frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 10 July 2025
DOI 10.3389/fpls.2025.1378958

OPEN ACCESS

EDITED BY
Zhenghong Yu,
Guangdong Polytechnic of Science and Technology, China

REVIEWED BY
Chao Ni,
Nanjing Forestry University, China
Zejun Zhang,
Zhejiang Normal University, China

*CORRESPONDENCE
Bin Wen
ynwenbin@163.com

RECEIVED 30 January 2024
ACCEPTED 11 June 2025
PUBLISHED 10 July 2025
CORRECTED 23 July 2025

CITATION
Deng H, Wen B, Gu C and Fan Y (2025) GrotUNet: a novel leaf segmentation method. *Front. Plant Sci.* 16:1378958. doi: 10.3389/fpls.2025.1378958

GrotUNet: a novel leaf segmentation method

Hongfei Deng[1,2], Bin Wen[1,3]*, Cheng Gu[3] and Yingjie Fan[3]

[1]Key Laboratory of Ethnic Education Informatization, Yunnan Normal University, Kunming, China, [2]School of Information Technology Industry, Yunnan Vocational Institute of Energy Technology, Qujing, China, [3]School of Information Science, Yunnan Normal University, Kunming, China

In the field of biology, the current leaf segmentation method still has problems such as missed inspections and duplication in the number of large, dense, mutual obstruction and vague division tasks. The reason for the above is that image semantic extraction is not satisfactory and semantic parsing is still insufficient. To address the above problems, this paper proposes GrotUNet, a novel leaf segmentation method that can be trained end-to-end. The algorithm is reconstructed in three aspects: semantic feature coding, hopping connectivity, and multiscale upsampling fusion. The semantic coding structure consists of GRblock, WGRblock, and OTblock modules. The former two make full use of the design ideas of GoogLeNet parallel branching and Resnet residual connectivity, while the latter further mines the fine-grained semantic information distributed in the feature space on the feature map after extraction by the WGRblock module to make the feature expression richer. Unlike UNet++ dense connectivity, jump connection reconstruction only uses 1×1 convolution for feature fusion of feature maps from different network hierarchies to enrich the semantic information at each location in the space. The multi-scale upsampling fusion design mechanism incorporates higher-order feature maps into each shallow decoding sub-network, effectively mitigating the loss of semantic parsing information of feature maps. In this paper, the method is fully demonstrated on CVPPP, KOMATSUNA and MSU-PID datasets. The experimental results show that GrotUNet segmentation outperforms existential UNet, ResUNet, UNet++, Perspective + UNet and other methods. Compared with UNet++, GrotUNet improves the key evaluation metrics (SBD) by 0.57%, 0.30%, and 0.27%, respectively.

KEYWORDS

instance segmentation, feature coding, jump connection, multi-scale fusion, GoogLeNet

1 Introduction

Instance segmentation has been the most challenging task in the field of computer vision, and its techniques are widely used in the fields of intelligent driving, intelligent medical imaging, remote sensing images, and biological phenotyping. In the field of biology, the extraction and analysis of plant phenotypic features is a meaningful research

work for describing organisms, which is very useful for agricultural decision-making and plant breeding industry, and helps to improve varieties and genes for plant industry, increase yield and reduce resource consumption. Currently, most plant phenotyping work is still done under field conditions by manual marking, these methods are time consuming, tedious and error prone, exploring new methods is the best way out of the dilemma. Leaf segmentation is a powerful tool for plant phenotyping. Plant leaves are characterised by various features, mainly including leaf area, leaf shape, leaf number, leaf texture, petiole and so on. Plant modal leaves have dense, mutually occluding and overlapping, large number and complex petiole joints, which makes it difficult for traditional segmentation methods to perform optimally.

The challenges of leaf segmentation and counting include two main aspects: on the one hand, challenges such as texture variations inherent to plant leaves, variations in leaf shape and size, overlap between leaves, and difficulty in distinguishing petioles. On the other hand, challenges such as variations in ambient brightness, shadows and blurring caused by wind shaking. Compared to the leaf blade, the petiole has different shapes, is very small, and exists in a very small localised area, making it difficult for some existing segmentation methods to achieve accurate segmentation. The reason is that traditional segmentation methods have insufficient extraction of fine-grained semantic features and insufficient semantic parsing reduction in the local region, which has a significant impact on segmentation performance.

In terms of feature encoding, thanks to ResNet the residual linking mechanism allows both the network layers to be deep and prevents the gradient from vanishing, evolving multiple series of architectures (He et al., 2016). Most existing segmentation architectures use it as the feature encoding extraction backbone network. GoogLeNet (Yuan et al., 2022) adopts the parallel branching idea, which allows parallel multi-branch extraction and fusion of the same input feature map, with a view to mining richer semantic information (Szegedy et al., 2015). The Outlooker neighborhood attention mechanism of the VOLO model aims at further refining the extraction of fine-grained semantics, while the Transformer can aggregate local region feature encoding to generate global contextual semantic information (Khan et al., 2022; Yuan et al., 2022). In terms of semantic parsing, UNet++ realizes the reconstruction and retention of local and global information by reconfiguring the sliding connections so that each layer carries as much local and global information as possible, and each layer is interconnected with each other, and finally shared to the last layer (Zhou et al., 2019). UNet3+ fuses the encoding module's output feature maps with multi-scale downsampling splicing into different sub-networks of the decoding backbone (Huang et al., 2020). At the same time, the decoding high-level semantics are fused into shallow sub-networks by multi-scale up-sampling, which preserves most of the semantic information to a certain extent. Bert et al (Bert De et al., 2017). proposed discriminative loss function, which uses clustering mainly in the embedding space to recover the test instances to perform segmentation. Based on the inspirations generated by the above methods, we reconstruct the design in the feature encoding backbone network, hopping connection, and decoding backbone network, and propose a new method GrotUNet for plant leaf segmentation. The main contributions are as follows:

1. Combining GoogLeNet parallel branching with the ResNet residual join idea, the feature encoding modules GRblock and WGRblock are designed. The former is used for shallow network feature extraction and the latter for high-level feature extraction.
2. The Outlooker attention and Transformer self-attention mechanisms are introduced to further mine the fine-grained semantic information of the feature maps output from the WGRblock module, the Outlooker being used to further refine the extraction of local regions of the feature maps, while the Transformer is used to gather the attention of the nearest neighbours to generate the global contextual semantic information.
3. Reconfiguring the sliding links so that the shallow coding module outputs feature maps and the higher layer feature maps are spliced by channel after upsampling and then fused across channel features using 1×1 convolution, helping to enrich the semantic information at each location in space.
4. The decoding backbone network adopts a multi-scale up-sampling fusion design mechanism, which fuses the multi-scale up-sampling of the feature maps output from the high-level network into the shallow subnetwork to mitigate the information loss in the semantic parsing process.
5. This paper conducts comprehensive empirical studies on the CVPPP, KOMATSUNA, and MSU-PID datasets. The experimental results demonstrate that GrotUNet outperforms state-of-the-art segmentation algorithms.

