldsymbol{X}; \theta))$ // α is the learning rate.
    end for
  end if
end for

3. Experiments and results

3.1. Implementation details

We randomly split the dataset into a training set and a test set at a ratio of 7 : 3, as shown in Table 2. We adopt cross validation method to evaluate the effectiveness of the proposed algorithms. To be specific, we performed 5-folder cross validation on training set and present the average values of them. The data preprocessing procedure includes resizing the image to 448 × 448 size and doing color transformation augmentation. We use the Adam optimizer with 0.001 and 1e-5 initial learning rate in warm-start phase and self-supervised phase, respectively. The batch size in the warm-start phase is set to the whole dataset because the supervised dataset is small, i.e., 10 in our experiments. The batch size in the self-supervised phase is set to 64. Warm-start phase and iterative self-supervised phase contain 100 and 300 epochs, respectively. All the experiments run on the Geforce RTX 3090 GPU with Pytorch, and the code/dataset is available.[1]

3.2. Evaluate metrics

Models are evaluated on the following metrics:

- **Acc:** Pixel accuracy (Acc) is a metric to evaluate the average accuracy of each pixel with the following formulation:

$$\text{Acc} = \frac{\sum_{i=0}^{C} n_{ii}}{\sum_{i=0}^{C} \sum_{j=0}^{C} n_{ij}}, \tag{20}$$

  where n_{ij} is the number of pixels with truth label i and prediction label j, $C+1$ is the number of classes (C foreground classes and the background).
- **IoU:** Intersection over Union (IoU) is one of the most commonly used metrics for image segmentation, defined as the ratio of overlap area to union area between the segmented map and the ground truth:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}, \tag{21}$$

  where A and B denote the segmented and the ground-truth maps, respectively.
- **Dice:** Dice coefficient (Dice) is a popular metric to assess the segmentation performance, formulated as follows:

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}. \tag{22}$$

  where A and B denote the segmented and the ground-truth maps, respectively.

In this paper, mAcc, mIoU and mDice are the mean Acc, mean IoU and mean Dice of all the test images.

[1] https://github.com/LeryLee/vein_segmentation

Table 2
The composition of Leaf Vein Dataset 2021 (LVD2021). In our experiments, the samples are randomly divided into training set and test set at a ratio of 7:3.

Category	#Train/#Test	Category	#Train/#Test	Category	#Train/#Test
Walnut	115/50	Grape	71/31	Thistle	91/40
Smoke Tree	91/40	Hibiscus	100/43	Mirabilis Jalapa	119/51
Poplar	140/61	Morning Glory	84/36	Sycamores	70/31
Oriental Cherry	102/44	Apricot	79/34	Lilac	93/41
Chinese Redbud	87/38	Chenopodium Album	90/39	Persimmon	74/32
Crape Myrtle	105/45	Phlox Paniculata	93/40	Mulberry	86/37
Hackberry	102/45	Callistephus Chinensis	93/41	Sichuan Pepper	79/34
Crataegus Pinnatifida	106/46	Maple Tree	95/41	Vitex Negundo Var	94/41
Virginia Creeper	102/45	Amaranth	98/42	Magnolia Denudata	103/45
Forsythia Suspensa	107/46	Honeysuckle	74/33	Chinese Rose	83/36
Fructus Xanthii	98/42	Sweet Potato	93/41	Elm	85/37
Cynanchum	79/35	Cedar	84/37	Holly	204/88

3.3. Comparison methods

Since no off-the-shelf benchmark methods are available to be compared with our method directly on the newly collected dataset VLD2021, we select some excellent models in semantic segmentation task for comparison. FCN (Long et al., 2015) and UNet Ronneberger et al. (2015) are very popular in image segmentation and inspired many great works. DeepLabV3 (Chen et al., 2017), DANet (Fu et al., 2019) and K-Net (Zhang et al., 2021) achieved great performance on semantic segmentation task. BiSeNetV2 (Yu et al., 2021) showed high efficiency for real-time semantic segmentation. UPerNet (Xiao et al., 2018) is able to segment the images into several concepts. GC-Net (Cao et al., 2019) could effectively model long-range dependency and EncNet (Zhang et al., 2018) could captures the semantic contextual information and selectively spotlight the class-dependent featuremaps. HRNet (Sun et al., 2019) has the ability to maintain high-resolution representations through the whole process.

