Instance Mask Growing on Leaf

Chuang Yang[1,2]
omtcyang@gmail.com
Haozhao Ma[1,2]
haozhaoma@mail.nwpu.edu.cn
Qi Wang[*2]
crabwq@gmail.com

[1] School of Computer Science, Northwestern Polytechnical University, Shaanxi, P. R. China
[2] School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Shaanxi, P. R. China

Abstract

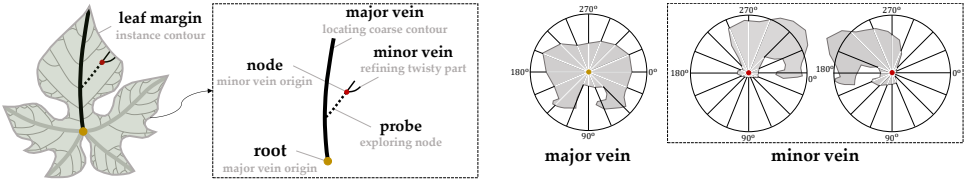
Contour-based instance segmentation methods represent masks through a series of points. However, the point number is fixed once the model is trained, which limits the model's flexibility in dealing with various instances. We follow this issue and present an idea to predict an appropriate number of points dynamically according to instance shapes. Concretely, we observe that the leaf locates coarse margins via major veins and grows minor veins to refine twisty parts, which helps cover any masks accurately. Meanwhile, major and minor veins share the same growth mode, which makes it possible to generate minor veins dynamically according to the trained major vein mode. Considering the superiorities above, we propose VeinMask to formulate the instance segmentation problem as the simulation of the vein growth process and to predict the major and minor veins in polar coordinates for instance segmenting.

Besides, centroidness is introduced for instance segmentation tasks to help suppress low-quality instances. Furthermore, a surroundings cross-correlation sensitive (SCCS) module is designed to enhance the feature expression by utilizing the surroundings of each pixel. Additionally, a Residual IoU (RIoU) loss is formulated to supervise the regression tasks of major and minor veins effectively. Experiments demonstrate the effectiveness of VeinMask. Particularly, our method outperforms existing one-stage contour-based methods on the COCO dataset with almost half the trained point number. Code is available at: https://github.com/omtcyang/veinmask.

1 Introduction

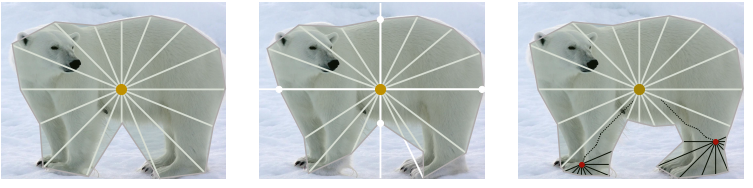
Instance segmentation is one of the challenging computer vision tasks, which provides essential information for many intelligent applications (such as security monitoring and self-driving). With the rapid development of deep learning, instance segmentation has achieved great progress. Some works [24, 29] follow the intuitive idea and aim to represent instances through dense contour points and predict them via detection techniques only, which ensures simple pipelines. However, these methods suffer from the same problem: the point number is fixed once the model is trained. It limits them to deal with various instances flexibly.

* denotes the corresponding author.



(a) VeinMask structure (b) Vein in polar coordinate

Figure 1: Visualization of VeinMask. (a) shows the details of VeinMask and (b) illustrates that how those major and minor veins are modeled in the same polar coordinate.



(a) PolarMask [24] (b) E2EC [29] (c) Ours

Figure 2: Visualization of the essential differences between VeinMask and previous works. The '**white lines**' in (a)–(c) are predefined for generating a series of points. The point number is fixed once the model is trained. The '**black lines**' in (c) are generated dynamically in the inference process according to instance geometries.