The paper is organized as follows: section 2 introduces the work related to instance segmentation. Section 3 gives a detailed description of the improved GrotUNet algorithm. Section 4 provides an experimental validation of the proposed algorithm, describing the dataset, the evaluation metrics and analyzing and discussing the experimental results in detail. Section 5 gives the concluding remarks and proposes the future direction of development.

2 Related work

Instance segmentation is mainly categorized into candidate box extraction based and pixel classification based instance segmentation methods. He He et al (He et al., 2017). proposed the Mask R-CNN algorithm by adding a mask sub-network to the Faster R-CNN. The method connects the mask with candidate frame extraction learning and uses RoIAlign to replace RoIPooling to reduce the loss of semantic information. PANet improves the

structure of the feature pyramid in the backbone network on top of the Mask R-CNN, introduces a new bottom-up pathway on the FPN, and performs aggregation between the pathways (Liu et al., 2018). DetNet introduces null convolution into the backbone framework of the network and proposes to re-train the backbone network for detection and segmentation tasks to achieve feature expressiveness and resolution (Li et al., 2018). PointRend optimizes the object edges for the up-sampling operation that get better boundary masks (Kirillov et al., 2020). RefineMask fuses more fine-grained information step-by-step in a multi-stage approach, and finally optimizes Mask R-CNN to generate rough mask edges using semantic segmentation information and edge profile information to output accurate boundary information (Zhang et al., 2021). Xue et al. (Sheng et al., 2023) improved YOLOv7 by optimizing the model structure and parameters, and then combined migration learning and optimized data enhancement methods to achieve good performance in detecting fine cigarette impurities in the stems.These methods are benchmarks for instance segmentation tasks combining target detection with target mask estimation. However, these methods become quite complex in method tuning and segmentation performance is limited when irregularly shaped targets are learned for detection. SSAP learns the pixel pair affinity pyramid. The probability of two pixels belonging to the same instance, and generates instances sequentially through cascaded graph segmentation (Gao et al., 2019). These methods generate instance masks by categorising pixels into any number of object instances in the image.

Leaf instance segmentation methods based on plant phenotypes. Romera et al. (Romera-Paredes and Torr, 2016) used LSTM network (Van Houdt et al., 2020) to train an end-to-end instance segmentation and counting network. Ren et al. (Mengye and Richard, 2017) proposed recurrent neural network combined with candidate box extraction, which showed good segmentation performance on plant leaf CVPPP dataset. Li et al. (Xingyou et al., 2024) generated pseudo defective candy images based on StyleGAN2 to enhance the negative sample data, and then background separated the color domain features of defective candies to solve the interference of the imbalance between intact and defective candy data on the detection performance. Deep Coloring simplified instance segmentation into a semantic segmentation while class labels are used for non-adjacent objects and then analyze the connected components to retrieve the instances (Kulikov et al., 2018). Victor K et al. (Kulikov and Lempitsky, 2020) proposed Harmonic algorithm which describes each object instance by using the expectation of a finite number of sinusoids and adjusts it to a specific object size and density using phase and frequency tuning. Tran et al. (Tuan et al., 2021) proposed an end-to-end reinforcement learning-based end reinforcement learning instance segmentation algorithm ColorRL. Sandesh B et al. (Bhagat et al., 2022) proposed a plant leaf segmentation algorithm Eff-UNet++. The algorithm not only adopts the lightweight Efficient-net network as the feature extraction backbone network, but also reconstructs the sliding connection part of UNet++, so that the number of parameters and the computation amount are greatly reduced. In the decoding backbone network, the high-dimensional and low-dimensional features are spliced and fused to obtain the boundary information of the object effectively. Eff-UNet++ method shows excellent performance in the dataset of plant phenotype feature segmentation.

In studies related to plant leaf segmentation, De Brabandere et al. (Bert De et al., 2017). used a discriminative loss function consisting of two parts: one part pushes the embedding means of different objects farther away from each other, and the other part pulls the embedded pixels of the same object closer to their means. The main idea is to embed the image pixels into the hidden high dimensional space, the pixels belonging to the same instance are close to each other in the space, while the pixels of different instances of the object will be embedded into different spaces, and then subsequently use clustering algorithms to generate separate instances, which is the basis of the study in this paper. In recent years, UNet architecture is widely used for segmentation tasks (Ronneberger et al., 2015). Diakogiannis et al. (Diakogiannis et al., 2020) proposed a ResUNet architecture by replacing the UNet feature extraction backbone network using a Resnet network to achieve better segmentation performance on remote sensing images. In the existing research instance segmentation end-to-end model still has more room for development. DeepLab v3+ added an effective decoder module to DeepLab v3 to recover object boundaries and achieved good performance (Chen et al., 2018). Zhou et al. (Zhou et al., 2019) proposed UNet++,interconnecting intermediate outputs between each layer with each other by means of thick connections, each module interacts with each other, and design a supervision mechanism to achieve better performance in medical image analysis. The disadvantage is that the introduction of dense connections leads to a drastic increase in the number of parameters in the model architecture, which consumes a lot of computational resources. Eff-UNet++ reduces the number of dense connections on the basis of UNet++, and fuses the high-level output feature maps into the decoding sub-networks of each layer by gradually up-sampling them, which greatly reduces the number of parameters.

Plant leaf contours, colors, and other features are very similar, coupled with the presence of occlusion and overlap between leaves, leading to tricky detection of leaf overlap and petiole regions by traditional methods. The analysis found that the network architecture of these methods causes semantic loss for feature map downsampling and upsampling. Simultaneously, there are limitations in the sensory field and insufficient ability to capture information around the spatial location of the feature map. Plant petiole features exist in a small localized area, which makes it difficult to extract such fine-grained semantic information. In order to overcome the above difficulties, this paper reconstructs the feature extraction backbone network, sliding connection, decodes the backbone network, and proposes a new plant leaf segmentation method.