In addition, to verify the importance of the initialization of the weights in self-supervision phase, we add a 16-layers CNN with residual connections for comparison, which is composed of 3×3 convolution layers, each followed by a rectified linear unit (ReLU). In order to obtain high-resolution segmentation predict, we keep the size of feature maps fixed, without any downsampling operations and the number of feature map channels are set to 64.

3.4. Comparison of leaf vein segmentation

We compare the proposed CoRE-Net with some state-of-the-art methods on segmentation tasks. Detailed results are shown in Table 3. The left side of the table shows the results of different methods under supervised training with the same setting of our warm-start phase, and the right side of the table shows the results of different methods under iterative self-supervised learning. Methods with "-SL" are all trained by an iterative supervised training with pseudo labels. "CoRE-Net-SL" is the proposed Algorithm 1. Besides, We compare our methods to traditional methods such as high-pass filters. We implement both Fast Fourier transform (FFT) high-pass filter and Wavelet high-pass filter for comparison.

At the bottom of the table, the "CoRE-Net-SL (supervised)" presents the upper limit of CoRE-Net-SL, which are under supervised training with the whole datasets ($D_l \cup D_u$). As self-supervised learning aims to predict the pseudo labels of the unlabeled samples to train the model, the best performance could be achieved by training the model using the ground truth labels directly.

Compared to HRNet, which is one of the best comparison methods, CoRE-Net-WS achieved **8.43%**, **2.46%** and **2.71%** improvement in terms of mAcc, mIoU and mDice, respectively. Furthermore, compared to CoRE-Net-WS, CoRE-Net-SL further achieved **9.38%**, **8.25%** and **7.77%** improvements on in terms of mAcc, mIoU and mDice, respectively. These obvious improvements demonstrate the effectiveness of both CoRE-Net and the iterative self-supervised training. From the table, other methods with "-SL" decline compared to the corresponding experiment without iterative self-supervised learning. That is because pseudo labels predicted by parameters in the previous iteration easily lead to overfitting. However, Point Refiner in CoRE-Net makes the pseudo labels converge and promote the model training. Thus, CoRE-Net outperforms other methods.

3.5. Effect of self-supervised learning

As present in Table 3, pseudo labels predicted by parameters in the previous iteration easily lead to other methods overfit. However, Point Refiner in CoRE-Net makes the pseudo labels more accurate and boost training performance. For better visualization, we show the performance curves of different models during self-supervised learning in Fig. 8. Firstly, the figure illustrates that the proposed model outperforms all other methods in a self-supervision manner. Secondly, we can see that our model benefits from the self-supervision process, while the performance drops for other methods. It is consistent with the above observation that a rough manner of self-labeling could lead to model collapse, hurting performance by a large amount.

Fig. 9 depicts sample results for our methods. We can see that our method is robust to the noisy pseudo label and delineates the target with accurate contours. It can be observed that only a few veins with high confidence are segmented in the beginning, then the high-confidence regions expand, and more veins are segmented during self-supervised learning. The proposed CoRE-Net keeps training in the right direction and takes advantage of the information from the unlabeled data to improve our prediction results.

3.6. Effect of labeled dataset size

We also analyze the impact of warm-start training with a different number of labeled data. Table 4 shows that more labeled data is useful in the warm-start training phase. In our experiments, we simply choose ten labeled images because it can obtain a better improvement compared to fewer images and does not cost much time for annotation.

3.7. Ablation study

Table 5 shows the ablation study when a particular component of the proposed pipeline is removed. Table 6 shows the ablation study when each component of point refiner is replaced with high-pass filter. Fig. 11 shows the performance curves with different settings during self-supervision training, i.e., with/without the point refiner block. As the point feature extractor and point head depend on the information flow from confidence mask module(CMM) and point correction module (PCM), thus, we focus ablation study on the latter two modules, more specifically, with/without CMM and PCM. Fig. 10 demonstrates the qualitative results of different settings. We designed the ablation experiments in three aspects: (1) CoRE-Net with or without point refiner