Considering the issues above, we aim for pursuing to generate an appropriate number of points to cover instance masks according to their shapes in the inference process. Specifically, We combine morphology with deep learning and propose to simulate the leaf vein growth process to segment instance masks. As shown in Figure 1(a), the leaf vein, a directed graph that can represent any complex geometries, is composed of major and minor veins. In the growth process, major veins sprout out from the root to locate the coarse leaf margin at first. Then, minor veins grow from the node to refine the twisty part. We observe the process and find that major and minor veins share the same growth mode. It means we can grow minor veins following the major vein mode, which makes it possible to generate appropriate points for various instances dynamically in the inference process. Inspired by the superiorities above, we design VeinMask and construct a single-shot instance segmentation framework based on it, which simulates the leaf vein growth process to segment instances dynamically according to instance mask shapes (Figure 2(a)–(c)). Concretely, we model major and minor veins through a polar coordinate with n directions (as depicted in Figure 1(b)) to assemble masks by the following steps: (1) locating the coarse contour through instance centroid ('root') classification and dense distance ('major vein') regression in the polar coordinate built at the root; (2) exploring the twisty part origin ('node') according to adjacent major veins; (3) refining the twisty part through dense distance ('minor vein') regression in the polar coordinate built at the node dynamically; (4) assembling instance contour by connecting all endpoints of major and minor veins in a clockwise direction.

Meanwhile, we present centroidness to help suppress low-quality instances in the inference process. It forces our method to focus on the instance centroid and spreads around in a Gaussian distribution, which is more effective for instance segmentation tasks and easier to learn for the model. Besides, considering the vein plays a key role in assembling instance

masks and the weak features make it hard to regress veins accurately, a surroundings cross-correlation sensitive (SCCS) module is proposed. It helps the model utilize the surrounding information of each pixel to encourage extracting strong expression features while ensuring the enhanced pixel dominance to the surroundings, which can suppress the negative effects brought by surrounding features. Furthermore, a Residual IoU (RIoU) loss is formulated for supervising the regression of the major and minor veins. It inherits the IoU loss [28] advantage to correlate the veins in all directions of polar coordinates. Particularly, our RIoU loss focuses on the residual between the predicted and real values, which can optimize the model more effectively. The contributions are summarized as follows:

1. We combine morphology with deep learning to design a VeinMask, which simulates the leaf vein growth process to generate an appropriate number of contour points for representing instances accurately. It helps enhance model's ability for dealing with different shape masks flexibly.

2. Centroidness is proposed for suppressing low-quality results. It focuses on the instance centroid and spreads around in a Gaussian distribution related to the instance geometries, which can easier bring significant performance gains for instance segmentation tasks compared to the centerness in FCOS and PolarMask.

3. A surroundings cross-correlation sensitive (SCCS) module is introduced. It helps our model enhance the feature expression by utilizing the surrounding information of each pixel to encourage regression tasks. Importantly, the module ensures the enhanced pixel dominates to surroundings, which can suppress the negative effects brought by surrounding features.

4. A Residual IoU (RIoU) loss is formulated for supervising the regression of the major and minor veins. Remarkably, it inherits the IoU loss [28] advantage and focuses on the residual between the predicted and real values, which is more effective for instance segmentation tasks compared to IoU loss and Polar IoU loss.

2 Related Work

Mask-based instance segmentation methods. The intuitive idea for instance segmentation is combining existing object detection and semantic segmentation techniques. Typically, Mask R-CNN [8] followed the design of Faster R-CNN [17]. It predicted the bounding boxes of instances at first and then segmented precise masks within the boxes, which inspired plenty of following works (such as PANet [13] and Cascade Mask R-CNN [2]). However, these methods failed to segment large instances precisely because of ROIAlign operators. With the development of one-stage detection methods, some works [1, 3, 20, 27] proposed to segment masks on the whole feature maps directly, which accelerated the inference speed while ensuring competitive performance. However, they had to assemble instance masks through complex processes. SOLO [21, 22] followed the design of YOLO [16]. It segmented masks by predicting the relationships between each pixel and all grids. However, the grid mechanism made the detection of small instances difficult and depended on the grids deeply.