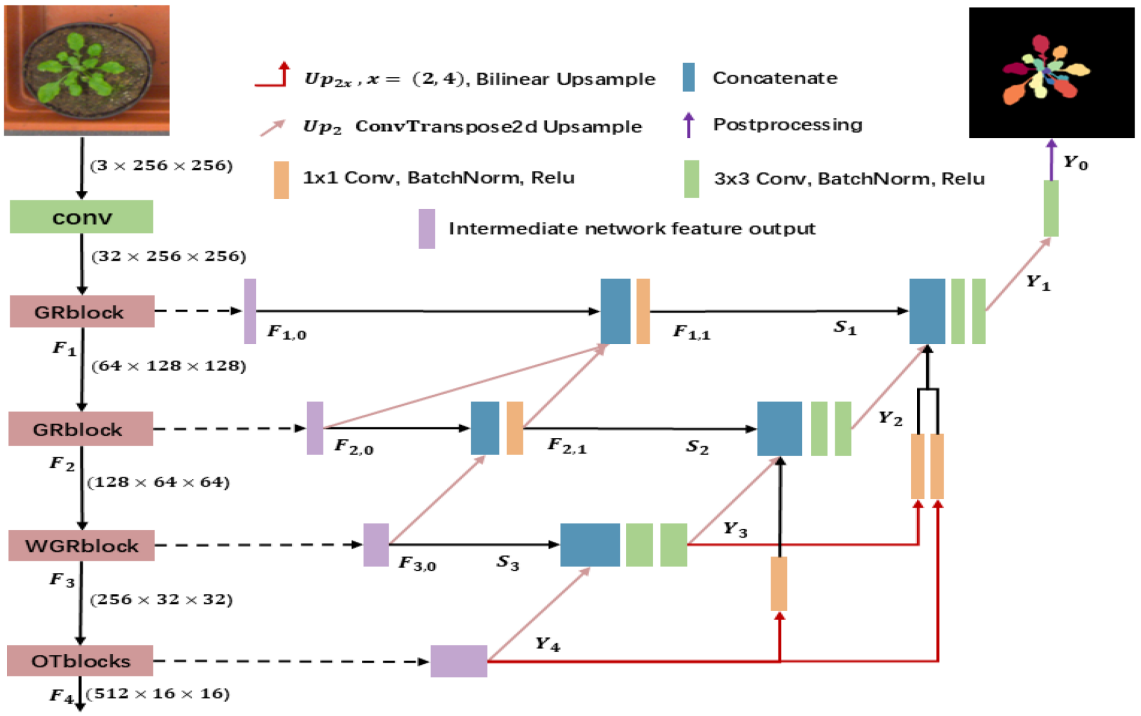

FIGURE 1
Overview of the framework.

3 Method

The method proposed in this paper is shown in Figure 1, with the reconstructed hybrid feature coding modules GRblock, WGRblock, and OTblock on the left part, the redesigned sliding connection in the middle part, and the multiscale upsampling fusion design mechanism represented in the right part. Next, the role of each part will be elaborated in detail.

3.1 GROT hybrid feature encoding module

The GrotUNet feature extraction backbone network consists of GRblock, WGRblock, and OTblock, as shown in Figure 2. Currently, the encoding backbone network of most mainstream segmentation methods is mainly based on the ResNet family of architectures. ResNet increases the depth of the network by stacking the residuals quickly, but its effective receptive field may not be as large as theoretically (He et al., 2016). Multiple Inception modules in GoogLeNet are able to capture richer feature information by applying convolution kernels of different sizes and pooling operations in parallel, but this can make the network structure relatively complex and seriously consume resources (Szegedy et al., 2015). The GRblock and WGRblock modules incorporate the parallel branching design ideas of residual block and Inception block, which can not only encode and extract different ranges of spatial feature information to enrich the feature expression, but also prevent the problem of gradient disappearance, so that the network will be more stable in the training process. The GRblock network structure is relatively simple and is mainly used to extract shallow feature maps, while the WGRblock structure is more complex and is mainly used to extract higher-order feature maps. The OTblock module is used to further extract higher-order feature maps, aiming to make fine-grained semantics further characterized. Next, the design of each coding module is described in detail.

3.1.1 GRblock encoding module

GRblock is mainly used for feature extraction in low-dimensional space, and its first half uses the idea of parallel branching and the second half uses residual join, as shown in Figure 2a. To reduce the loss of semantic information, the module reduces the feature map size using maximum pooling and convolutional parallel two-branch downsampling with a convolutional kernel of 2. In addition, a large number of asymmetric convolutions are used in the hidden space for reducing the network computation and the number of parameters, and asymmetric convolutions are also used in the subsequent coding module. With the fast encoding and the increase of network layers, the gradient backpropagation may be decayed at each layer, which is very likely to cause the gradient vanishing problem. Residual connection is a method that can effectively solve the gradient vanishing problem, different from ResNet's residual connection, this paper adopts the cross-residual connection to prevent the gradient vanishing problem, as shown in

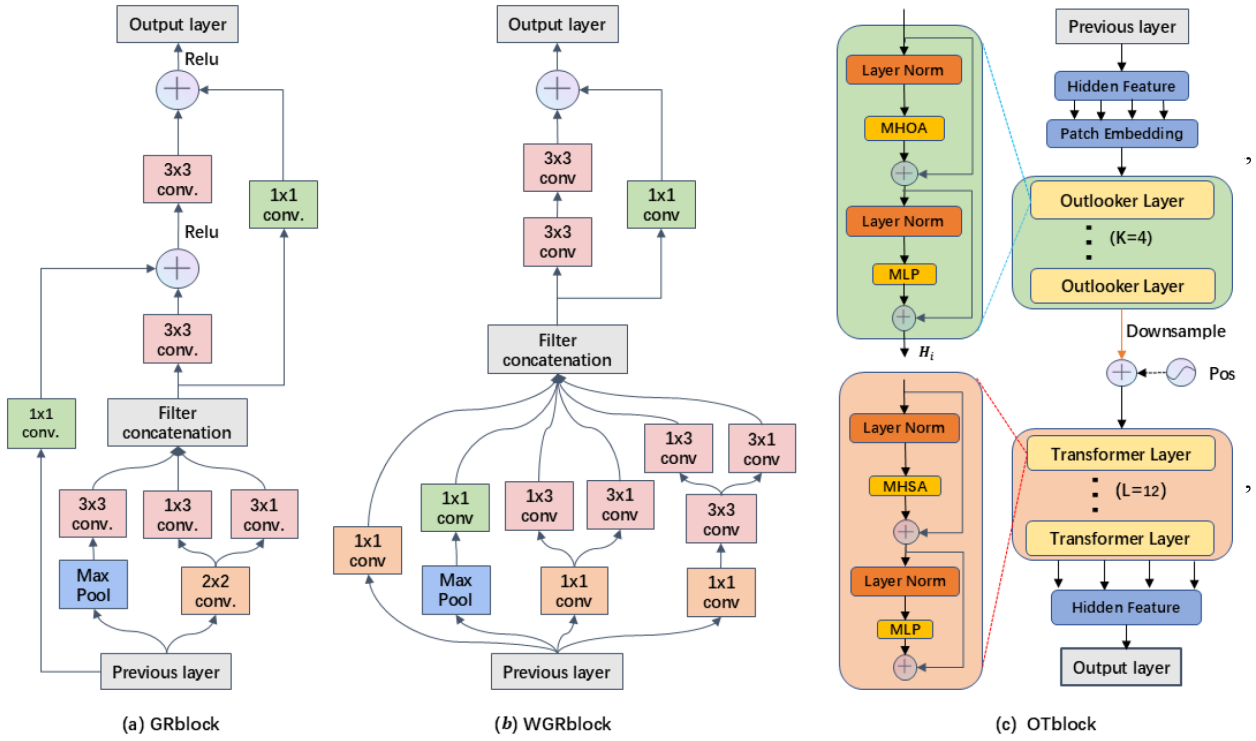

FIGURE 2
Feature extraction coding module.