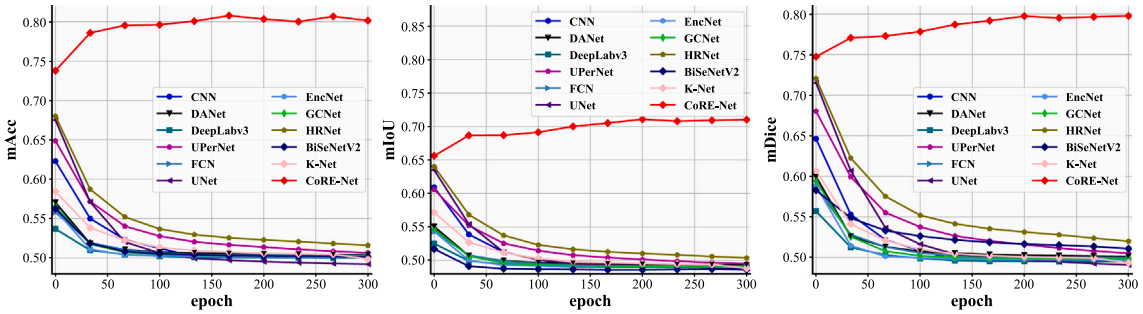

Fig. 8. The performance under various models during self-supervised learning. All of the models present a degraded performance expect the proposed CoRE-Net.

Fig. 9. Qualitative results of our method (GT: ground truth), taking *Holly* for example.

Table 3
Comparing our results against various approaches measured in % of mAcc, mIoU and mDice.

Method	mAcc	mIoU	mDice	Method	mAcc	mIoU	mDice
FFT high-pass filter	55.22	28.72	38.06				
Wavelet high-pass filter	42.43	22.11	31.98				
CNN (16 layers)	62.29	51.59	57.05	CNN-SL[a]	49.98	48.78	49.42
DANet (Fu et al., 2019)	57.06	55.05	59.97	DANet-SL	50.01	48.92	49.46
DeepLabV3 (Chen et al., 2017)	53.70	52.51	55.72	DeepLabV3-SL	49.81	48.56	49.28
UPerNet (Xiao et al., 2018)	64.88	60.60	68.03	UPerNet-SL	50.00	48.95	49.47
FCN (Long et al., 2015)	55.79	54.30	58.69	FCN-SL	49.98	48.71	49.41
UNet (Ronneberger et al., 2015)	67.52	63.50	71.53	UNet-SL	48.59	48.60	48.95
EncNet (Zhang et al., 2018)	56.06	54.32	58.87	EncNet-SL	50.00	48.89	49.44
GCNet (Cao et al., 2019)	56.56	54.68	59.46	GCNet-SL	49.99	48.88	49.43
BiSeNetV2 (Yu et al., 2021)	56.25	54.46	59.01	BiSeNetV2-SL	50.11	48.97	49.48
K-Net (Zhang et al., 2021)	58.44	54.89	59.12	K-Net-SL	49.87	48.99	49.49
HRNet (Sun et al., 2019)	68.02	64.03	72.06	HRNet-SL	50.38	49.13	49.95
CoRE-Net-WS	**73.76**	**65.61**	**74.01**	CoRE-Net-SL	**80.68**	**71.02**	**79.76**
	–	–	–	CoRE-Net-SL (supervised)	85.68	74.01	82.78

[a]SL: methods with self-supervised learning.
Bold entries represent the best values in pre-training and self-learning methods and underlined values represent the second best results. Note that only 10 labeled samples are used under these circumstances.

Table 4
Performances of different labeled dataset size on warm-start training.

Size	mAcc	mIoU	mDice
1	50.12	49.06	56.85
3	59.75	55.75	64.04
5	64.37	61.36	70.43
8	70.09	63.11	72.20
10	73.76	65.61	74.01
All	85.68	74.01	82.78

(namely CMM + PCM). We also conduct experiments removing the CMM or PCM individually from point refiner module. (2) Replace the backbone of CoRE-Net with CNN, i.e., CNN with point refiner module. (3) Replace the point refiner (namely CMM + PCM) with high-pass filter, and replace CMM or PCM with high-pass filter individually. We implement both Fast Fourier transform (FFT) high-pass filter and Wavelet high-pass filter for comparison.