Contour-based instance segmentation methods. Mask-based methods adopt a hybrid design of detection and segmentation, which leads to inherent model complexity. To pursue a straightforward and effective instance segmentation pipeline, recent methods [5, 12, 14,

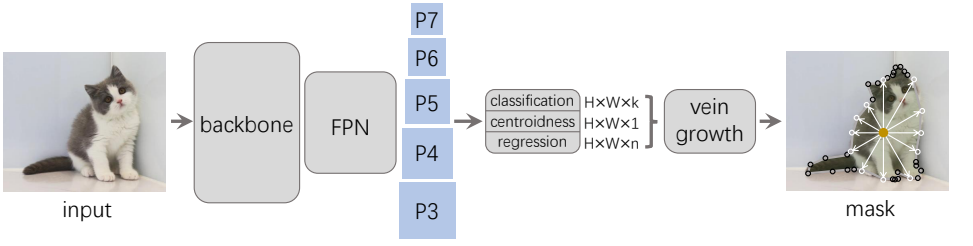


Figure 3: Overall architecture. H and W are the height and width of feature maps. k and n are the instance category number and the polar coordinate direction number. Vein growth is responsible for assembling masks. The '**white circles**' are fixed contour points. The '**black circles**' are generated dynamically according to instance shapes.

[15, 23, 24, 26, 29] formulated instance segmentation problem as contour points prediction. DenseReppoints [26] predicted a series of boundary points and the corresponding categories to represent instance masks. Some researches [12, 14, 15, 23, 29] proposed to regress ordered contour point sequences. They improved the predicted point reliability through a multi-stage point refinement structure and achieved remarkable performance. But the refinement structure complicated the model seriously and brought expensive research costs (such as computational and time costs) for the following researchers. PolarMask [24] inherited the anchor-free detection framework [19]. The authors regressed the offsets between instance centers and boundaries to generate dense ordered contour points for rebuilding masks, which could reach a quite fast detection speed that is almost equivalent to that of the one-stage detectors. However, the mask cover quality of PolarMask [24] is limited.

3 Methodology

We construct a single-shot framework based on VeinMask to segment instances precisely through an appropriate number of contour points dynamically. Our method aims to simulate the leaf vein growth process to reconstruct instance masks. It assembles contours by the 'root' and 'node' classification and 'major vein' and 'minor vein' (see Figure 1(a)) regression. Figure 3 illustrates the details of our framework, which consists of backbone, FPN [10], three heads (including classification, centroidness, and regression heads), and vein growth. They will be introduced next in detail.

Building veins in polar coordinates. As depicted in Figure 1, VeinMask represents instance masks according to the combinations of 'root' and 'major vein', and 'node' and 'minor vein' respectively. In practice, we train our model to fit a polar coordinate with n directions (see Figure 1(b)), where each direction will generate a single contour point. For some complex masks, our VeinMask will generate extra points dynamically based on the polar coordinate for refining twisty parts of contours (black lines in Figure 2(c)).

Backbone and FPN. Following previous works, we extract hierarchical features (see Figure 3$P_3 \sim P_7$). Meanwhile, we follow FCOS [19] to allocate instances into different level feature maps from FPN. Concretely, supposing l^*, t^*, r^*, b^* are the distances between centroids and instance bounding boxes in left, top, right, and bottom directions, the instance is distributed to P_k if $\max(l^*, t^*, r^*, b^*) \in S_k$, where $\{S_k | , k = 3,4,5,6,7\} = \{(-1,64),(64,128),(128,256),(256,512),(512,+\infty)\}$.

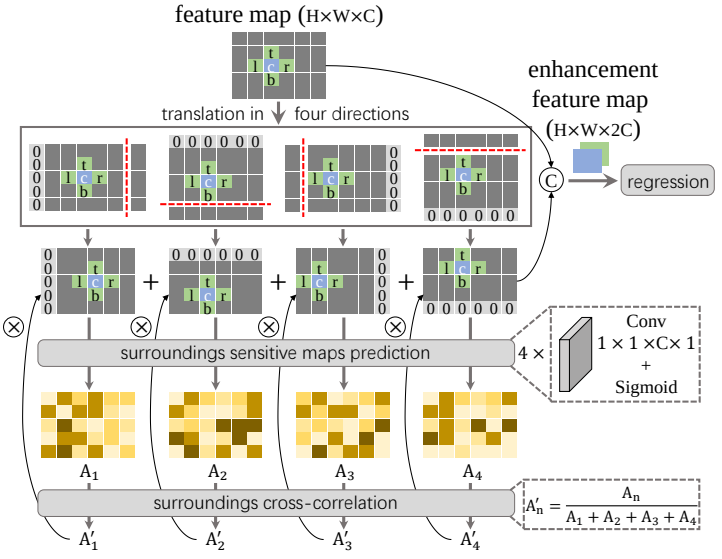


Figure 4: Visualization of SCCS module. This module aims to enhance feature expression through surrounding information.