Figure 2a. Suppose the input of the module is X. The GRblock module definition Equations 1–3 is shown:

$$\mathcal{H} = [\boldsymbol{C}_{33}(\boldsymbol{Maxpool}(\boldsymbol{X})), \boldsymbol{C}_{13}(\mathcal{C}_{11}^{1}(\boldsymbol{X})), \boldsymbol{C}_{31}(\boldsymbol{C}_{11}^{1}(\boldsymbol{X}))] \quad (1)$$

$$\mathcal{M} = \boldsymbol{Relu}(\boldsymbol{C}_{33}(\mathcal{H}) + \boldsymbol{C}_{11}^{1}(\mathcal{H})) \quad (2)$$

$$\mathcal{F}_i = \boldsymbol{Relu}(\boldsymbol{C}_{33}(\mathcal{M}) + \boldsymbol{C}_{11}(\mathcal{H})) \quad (3)$$

Where *Maxpool* and $\mathcal{C}_{11}^{1}$ represent max pooling and convolutional downsampling, respectively. C_{XY} represents a convolution operation with a kernel of $X \times Y$. *Relu* denotes the activation function.

3.1.2 WGRblock encoding module

The WGRblock module structure is designed as shown in Figure 2b. The module uses the GoogLeNet parallel branching approach for feature extraction in different scale ranges of the input feature map, which consists of one maximum pooling branch, three *stride* = 2 convolutional downsampling branches, and convolutional operations in tandem with each score. Maximum pooling preserves the leaf instance edge semantic information, while convolutional downsampling carries more local semantic information. Each branch, after the corresponding operation, splices and fuses the feature maps in channel direction to pass into the residual block for further feature extraction. Assuming that one of the intermediate inputs is $\mathcal{F}_i$, The definition of WGRblock is shown as Equations 4–7:

$$\boldsymbol{s}_1 = \boldsymbol{C}_{11}^{1}(\mathcal{F}_i) \quad (4)$$

$$\boldsymbol{s}_2 = \boldsymbol{C}_{33}\left(\boldsymbol{C}_{11}^{1}(\mathcal{F}_i)\right) \quad (5)$$

$$\boldsymbol{g} = [\boldsymbol{C}_{11}^{1}(\mathcal{F}_i), \boldsymbol{C}_{11}(\mathcal{M}(\mathcal{F}_i)), \boldsymbol{C}_{13}(\boldsymbol{s}_1), \boldsymbol{C}_{31}(\boldsymbol{s}_1), \boldsymbol{C}_{13}(\boldsymbol{s}_2), \boldsymbol{C}_{31}(\boldsymbol{s}_2)] \quad (6)$$

$$\mathcal{F}_{i-1} = \boldsymbol{Relu}(\boldsymbol{C}_{33}(\boldsymbol{C}_{33}(\boldsymbol{g})) + \boldsymbol{C}_{11}(\boldsymbol{g})) \quad (7)$$

Where C_{11}^{1}, $\mathcal{M}$ represent convolution kernel 1×1, sliding to 2 convolution and max pooling downsampling operations, respectively. C_{XY} represents a convolution operation with a kernel of $X \times Y$. $[\cdot]$ represents the splicing fusion operation by channel direction. Relu represents the activation function.

3.1.3 OTblock encoding module

OTblock module consists of multiple Outlooker and Transformer attention layers, as shown in Figure 2c. The Outlooker neighborhood attention mechanism originates from VOLO, which was initially created to make each spatial location on the image sufficiently representative, and is designed to aggregate the attention weights of each neighboring location in the generative space, to further refine the local features. Transformer has a powerful ability to encode contextual information, and can aggregate local spatial semantic information to generate global

contextual information. OTblock uses 4 Outlooker neighborhood attention layers in combination with 12 Transformer self-attention layers to further mine each location semantic information in the higher-order feature space, which is then aggregated to generate globally richer semantic information. Assume h_{k-1} is the input of a layer in the middle of Outlooker, and s_p^i is the intermediate data token obtained by downsampling after Outlooker extracts the neighborhood weights. The OTblock encoding definition is as shown in Equations 8–12:

$$\boldsymbol{h_k^{'}} = \boldsymbol{MHOA}(\boldsymbol{LN}(\boldsymbol{h_{k-1}})) + \boldsymbol{h_{k-1}} \tag{8}$$

$$\boldsymbol{h_k} = \boldsymbol{MLP}(\boldsymbol{LN}(\boldsymbol{h_k^{'}})) + \boldsymbol{h_k^{'}} \tag{9}$$

$$G_0 = [\boldsymbol{s_p^1 w}; \boldsymbol{s_p^2 w}; \ldots; \boldsymbol{s_p^n w};] + W_{pos} \tag{10}$$

$$G_l^{'} = \boldsymbol{MHSA}(\boldsymbol{LN}(G_{l-1})) + G_{l-1} \tag{11}$$

$$G_l = \boldsymbol{MLP}(\mathcal{L}N(G_l^{'})) + G_l^{'} \tag{12}$$

Where $\mathcal{W} \in \mathbb{R}^{(p)^2+\mathcal{D}}$ is the projection of the patch embedding. $\mathcal{W}_{pos} \in \mathbb{R}^{\mathcal{N}\times\mathcal{D}}$ is the positional embedding vector, h_k denotes the output of the k-th layer Outlooker, and $\mathcal{G}_l$ denotes the output of the l-th layer Transformer. *MHOA*、 *MHSA*、 *MLP*, and *LN* stand for Multi-Headed Outlooker Attention, Multi-Headed Self-Attention, Multi-Layer Perceptual Machine, and Layer Normalization Operation, respectively.

3.2 Reconstructing Skip Connections (R-Skip)

Compared with the UNet++ sliding connection, the sliding connection reconstructed by the method in this paper reduces a large number of intermediate nodes and retains at most one node per layer, as shown in the middle part of Figure 1. The role of this node is mainly to aggregate the output feature maps from the current layer coding block with the feature maps output from all coding blocks and nodes of the higher layer. The purpose of using 1×1 convolution in the node is to realize cross-channel information aggregation retaining the semantic independence of each spatial location, providing rich spatial semantic features for the decoding backbone network. The definition of reconstructing the sliding connection is as shown in Equations 13–15:

$$S_3 = \mathcal{F}_{3,0} \tag{13}$$

$$S_2 = \boldsymbol{C_{11}}([\mathcal{F}_{2,0}, U_2(S_3)]) \tag{14}$$

$$S_1 = \boldsymbol{C_{11}}([\mathcal{F}_{1,0}, U_2(\mathcal{F}_{2,0}), U_2(S_2)]) \tag{15}$$

Where C_{11} denotes the convolution kernel for 1×1 convolution operation, U_2 represents a bilinear interpolation operation with an upsampling factor of 2. S_i denotes the output of reconstructed sliding connection.