As for (1), it can be seen that removing the CMM and PCM leads to a degraded performance by a large amount, i.e., the mAcc/mIoU/mDice drops 8.78%/42.54%/38.16%, likely because of error accumulation leading to model collapse, which verifies the effectiveness of the point refiner module. We observe that collapse can be avoided with the CMM, and it helps to improve mIoU and mDice by 2.69% and 3.17%, respectively. In contrast, in the absence of CMM, while adding the PCM, the mAcc/mIoU/mDice still drops 7.23%/22.03%/20.15%, but much lower than that without PCM, which indicates that this module works as a correction mechanism but cannot catch up with the speed of error accumulation. We also observe that only if we keep the CMM can the model benefit most from the PCM. Note while directly removing CMM and PCM, namely removing the whole point refiner block, it will result in model collapse but have little impact on mAcc, which may be caused by the unbalanced distribution of background and foreground, i.e., for an input image, the number of background pixels is far more than that of foreground pixels, thus even if the most pixels are predicted as background, the mAcc could remain to a high value.

Table 5
Ablation study.

Phase	Backbone	Point refiner		mAcc	mIoU	mDice
		CMM	PCM			
Warm-start training	CNN (16 layers)			62.29	51.59	57.05
	CoRE-Net			73.76	65.61	74.01
Self-supervised training	CNN (16 layers)	✓	✓	73.14 (+17.42)	58.88 (+14.13)	67.69 (+18.65)
	CoRE-Net	✗	✗	73.60 (−0.08%)	40.81 (−37.39%)	49.32 (−33.36%)
		✗	✓	68.42 (−7.23%)	51.15 (−22.03%)	59.09 (−20.15%)
		✓	✗	69.52 (−5.74%)	67.38 (+2.69)	76.36 (+3.17%)
		✓	✓	**80.68** (+9.38%)	**71.02** (+8.25)	**79.76** (+7.77%)

Bold entries represent the best values self-learning methods. The values in brackets represent the increased percentage points improved by self-supervision training under the same backbone setting.

Table 6
Ablation study of replacing point refiner module with high-pass filters.

Phase	Point refiner		**mAcc**	**mIoU**	**mDice**
Warm-start training			73.76	65.61	74.01
Self-supervised training	CMM	PCM	**80.68**	**71.02**	**79.76**
	FFT	PCM	30.57 (−62.11%)	17.91 (−74.78%)	26.85 (−66.34%)
	Wavelet		50.00 (−38.03%)	48.67 (−31.47%)	49.33 (−38.15%)
	CMM	FFT	29.55 (−63.37%)	18.79 (−73.54%)	27.74 (−65.22%)
		Wavelet	50.00 (−38.03%)	48.64 (−31.51%)	49.31 (−38.18%)
	FFT		31.11 (−61.44%)	18.05 (−74.58%)	27.01 (−66.14%)
	Wavelet		50.00 (−38.03%)	48.64 (−31.51%)	49.31 (−38.18%)

Bold entries represent the best values self-learning methods. The values in brackets represent the decline percentage points compared to the best result.

As for (2), it can be observed that CNN benefits from the point refiner module and is prompted to achieve better performance, i.e., the mAcc/mIoU/mDice improves 17.42%/14.13%/18.65%, which shows that this module can be flexibly added to other networks so as to boost its performance. Moreover, the performance of CNN with point refiner module lagged behind the CoRE-Net with the same module, which demonstrates that a good initialization of the weights is critically important.

As for (3), we found that whether replace the CMM and PCM individually, or replace the whole point refiner module with high-pass filter, all lead to pattern collapse. As shown in Table 6, replacing the Point Refiner (namely CMM + PCM) with FFT(Wavelet), the mAcc/mIoU/mDice relatively drops 61.44%(38.03%)/74.58%(31.51%) /66.14%(38.18%), respectively. This may because the high pass filter is not flexible enough to select key points for prediction, destroying the stability of the CMM and resulting in error accumulation, which makes the model go further and further in the wrong direction.

The ablation study shows the importance of the components in the pipeline, namely CMM and the PCM of point refiner in the self-supervision phase. Training on the pseudo-labels from the warm-started model without CMM and PCM directly causes consistently poor performance. Filtering the particular confidence maps with CMM step by step already leads to substantial performance improvement, and the performance further increases with PCM by fixing the breakpoints and branching points. This additional refinement improves our prediction results with more iterations of self-supervised training.