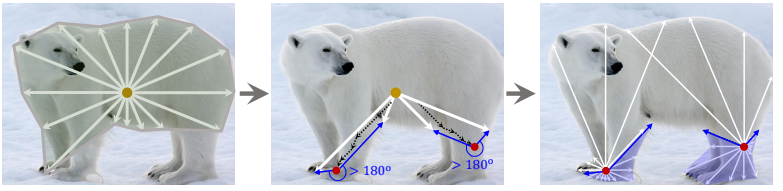
Classification head. The head is responsible for locating instance centroids (the root in Figure 1(a)) by classifying each pixel into k categories, where k is the instance category number. It applies to all feature maps ($P_3 \sim P_7$) from FPN and detects different sizes of instances on different levels of feature maps. We refer to FCOS [19] to construct the head through a few CNN layers.

Centroidness head. Centerness is introduced to suppress the low-quality results in FCOS [19] and PolarMask [19]. However, it may lead to information distortion that hinders the model from learning the correct centroid importance information in instance segmentation tasks. We follow the above issue and design centroidness for improving the reliability of the instance centroids. Concretely, centroidness is formulated as:

$$w_{x,y} = \mathbb{1}_{\{l^*_{x,y} \in mask\}} \frac{\min\{d^1_{x,y}, d^2_{x,y}, ..., d^m_{x,y}\}}{d^c_{x,y} + \min\{d^1_{x,y}, d^2_{x,y}, ..., d^m_{x,y}\}}, \tag{1}$$

where $w_{x,y}$ is the centroid weight at the coordinate of (x,y). $d^c_{x,y}$ is the Euclidean distance between the coordinate of (x,y) and the instance centroid. $\{d^1_{x,y}, d^2_{x,y}, ..., d^m_{x,y}\}$ denotes the set of Euclidean distances between the coordinate (x,y) and all m instance contour points. $\mathbb{1}_{\{l^*_{x,y} \in mask\}}$ is the indicator function, being 1 if the pixel location $(l^*_{x,y})$ within the range of instance masks and 0 otherwise. The centroidness weight spreads around in a Gaussian distribution related to instance shapes. It makes the weight distribution easier to learn, which encourages our model for retaining high-quality instances more effectively.

Regression head. The regression head is used to regress the veins. As depicted in Figure 2(c), our method needs to regress minor veins at the node for refining mask twisty parts. However, different from the root, the node is close to the instance contour, which leads to weak features for regressing minor veins. We follow this issue and introduce a surroundings cross-correlation sensitive (SCCS) module (details can be found in Figure 4) to help regress veins at the node.



(a) coarse contour (b) searching node (c) refining twisty parts by valid minor veins

Figure 5: Visualization of the growth process of minor veins.

The module aims to enhance feature expression through surrounding information. Considering the enhanced pixel plays a dominant role in the vein regression tasks, we concatenate the pixel feature and the cross-correlated surrounding features instead of concatenating all features. Meanwhile, since the different sensitivity of enhanced pixels to surrounding information in different directions, surroundings sensitive prediction is introduced to help focus on the key information. Specifically, as shown in Figure 4, each feature map from FPN is translated in left, top, right, and bottom directions to generate four translation features at first. Then, the corresponding surroundings sensitive maps are predicted (A_1, A_2, A_3, A_4) and cross-correlated $(A_1^{'}, A_2^{'}, A_3^{'}, A_4^{'})$. In the end, combining the translation features and surroundings sensitive maps for enhancing the vein regression at the node.