3.3 Multi-scale upsampling fusion decoder

In the decoding stage, the traditional method of recovering semantic information by layer-by-layer up-sampling will cause part of the semantic information to be lost, resulting in limited segmentation performance enhancement. UNet3+ designs a multi-scale up-sampling feature map fusion mechanism in the decoding backbone network, which fuses feature maps at different scales together through bilinear interpolation up-sampling splicing and aims to alleviate the loss of semantic information in the process of semantic parsing. In this paper, we reduce the number of multiscale upsampling connections on the basis of the UNet3+ decoding design, as shown in the decoding section on the right side of Figure 1. Assume that the output of the encoding is $\mathcal{F}_4$, $\mathcal{Y}_4 = \mathcal{F}_4$. The decoding process is defined as shown in Equations 16–19:

$$Y_3 = \boldsymbol{C_{33}}(\mathcal{U}_2(Y_4)) \tag{16}$$

$$Y_2 = \boldsymbol{C_{33}}([U_2(Y_3), U_4(\boldsymbol{C_{11}}(Y_4))]) \tag{17}$$

$$Y_1 = \boldsymbol{C_{33}}([U_2(Y_2), U_4(\boldsymbol{C_{11}}(Y_3)), U_8(\boldsymbol{C_{11}}(Y_4))]) \tag{18}$$

$$Y_0 = \boldsymbol{C_{33}}(U_2(Y_1)) \tag{19}$$

Where C_{XY} denotes the operation on the corresponding convolution. U_d represents a bilinear interpolation operation with an upsampling factor of d. $[\cdot]$ stands for splicing operation by channel direction.

3.4 Loss functions

The discriminative loss function performs well in the field of leaf segmentation and is frequently used in many segmentation models (Bert De et al., 2017). This loss function is defined as shown in Equations 20–23:

$$\mathcal{L}_{var} = \frac{\mathbf{1}}{C}\sum_{\mathbf{c}=\mathbf{1}}^{\mathbf{1}}\frac{\mathbf{1}}{N_c}\sum_{i=\mathbf{1}}^{N_c}[\|\boldsymbol{\mu_c} - \boldsymbol{x_i}\| - \boldsymbol{\delta_v}]_+^2 \tag{20}$$

$$\mathcal{L}_{\mathbf{dist}} = \frac{\mathbf{1}}{C(C-\mathbf{1})}\sum_{\mathbf{c}_A=\mathbf{1}}^{C}\sum_{\mathbf{c}_B=\mathbf{1}}^{C}[\mathbf{2}\boldsymbol{\delta}_{\mathbf{d}} - \|\boldsymbol{\mu}_{\mathbf{c}A} - \boldsymbol{\mu}_{\mathbf{c}B}\|]_+^2, \ (\mathbf{c}_A \neq \mathbf{c}_B) \tag{21}$$

$$\mathcal{L}_{reg} = \frac{\mathbf{1}}{C}\sum_{c=\mathbf{1}}^{C}\|\boldsymbol{\mu_c}\| \tag{22}$$

$$\mathcal{L} = \boldsymbol{\alpha}\cdot\mathcal{L}_{var} + \boldsymbol{\beta}\cdot\mathcal{L}_{dist} + \boldsymbol{\gamma}\cdot\mathcal{L}_{reg} \tag{23}$$

Where $\mathcal{C}$ denotes the number of instances of the real labeled image, $\mathcal{N}_c$ denotes the number of pixels in a particular instance $\mathcal{C}$, x_i denotes the i-th pixel in the instance that generates the embedding

vector, and μ_c is the mean vector of the real labeled instances $\mathcal{C}$, which represents the clustering center. $\|\cdot\|$ is the $\mathcal{L}_1$ or $\mathcal{L}_2$ distance, which represents the canonical term. δ_v and δ_v represent the variance and distance of the margins, respectively. The discriminative loss function aims to bring the embedding vectors of the pixels inside the same instance as close as possible to the center of the mean value of that instance in the mapping space. The mean vectors of different instances are to be as far away from that mean center as possible.

4 Experiments and analysis

This section will detail the experimental design and analysis. Firstly, the different characteristics of the three datasets are briefly introduced. Secondly, the evaluation metrics for instance segmentation leaf segmentation are presented. Then, some details of the algorithm training configuration are presented. Finally, the performance of the proposed method is evaluated and compared with state-of-the-art methods.

4.1 Dataset and evaluation metrics

4.1.1 Dataset

In the process of experimental demonstration, three datasets, CVPPP, KOMATSUNA, and MSU-PID, are selected to verify the effectiveness and segmentation performance of GrotUNet. Next, the reasons and circumstances of data set selection are described in detail.

Due to the small data volume of the CVPPP A1 dataset, the blade contour is clear and the labeled file contour is delicate, which can effectively test the performance of the algorithm (Minervini et al., 2016). Therefore, it is often used as a benchmark evaluation dataset for mainstream instance segmentation methods.A1 contains a total of 161 leaf images, 128 in the training set and 33 in the test set. In order to better evaluate the performance of the algorithm, in the experiments, the training set A1 is divided into 85% for training and 15% for validation, i.e., 108 sheets are used for model training and 20 sheets are used for evaluating the qualitative test.

The KOMATSUNA leaf dataset acquisition was done at 4-hourly intervals under an ambient condition of illumination of 2400 lua, temperature of 30°C, and humidity of 30%, with a total of 900 images (Uchiyama et al., 2017). The moderate data volume of KOMATSUNA, along with the low number of leaves per image, facilitates the observation of segmentation of details such as petiole and helps in the validation of the model.The KOMATSUNA dataset is divided into 80% training dataset and 20% testing dataset, i.e., 720 images for training the model and 180 images for evaluation testing.

MSU-PID is the first multimodal plant image database, which contains two kinds of plants, Arabidopsis and Bean (Cruz et al., 2016). The Arabidopsis data contains four modalities, 2160 RGB modal images and 576 labeled images. The leaf overlap of the Arabidopsis data is blurred, which is difficult to distinguish, and it is more testing for the performance of the model. In the experiment, the Arabidopsis data was preprocessed to collect 576 source data that corresponded one-to-one with the labeled images, which were divided into 80% for the training set and 20% for the testing set, i.e., 460 for training and 116 for evaluation testing.

The method proposed in this paper uses data augmentation techniques to expand the training set prior to training, including random cropping, random up and down flipping, and random left and right flipping.

4.1.2 Evaluation metrics

For the evaluation metrics to evaluate the model segmentation performance, *FBD*, *SBD* are used. the evaluation metrics for the number of instances, *DiC*, $|DiC|$ are used, and the details of each metric are introduced as follows:

Foreground Background Dice (FBD) is the foreground mask dice coefficient (Scharr et al., 2016). It mainly measures the degree of overlap between the real labeling P^{gt} and the background binary segmentation mask of the algorithm's prediction result P^{pre}. It is used to evaluate the ability of the algorithm to recognize the target from the background and perform binary segmentation. FBD is defined as shown in (Diakogiannis et al., 2020):

$$FBD(\%) = \frac{2|P^{gt} \cap P^{pre}|}{|P^{gt}| + |P^{pre}|} \tag{24}$$

Symmetric Best Dice (SBD) denotes the average Dice between all the instances (Scharr et al., 2016). Each predicted label produces dice with the real label and then averages them to estimate the average instance segmentation accuracy. BD is defined as follows:

$$BD(L^a, L^b) = \frac{1}{M}\sum_{i=1}^{M} \max_{1\le j\le N} \frac{2\left|L_i^a \cap L_j^b\right|}{\left|L_i^a\right| + \left|L_j^b\right|} \tag{25}$$

$|\cdot|$ denotes the number of pixels. L_i^a $(1 \le i \le M)$和$L_j^b(1 \le j \le N)$ belong to the segmentation sets L^a, L^b respectively.