4. Discussion

4.1. Why does the confidence mask module works?

We explore the distribution of the prediction results in the self-supervision phase as shown in Fig. 12. For the leaf vein segmentation task, the model predicts a mask where the background pixels are 0, and foreground pixels are 1. We can see in Fig. 12 on the left that the output distribution of the self-training model clustered on the ends of 0 and 1 compared to the pre-training model. As stated in Section 2.2, the confidence mask keeps the regions with high confidence i.e., close to 0 or 1, while ignores the regions with low confidence, forcing the model to focus on the regions that are most likely to be correct and learn how to generalize these features to the uncertain regions. Therefore, such a strategy helps the outputs to be close to 0 or 1. A proper metric to measure the model's ability to separate the background and foreground pixels is Area under the ROC Curve (AUC). AUC measures the entire two-dimensional area underneath the entire ROC curve, and the higher AUC score represents the stronger ability at distinguishing between the background and foreground pixels. As shown in Fig. 12 on the right, the AUC score increased from 0.9532 to 0.9745 increased by 2.23% with our self-supervision training.

4.2. Why does the point correction module works?

To further understand the reasons why the point correction module works well in our framework, we study its point selection results during training. Fig. 5(b) shows the uncertainty map, we can find that the most uncertain area is precisely the boundaries between the leaves and the outside background, and the boundaries between the mesophyll and the veins, which is in line with our intuition as these regions are most difficult to distinguish. Fig. 5(c),(d) and (e) shows the selected points towards uncertain regions, breakpoints and branching areas, respectively, making predictions on these carefully selected points could result in sharper boundaries and more precise leaf veins.

Besides, the selected points supplement key information for the model to identify the regions that are hard to distinguish or missing surrounding information. The self-supervision training strategy iteratively enhance the model's ability to generate leaf vein segmentation results more accurate with higher confidence. Our proposed model makes full use of the crucial characteristics of leaf veins, i.e., continuity and branching, to smooth the break regions along the veins and spread the branching information to the surround areas, e.g., from primary vein to secondary vein. Benefiting from this, the model is able to generate the pseudo labels with high quality and the refined labels in turn promote the model's performance.