Algorithm 1 Search Node

Require: The offset maps M; the instance centroid coordinate c; searching deep s;
Ensure: The node coordinate c_n of nth part;

for $i = 0 \rightarrow s$ **do**
  if $i == s-1$ **then**
    $\lambda_n^1 \leftarrow \frac{1}{2}$; $\lambda_n^2 \leftarrow \frac{1}{2}$
  else
    $\lambda_n^1 \leftarrow \frac{1}{2(s-i)-1}$; $\lambda_n^2 \leftarrow \frac{1}{2(s-i)-2}$
  end if
  $l_n \leftarrow M[c[1], c[0]][n]$
  $c_n \leftarrow (\lambda_n^1 l_n + c[0], \lambda_n^1 l_n + c[1])$
  $l_{n+1} \leftarrow M[c_n[1], c_n[0]][n+1]$
  $c_n \leftarrow (\lambda_n^2 l_{n+1} + c_n[0], \lambda_n^2 l_{n+1} + c_n[1])$
  $c \leftarrow c_n$
end for

Vein growth. We introduce a natural and effective representation method (VeinMask) for covering instances masks with various shapes through an appropriate number of points dynamically. It assembles instance contours by simulating the leaf vein growth process.

Concretely, see Figure 5, given an instance, our method first locates the instance centroid (root) through classification and centroidness heads. Then, the distances between the centroid and instance contour in n directions of the polar coordinate are obtained by the regression head. Combining the centroid and the distances to generate n major veins, where a coarse instance contour can be constructed by connecting the vein endpoints in the clockwise direction. It is found that the instance is decomposed into n parts via major veins. Next, all twisty parts are refined by minor veins one by one (see Figure 5). For a specific part,

our method searches the corresponding node under the guidance of adjacent major veins (the details can be referred to Algorithm 1) and determines it is a twisty part if the angle $< \overrightarrow{c_n e^n_{pre}}, \overrightarrow{c_n e^n_{nxt}} >$ is bigger than 180°, where c_n is the node coordinate of nth part. e^n_{pre} and e^n_{nxt} denote the coordinates of nth part adjacent major vein endpoints, respectively. At the location of the twisty part node, the directions in the range of $(\overrightarrow{c_n e^n_{pre}}, \overrightarrow{c_n e^n_{nxt}})$ are treated as minor veins and the vein endpoints are connected in the clockwise direction can be obtained the contour points of this twisty part.

Residual IoU loss. Our method simulates the leaf growth process to assemble instance contours via the veins. To train them effectively, we introduce Residual IoU (RIoU) loss.

Concretely, IoU loss [28] and the corresponding improved versions [18, 30, 31] are designed for supervising the four bounds of the box as a whole. Considering the IoU of irregular-shaped instances is difficult to compute, Polar IoU loss [24] expresses it via the ratio of dense discrete offsets. However, the loss may result in different optimization gradients for the same differences between the predicted and real values, which brings interference to the optimization process. we follow this issue and design RIoU loss:

$$\mathcal{L}_{RIoU} = \frac{\sum_{k=1}^{n} |d_k - d_k^*|}{\sum_{k=1}^{n} d_k^*}, \tag{2}$$

where d_k and d_k^* are the predicted and real values, respectively. n is the direction number of polar coordinate. Rather than computing Polar IoU loss via the ratio of the predicted and real value, we explicitly let the model takes the residual over them as the optimization object directly, which avoids the problem that exists in Polar IoU loss effectively and is easier to learn for the model. Moreover, compared with the logarithm formulation of Polar IoU loss, the derivation of our loss is simpler, which helps improve the back propagation efficiency.

For classification and centroidness heads, we supervise them by cross-entropy loss ($\mathcal{L}_{CE}$) and focal loss ($\mathcal{L}_{FL}$) [11]. The overall loss $\mathcal{L}$ can be formulated as $\mathcal{L} = \mathcal{L}_{RIoU} + \mathcal{L}_{FL} + \mathcal{L}_{CE}$.

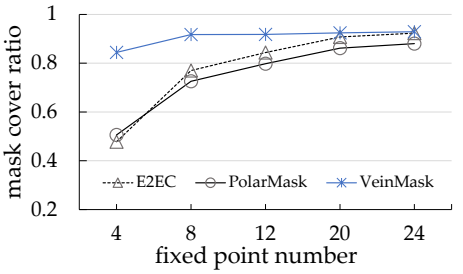
4 Experiments

We show the experimental results on the SBD [6] and COCO *test-dev* [9] datasets in this section. We typically analyze the superiorities of VeinMask over the other contour-based methods on the SBD dataset in detail. Meanwhile, we compare our approach with existing state-of-the-art (SOTA) methods on the two datasets.