The SBD of the true labeled set L^{gt} and the predicted labeled set L^{pre} is defined as follows:

$$SBD(L^{gt}, L^{pre}) = \min\left\{BD(L^{gt}, L^{pre}),\ BD(L^{pre}, L^{gt})\right\} \tag{26}$$

Difference in Count (DiC) represents a measure of the difference between the predicted number of instances and the true number of instances (Scharr et al., 2016). $|DiC|$ is the absolute value of DiC. DiC is defined as follows:

$$DiC = \#L^{pre} - \#L^{gt} \tag{27}$$

4.2 Experimental details

The experimental demonstration is mainly based on the deep learning framework PyTorch, and the specific environment configuration and parameter settings are shown in Table 1. The parameter α, β, γ value settings in the loss function are consistent

TABLE 1 Environment configuration and parameter configuration during experiment implementation.

Experimental setting	Configurations	Parameter setting	Configurations
Operating system	Ubuntu20.04	Batch size	16
CPU	12 vCPU Intel(R) Xeon(R) Platinum 8352V CPU @ 2.10GHz	epochs	200
GPU	vGPU-32GB(32GB) * 1	Ir	1-e3
CUDA Versions	CUDA 11.3	Weight decay	5-e2
Python Edition	Python 3.8	α	1
Deep Learning framework	PyTorch	β	1
Torch versions	1.10.0	γ	0.001

with the study of Bert et al. (Bert De et al., 2017). The optimizer selects AdamW, and the weight decay is 0.05. The initial value of the learning rate is set to 0.001, and the decay factor is 0.1. In order to verify the validity and generality of the model the proposed method in this paper, the hyperparameters are configured identically during the training of the three datasets, CVPPP, KOMATSUNA and MSU-PID. The image sizes of the model inputs are $256 \times 256 \times 3$, $256 \times 256 \times 3$, and $128 \times 128 \times 3$, respectively.

4.3 Experimental analysis and discussion

The method in this paper evaluates the segmentation performance through two dimensions: the intuitive visual perception of the visualization effect and the instance segmentation evaluation metrics. The training and test sets were kept constant during the experimental implementation and were trained and tested independently at CVPPP, KOMATSUNA and MSU-PID using the same parameter settings.

4.3.1 Comparison of state-of-the-art methods

To verify the segmentation performance of the GrotUNet model on plant leaves, this paper carries out experimental comparisons using six state-of-the-art segmentation methods, namely UNet (Ronneberger et al., 2015), ResUNet (Diakogiannis et al., 2020), UNet++ (Zhou et al., 2019), DeepLab V3 (Chen et al., 2018), DSNet (Guo et al., 2024), and Perspective + UNet (Hu et al., 2024). All methods keep the same parameter settings and loss functions during the experiment. The results of the evaluation of this paper's methods on CVPPP, KOMATSUN and MSU-PID

TABLE 2 Comparison results with state-of-the-art methods on the CVPPP dataset.

Method	Flops(G)	Parms(M)	FPS	FBD(%)	SBD(%)	DiC	|DiC|
UNet (Ronneberger et al., 2015)	43.45	14.02	**177.72**	96.80	82.67	-0.1	0.8
ResUNet (Diakogiannis et al., 2020)	23.87	69.31	104.84	96.68	86.51	-0.1	0.6
UNet++ (Zhou et al., 2019)	200.96	47.20	84.75	98.36	88.93	0.05	0.55
DeepLabv3+ (Chen et al., 2018)	**7.79**	**5.86**	117.38	96.63	84.97	0.1	0.7
DSNet (Guo et al., 2024)	33.03	29.33	51.19	93.21	73.90	0.1	0.7
Perspective + UNet (Hu et al., 2024)	90.49	103.85	39.83	97.98	87.86	0.25	0.75
GrotUNet	47.99	104.45	36.82	**98.07**	**89.50**	**0.05**	**0.55**

TABLE 3 Comparison results with state-of-the-art methods on the KOMATSUNA dataset.

Method	FBD(%)	SBD(%)	DiC	|DiC|
UNet (Ronneberger et al., 2015)	96.58	83.71	0.23	0.47
ResUNet (Diakogiannis et al., 2020)	96.39	88.54	**-0.04**	0.28
UNet++ (Zhou et al., 2019)	97.90	92.14	-0.05	0.20
DeepLabv3+ (Chen et al., 2018)	96.85	86.78	-0.1	0.39
DSNet (Guo et al., 2024)	93.88	80.78	0.05	0.38
Perspective + UNet (Hu et al., 2024)	97.58	92.15	**-0.04**	**0.16**
GrotUNet	**97.80**	**92.44**	-0.07	**0.16**

TABLE 4 Comparison results with state-of-the-art methods on MSU-PID dataset.

Method	FBD(%)	SBD(%)	DiC	|DiC|
UNet (Ronneberger et al., 2015)	88.84	80.81	-0.13	0.53
ResUNet (Diakogiannis et al., 2020)	91.11	84.76	-0.12	0.35
UNet++ (Zhou et al., 2019)	91.00	85.21	-0.17	0.35
DeepLabv3+ (Chen et al., 2018)	90.23	82.38	**-0.01**	0.31
DSNet (Guo et al., 2024)	87.16	75.36	-0.29	0.58
Perspective + UNet (Hu et al., 2024)	90.83	84.67	**0.01**	**0.26**
GrotUNet	**91.20**	**85.48**	-0.06	0.28