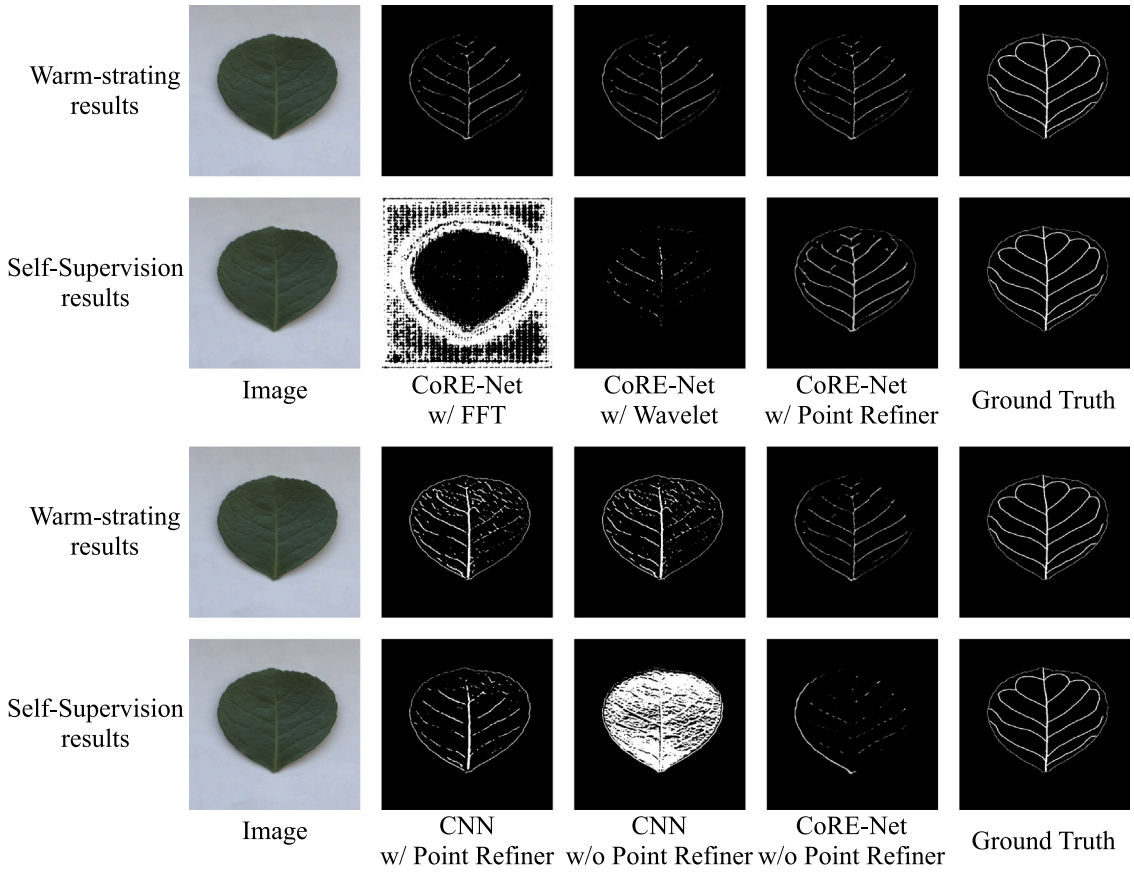

Fig. 10. Qualitative results of different methods, taking *Holly* for example.

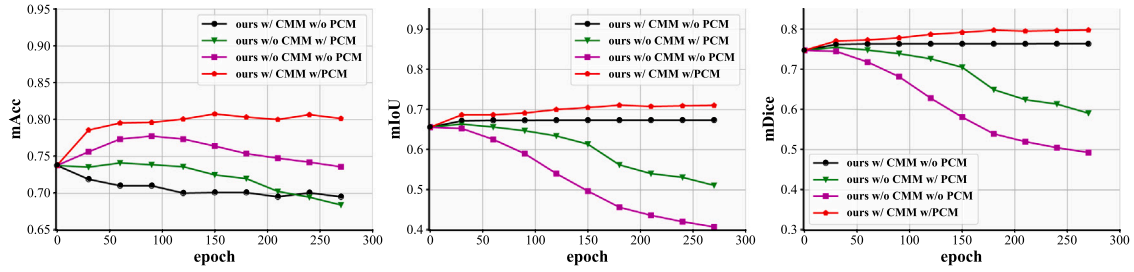

Fig. 11. The performance under various settings during self-supervision training for ablation study.

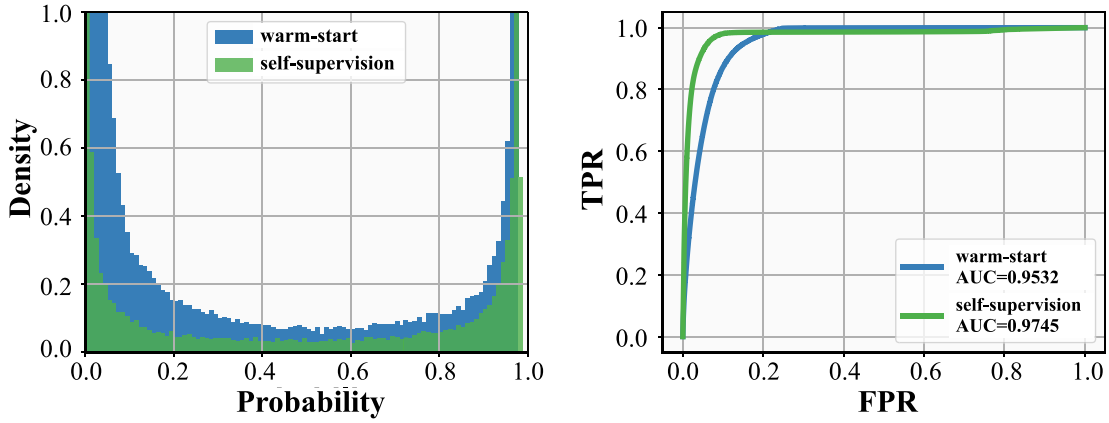

Fig. 12. The distribution and AUC results for warm-start and self-supervised training. **Left**: The distribution of the output probability. The self-supervised training model with CMM results in a better distribution clustering more pixels close to 0 and 1. **Right**: AUC results for warm-start and self-supervised training. The latter achieves higher AUC score.