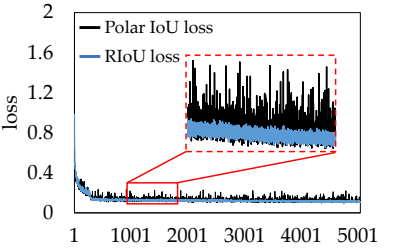
Training details. In ablation study, ResNet-50 [7] pre-trained on ImageNet [4] is adopted as backbone unless otherwise noted. We resize the input into 768×768. The model is trained by the optimizer of stochastic gradient descent (SGD) for 144 epochs on the SBD dataset, where weight decay and momentum are set as 0.0001 and 0.9. The initial learning rate and batch size are 0.01 and 16. In comparison experiments, the images on the COCO *test-dev* dataset are scaled to 800×1333 and padded to 896×1408 with 0.

4.1 Ablation Study

We ablate our approach in Figure 6, Table 1, and Table 2 to verify the effectiveness of VeinMask, centroidness, SCCS, and RIoU loss. The fixed point number is 12 unless otherwise noted. The default model in Table 1 is constructed with the centerness and polar IoU loss proposed in PolarMask [24] and without VeinMask and SCCS.



(a) Mask cover ratio



(b) Loss curve

Figure 6: Ablation experiments on SBD dataset. (a) shows the mask cover ratio upper bound analysis of different methods and different fixed point numbers. (b) demonstrates RIoU loss enjoys more efficient training process than Polar IoU loss.

VeinMask	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	24.4	53.3	19.4	8.4	17.0	29.8
with	**27.5**	**53.7**	**24.3**	**8.4**	**17.7**	**34.6**

centroidness	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	27.5	53.7	24.3	8.4	17.7	34.6
with	**29.1**	**56.6**	**26.1**	**7.4**	**19.0**	**36.9**

SCCS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	27.5	53.7	24.3	8.4	17.7	34.6
with	**29.0**	**56.6**	**25.8**	**7.7**	**19.3**	**36.8**

RIOU Loss	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	27.5	53.7	24.3	8.4	17.7	34.6
with	**28.8**	**55.6**	**26.0**	**6.7**	**18.0**	**37.3**

Table 1: Ablation experiments on the SBD dataset. We report the effectiveness of VeinMask, centroidness, SCCS, and RIoU loss, respectively.

Mask cover ratio. VeinMask is designed to generate an appropriate number of points dynamically to cover various masks precisely. We compare the mask cover ratio upper bound of it with previous representative contour-based methods (PolarMask and E2EC) in different fixed point numbers. Notably, the fixed point number denotes the direction number of the polar coordinate in our method and PolarMask, and the vertices number for E2EC. As depicted in Figure 6(a), benefiting from the advantages of the dynamic fitting process, VeinMask achieves almost 92% mask cover ratio with only eight fixed points, which outperforms the others a lot. Meanwhile, VeinMask is still in a dominant position even when tuning the point number to large. The results verify the VeinMask's superior ability to cover masks.
VeinMask. The effectiveness of VeinMask is studied in Table 1. It is found that refining twisty parts via the minor vein of VeinMask can bring 3.1% improvements in mAP, which verifies the VeinMask's strong ability to segment instance masks with complex shapes.

Centroidness *vs*. centerness. An important design of contour-based approaches is to filter low-quality results via centerness. In this work, we propose centroidness for instance segmentation according to the relation between the instance shape and centroid distribution.

Table 1 studies this design. We report the performance of models that are equipped with centerness and centroidness. It is found that centroidness brings 1.6% mAP for our model, which benefits from the advantage that centroidness ensures the result's reliability in the inference process. The results verify the superior of centroidness over centerness.