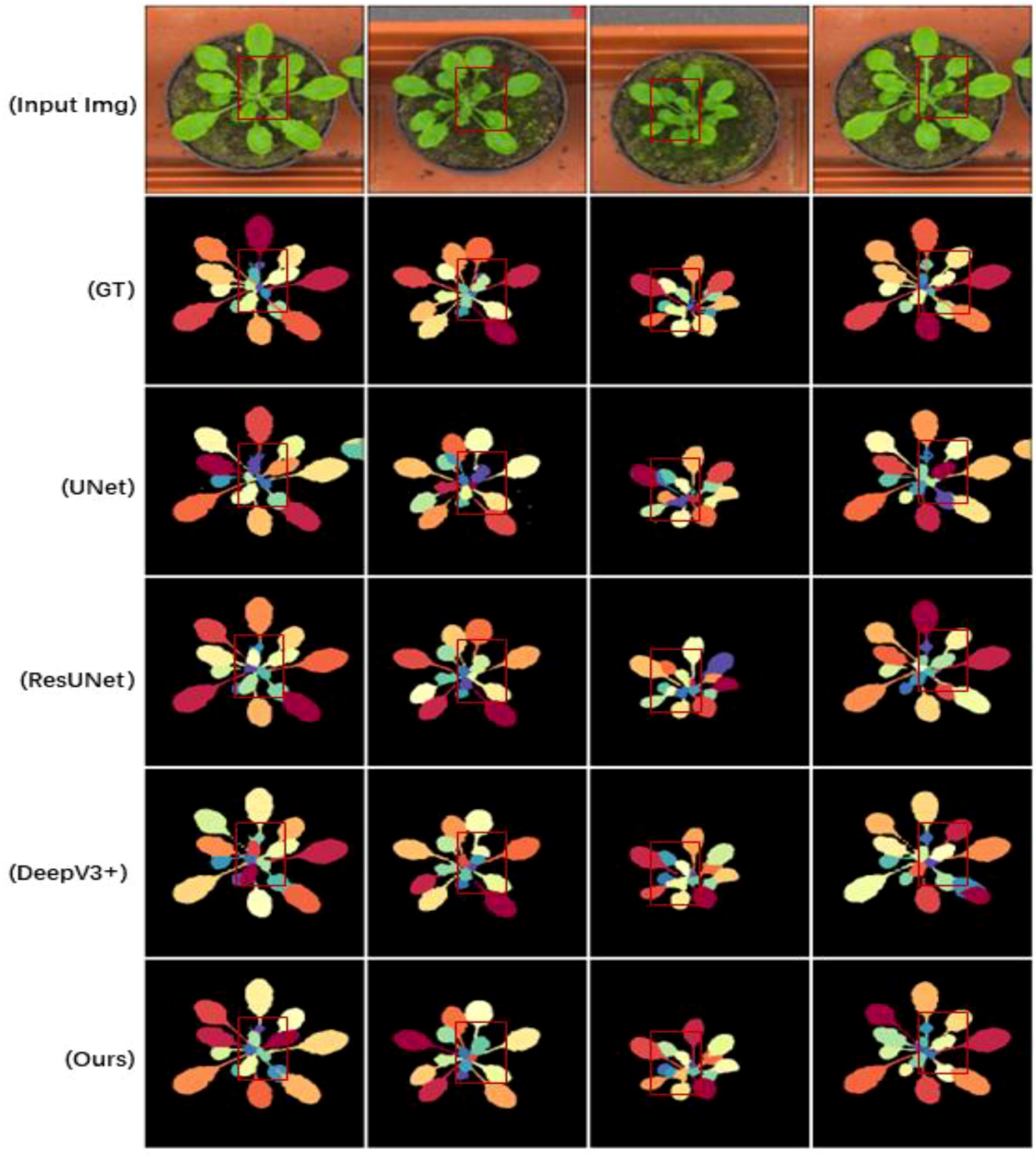

FIGURE 3
Visualizing sample results on the CVPPP dataset.

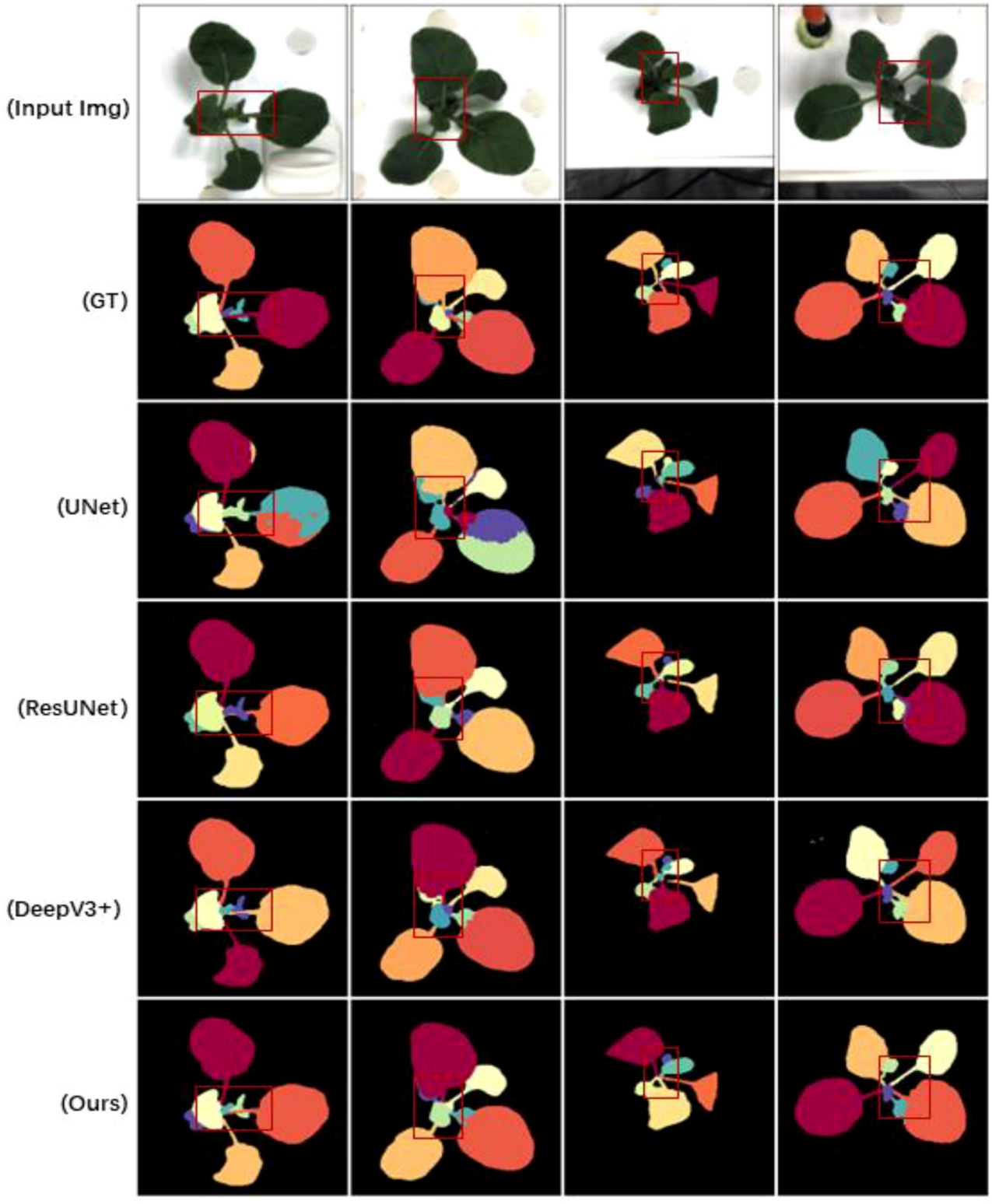

FIGURE 4
Visualizing sample results on the KOMATSUNA dataset.