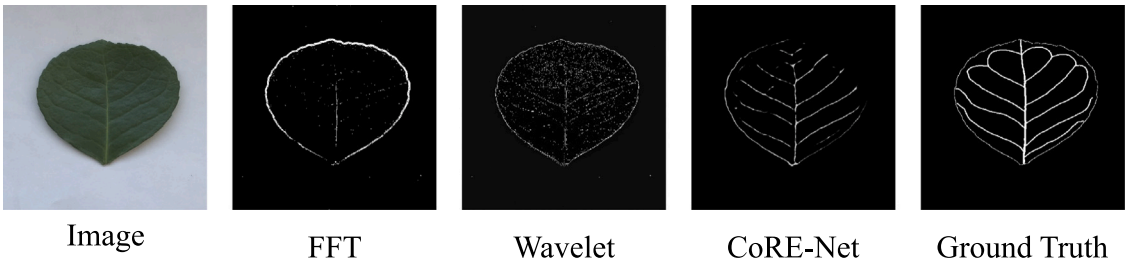

Fig. A.13. Qualitative results of traditional methods, taking *Holly* for example.

5. Conclusion

In this study, a novel and effective leaf vein segmentation framework based on self-supervised learning was proposed, namely CoRE-Net, which aimed to extract the leaf vein from coarse to fine in a self-supervised manner with unlabeled samples. This technology explored the leaf veins' characteristics for the leaf vein segmentation with a specially designed module, point refiner. The results showed that this module has ability to capture the key information according to the particularities of the leaf veins, namely continuity and branching, and refine the pseudo labels. It could also infer the uncertain regions and correct errors, so as to improve the supervisory signal for training the model.

We exploited the proposed approach to evaluate on the newly collected dataset LVD2021, the results demonstrated the strong potential of the self-supervised methods, and validate that our CoRE-Net with point refiner can lead to more precise and continuous leaf vein segmentation results than other methods.

Overall, this research has developed a method which has significant potential to boost the segmentation performance. Keeping the high confidence regions and utilizing veins' particularities prompted the self-supervised training. It is likely that this strategy reserved and spread the supervision signal and the model benefiting from it could generate more precise pseudo labels.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data/code in the paper.

Acknowledgments

This work is funded by the Natural Science Foundation of China (NSFC No. 62061136001 and No. 62176132) and the German Research Foundation (DFG) in Project Crossmodal Learning, DFG TRR-169.

Appendix. Supplementary material

We compare our methods to traditional methods such as high-pass filters. We implement both Fast Fourier transform (FFT) high-pass filter and Wavelet high-pass filter for comparison. We use high-pass filter to process the input image to obtain the contour edge and vein without prior training. For neural network, i.e. CoRE-Net, we train the network with only 10 labeled samples. Fig. A.13 and Table 3 show the qualitative and quantitative results of these methods, respectively.

The difficulty of leaf vein segmentation is the color of veins and mesophyll is indistinguishable and we focus on the internal details inside the mesophyll. High-pass filter is one of the most effective methods in traditional image processing, especially when extracting the edges with obvious image contrast. Thus the high-pass filter is able to segment the contour edge of the leaves, but this methods cannot extract clear veins accurately because some small leaf veins are low-frequency parts, which makes the leaf veins and mesophyll indistinguishable. The experiments show that the CoRE-Net outperforms the high-pass filters by a large margin.

References

Boyce, C.K., Brodribb, T.J., Feild, T.S., Zwieniecki, M.A., 2009. Angiosperm leaf vein evolution was physiologically and environmentally transformative. Proc. R. Soc. B: Biol. Sci. 276 (1663), 1771–1776.

Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H., 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE International Conference on Computer Vision Workshops.

Carvalho, M.R., Losada, J.M., Niklas, K.J., 2018. Phloem networks in leaves. Curr. Opin. Plant Biol. 43, 29–35.