SCCS. We design SCCS module to enhance feature expression to help regress veins more accurately. As shown in Table 1, the module brings 1.5% mAP improvements to the baseline. Rather than extracting features of all contour points and concatenating them, SCCS ensures the regression head and the other two heads run in parallel. It avoids multiple stages

deep	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
none	25.9	55.2	21.3	7.8	17.9	32.1
+P7	26.0	55.3	21.3	7.8	17.9	32.2
+P6	28.3	55.8	25.0	7.8	18.0	36.2
+P5	29.3	56.3	26.8	7.8	19.2	37.4
+P4	**29.4**	56.4	26.8	7.8	20.0	37.0
+P3	29.1	56.1	26.3	6.7	19.1	37.4

(a) Deep for refining the twisty part

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
PolarMask [24]	25.9	57.0	20.3	8.0	18.4	32.0
+ RIoU loss	27.8	59.1	22.3	8.8	19.7	34.4
+ centroidness	28.3	59.9	22.6	8.4	19.8	35.0

(b) Generalization of RIoU loss and centroidness

Table 2: Ablation experiments on the SBD dataset. (a) shows growing minor veins on more different-sized feature maps brings more gains except on P3. We embed RIoU loss and centroidness into PolarMask in (b) to show the brought performance gains.

method	N	SBD			COCO		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
PolarMask [24]	4	5.1	21.1	0.1	1.1	6.2	0.0
E2EC [29]	4	2.3	16.4	0.0	2.0	11.5	0.0
VeinMask	4	**21.1**	50.7	15.2	**14.8**	34.2	11.8
PolarMask [24]	8	20.7	52.5	12.9	5.6	14.0	3.2
E2EC [29]	8	26.5	57.6	20.7	17.3	37.4	14.3
VeinMask	8	**28.6**	56.0	25.3	**25.3**	47.5	24.0

(a) Comparisons in less fixed point number

contour-based	N	AP	AP_{50}	AP_{75}
multi-stage				
DeepSnake [15]	128	30.3	-	-
E2EC [29]	128	31.7	52.2	32.8
one-stage				
PolarMask [24]	36	30.4	51.9	31.0
PolarMask++ [25]	36	31.6	54.5	32.2
VeinMask	20	32.4	55.7	32.7

(b) Comparisons with SOTA methods

Table 3: Comparisons with previous **contour-based methods** on the SBD and COCO *test-dev* datasets. N denotes the fixed point number.

serial feature enhancement process in [15, 29], which brings gains with negligible costs.

RIoU loss *vs.* Polar IoU loss. In Table 1 we compare our RIoU loss with Polar IoU loss. We force the model to focus on the residual between the predicted and real values via RIoU loss, a more effective optimization object than Polar IoU loss. It enjoys a simpler derivative formulation and ensures the same gradients for the same residuals. The experimental results in Table 1 show that our RIoU loss brings 1.3% mAP gains, which verifies its superiority over Polar IoU loss. We report the training processes of PolarMask equipped with Polar IoU and RIoU losses respectively in Figure 6(b). RIoU loss enjoys a smoother descending process than Polar IoU loss, which shows the superiority of RIoU loss.

Deep for refining the twisty part. VeinMask helps improve performance by refining twisty parts of instance masks via minor veins (see Table 2(a)). We further explore the performance gains when growing minor veins on feature maps with different sizes. In Table 2(a), we grow the minor veins from P7 to P3, it is found that our strategy brings significant performance gain for large instances, while performs not well for small ones since they are very sensitive to the predicted veins.

Generalization of centroidness and RIoU loss. Centroidness and RIoU loss are embedded into PolarMask to replace the centerness and Polar IoU loss. In Table 2(b), they can bring 2.4% and 1.9% mAP gains, respectively. In particular, centroidness and RIoU loss gain performance significantly for large-scale instances because centroidness helps suppress low-quality results more effectively and RIoU loss makes the model learn more differences between the predicted and real values. The results show their generalization and verify that they can be embedded into other methods seamlessly to gain performance without costs.

(a) w/o VeinMask (b) with VeinMask (c) result in SBD (d) result in COCO

Figure 7: Visualization of some qualitative results on the SBD and COCO *test-dev* datasets.

4.2 Comparisons with Previous Results

In Table 3(a), we compare our method with existing contour-based methods to show superior performance. Specifically, VeinMask outperforms PolarMask [24] 19.7% mAP on the COCO *test-dev* when tuning the fixed point number to 8 and outperforms E2EC [29] 18.8% mAP on the SBD when tuning the fixed point number to 4, which demonstrates the strong ability of VeinMask to segment instance masks in low fixed point number.

Furthermore, we explore the upper performance of VeinMask on the COCO *test-dev* and compare it to existing SOTA approaches. As depicted in Table 3(b), for multi-stage contour-based methods (DeepSnake [15] and E2EC, our method can achieve 32.4% mAP with the one-sixth fixed point number of them. For the one-stage contour-based method (PolarMask), our approach even outperforms it 2% in mAP with almost half the fixed point number. Additionally, we visualize some qualitative results in Figure 7. It can be found in Figure 7(a) and Figure 7(b) that VeinMask helps to refine twisty parts effectively, which benefits from the flexible fitting ability brought by minor veins. Figure 7(c) and Figure 7(d) demonstrate the strong ability of our approach for segmenting complex and dense objects, respectively. The combination of Table 3 and Figure 7 verify the superior performance of VeinMask and the effectiveness of the representative flexible fitting strategy.

5 Discussion and Conclusion

Unifying object detection and instance segmentation problems is a valuable exploration direction. In this work, we follow the intrinsic idea of box detectors to segment instances with center classification and offset regression. We observe leaf vein growth mode and design VeinMask to provide a natural and interesting solution for enhancing the flexibility of contour-based methods. It can generate an appropriate number of points to cover instance masks according to their shapes in the inference process. Meanwhile, we introduce centroidness and RIoU loss for enhancing the model's ability to segment instances without costs. Importantly, they can be used directly in existing state-of-the-art center classification and offset regression-based architectures. The performance can be effectively enhanced by replacing their centerness and loss function with our centroidness and RIoU loss. Additionally, we construct SCCS to help feature expression of the offset regression. It can be embedded into other architectures and brings performance gains with slight computational costs. We hope the idea that combines leaf vein morphology with instance geometric characteristics will inspire future work, and the proposed centroidness, RIoU loss, and SCCS can become basic components of other approaches.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proc. ICCV*, pages 9157–9166, 2019.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proc. CVPR*, pages 6154–6162, 2018.

[3] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Proc. CVPR*, pages 8573–8581, 2020.

[4] Jia Deng, Wei Dong, Richard Socher, Lijia Li, Kai Li, and Feifei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[5] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Location-sensitive visual recognition with cross-iou loss. *arXiv preprint arXiv:2104.04899*, 2021.

[6] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proc. ICCV*, pages 991–998. IEEE, 2011.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, pages 2961–2969, 2017.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755. Springer, 2014.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, pages 2117–2125, 2017.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, pages 2980–2988, 2017.

[12] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proc. CVPR*, pages 5257–5266, 2019.

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proc. CVPR*, pages 8759–8768, 2018.

[14] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. Dance: A deep attentive contour model for efficient instance segmentation. In *Proc. WCACV*, pages 345–354, 2021.

[15] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proc. CVPR*, pages 8533–8542, 2020.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, pages 779–788, 2016.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015.

[18] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proc. CVPR*, pages 658–666, 2019.

[19] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proc. ICCV*, pages 9627–9636, 2019.

[20] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Proc. ECCV*, pages 282–298. Springer, 2020.

[21] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *Proc. ECCV*, pages 649–665. Springer, 2020.

[22] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *NeurIPS*, 33:17721–17732, 2020.

[23] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *Proc. ECCV*, pages 527–544. Springer, 2020.

[24] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proc. CVPR*, pages 12193–12202, 2020.

[25] Enze Xie, Wenhai Wang, Mingyu Ding, Ruimao Zhang, and Ping Luo. Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5385–5400, 2022.

[26] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin, and Han Hu. Dense reppoints: Representing visual objects with dense point sets. In *Proc. ECCV*, pages 227–244. Springer, 2020.

[27] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou. Embedmask: Embedding coupling for instance segmentation. In *Proc. IJCAI*, pages 1266–1273, 2021.

[28] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proc. ACM MM*, pages 516–520, 2016.

[29] Tao Zhang, Shiqing Wei, and Shunping Ji. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In *Proc. CVPR*, pages 4443–4452, 2022.

[30] Yi-Fan Zhang, Weiqiang Ren, Zhang Zhang, Zhen Jia, Liang Wang, and Tieniu Tan. Focal and efficient iou loss for accurate bounding box regression. *Neurocomputing*, 506:146–157, 2022.

[31] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proc. AAAI*, volume 34, pages 12993–13000, 2020.