Chakkaravarthy, S.S., Sajeevan, G., Kamalanaban, E., Kumar, K.V., 2016. Automatic leaf vein feature extraction for first degree veins. In: Advances in Signal Processing and Intelligent Recognition Systems. Springer, pp. 581–592.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

Chen, L.C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Fu, H., Cheng, J., Xu, Y., Wong, D.W.K., Liu, J., Cao, X., 2018. Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE Trans. Med. Imaging 37 (7), 1597–1605.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H., 2019. Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3146–3154.

Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., Zhang, T., Gao, S., Liu, J., 2019. CE-net: Context encoder network for 2D medical image segmentation. IEEE Trans. Med. Imaging 38 (10), 2281–2292. http://dx.doi.org/10.1109/tmi.2019.2903562.

Han, J., Luo, P., Wang, X., 2019. Deep self-learning from noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5138–5147.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Iscen, A., Tolias, G., Avrithis, Y., Chum, O., 2019. Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5070–5079.

Jeon, W.S., Rhee, S.Y., 2017. Plant leaf recognition using a convolution neural network. Int. J. Fuzzy Logic Intell. Syst. 17 (1), 26–34.

Lalonde, S., Tegeder, M., Throne-Holst, M., Frommer, W., Patrick, J., 2003. Phloem loading and unloading of sugars and amino acids. Plant Cell Environ. 26 (1), 37–56.

Larese, M.G., Granitto, P.M., 2016. Finding local leaf vein patterns for legume characterization and classification. Mach. Vis. Appl. 27 (5), 709–720.

Larese, M.G., Namías, R., Craviotto, R.M., Arango, M.R., Gallo, C., Granitto, P.M., 2014. Automatic classification of legumes using leaf vein image features. Pattern Recognit. 47 (1), 158–168.

Lee, D.H., 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, Vol. 3. (2).

Lin, K., Zhao, H., Lv, J., Li, C., Liu, X., Chen, R., Zhao, R., 2020. Face detection and segmentation based on improved mask R-CNN. Discrete Dyn. Nat. Soc. 2020.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8759–8768.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440.

Nguyen, D.T., Dax, M., Mummadi, C.K., Ngo, T.P.N., Nguyen, T.H.P., Lou, Z., Brox, T., 2019. Deepusps: Deep robust unsupervised saliency prediction with self-supervision. arXiv preprint arXiv:1909.13055.

Patro, B.N., GS, K., Jain, A., Namboodiri, V.P., 2021. Self supervision for attention networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 726–735.

Radha, R., Jeyalakshmi, S., 2014. An effective algorithm for edges and veins detection in leaf images. In: 2014 World Congress on Computing and Communication Technologies. IEEE, pp. 128–131.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sack, L., Frole, K., 2006. Leaf structural diversity is related to hydraulic capacity in tropical rain forest trees. Ecology 87 (2), 483–491.

Selda, J.D.S., Ellera, R.M.R., Cajayon, L.C., Linsangan, N.B., 2017. Plant identification by image processing of leaf veins. In: Proceedings of the International Conference on Imaging, Signal Processing and Communication. pp. 40–44.

Shi, W., Gong, Y., Ding, C., Tao, Z.M., Zheng, N., 2018. Transductive semi-supervised deep learning using min-max features. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 299–315.

Sun, K., Xiao, B., Liu, D., Wang, J., 2019. Deep high-resolution representation learning for human pose estimation. In: CVPR.

Tan, J.W., Chang, S.W., Kareem, S.B.A., Yap, H.J., Yong, K.T., 2018. Deep learning for plant species classification using leaf vein morphometric. IEEE/ACM Trans. Comput. Biol. Bioinform..

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 418–434.

Xu, H., Blonder, B., Jodra, M., Malhi, Y., Fricker, M., 2020. Automated and accurate segmentation of leaf venation networks via deep learning. BioRxiv.

Xu, C., Prince, J.L., 1998. Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. 7 (3), 359–369.

Yan, M., Wang, J., Li, J., Zhang, K., Yang, Z., 2020. Traffic scene semantic segmentation using self-attention mechanism and bi-directional GRU to correlate context. Neurocomputing 386, 293–304.

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N., 2021. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. Int. J. Comput. Vis. 129 (11), 3051–3068.

Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A., 2018. Context encoding for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition. CVPR.

Zhang, W., Pang, J., Chen, K., Loy, C.C., 2021. K-net: Towards unified image segmentation. Adv. Neural Inf. Process. Syst. 34.