10.3389/fpls.2025.1378958

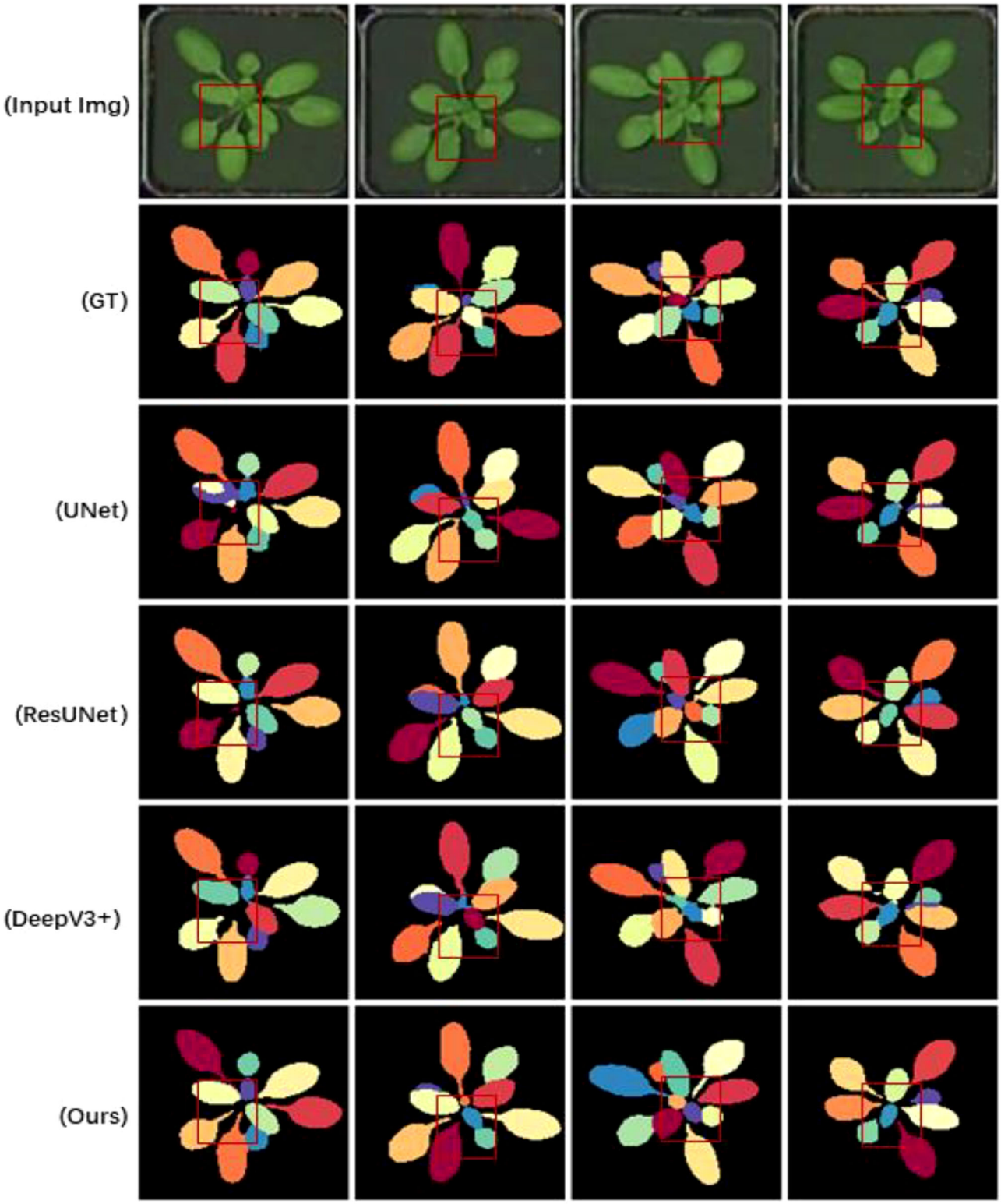

FIGURE 5
Visualizing sample results on the MSU-PID dataset.

TABLE 5 Comparison of SBD and |DiC| results between GrotUNet and state-of-the-art methods on CVPPP. Dataset.

Method	|DiC|	SBD (%)
IPK (Pape and Klukas, 2015)	2.6	74.4
Nottingham (Hu et al., 2024)	3.8	68.3
MSU (Hu et al., 2024)	2.3	66.7
Wageningen (Yin et al., 2014a)	2.2	71.1
Recurrent IS+CRF (Romera-Paredes and Torr, 2016)	1.1	66.6
E2E (Mengye and Richard, 2017)	0.8	84.9
DLoss (Bert De et al., 2017)	1.0	84.2
Deep coloring (Kulikov et al., 2018)	2.0	80.4
ColorRL (Tuan et al., 2021)	1.34	87.3
Eff-UNet++ (Bhagat et al., 2022)	1.15	85.0
GrotUNet	**0.55**	**89.5**

TABLE 6 Comparison of SBD results between GrotUNet and advanced methods on KOMATSUNA and MSU-PID Dataset.

KOMATSUNA		MSU-PID	
Method	SBD (%)	Method	SBD (%)
CVPPP-All (Ward and Moghadam, 2020)	51.34	(Yin et al., 2014a).	63.0
Ward et al (Ward et al., 2018).	62.43	(Yin et al., 2014b).	64.4
UPGen (Ward and Moghadam, 2020)	71.69	(Yin et al., 2017).	65.2
Upen-Incontext (Ward and Moghadam, 2020)	77.76	(Yin et al., 2017).	61.0
Eff-UNet++ (Bhagat et al., 2022)	83.44	Eff-UNet++ (Bhagat et al., 2022)	71.17
GrotUNet	**92.44**	GrotUNet	**85.48**

datasets are given in Tables 2, 3 and 4, respectively. It is observed that GrotUNet achieves FgBgDice: 98.07, 97.80, 91.20; SBD: 89.50, 92.44, 85.48 for the leaf segmentation evaluation metrics on the three datasets, respectively.Meanwhile, the counting evaluation metrics on the three datasets achieves DiC: 0.05, -0.07, -0.06; |DiC|. 0.55, 0.16, 0.28. In terms of the key segmentation evaluation metrics FBD and SBD evaluation, GrotUNet performs the best on all three datasets, exhibiting strong segmentation performance.

Table 2 demonstrates the results of Flops, Parms, and FPS comparisons, where GrotUNet has a large number of parameters and is too slow for inference, but the computational complexity is better than UNet++. Figures 3, 4, and 5 show the visualized qualitative results of GrotUNet compared with several other state-of-the-art methods on CVPPP, KOMATSUN, and MSU-PID datasets, respectively. Observing the three visualizations, it is easy to find that the segmentation of petiole aggregation region by UNet, ResUNet, and DeepV3+ methods on the three plant datasets is unsatisfactory, and some petiole features are not captured. In addition, at the overlap of petiole and leaf blade, and at the boundary between petioles, the ability of GrotUNet to capture fine-grained semantics in the local area is significantly better than that of UNet, ResUNet, and DeepV3+, and the fine features such as petiole are almost completely recognized. Traditional segmentation methods in feature map scale transformation downsampling and upsampling will cause some key semantic information to be lost. Furthermore, the range of sensing field is more limited, and richer features cannot be acquired. The method proposed in this paper prevents the loss of semantic information during the flow of image semantics through the reconstruction of the feature extraction backbone network, sliding connection and multiscale up-sampling fusion mechanism in order to prevent the loss of semantic information during the flow of image semantics in the feature layer.

4.3.2 Comparison with existing studies

To validate the performance of GrotUNet for further verification, this paper compares the evaluation results on CVPPP, KOMATSUNA, and MSU-PID datasets with the extant research methods. Table 5 demonstrates the SBD, |DiC| comparison results on the CVPPP dataset. Table 6 gives the results of comparison between KOMATSUNA and MSU-PID datasets on SBD. Comparing the 2 tables, the performance of GrotUNet leaf segmentation is better than the existing research methods. IPK performs poorly in leaf segmentation and counting due to overlapping leaf blades and crossing leaf margins (Pape and Klukas, 2015). The presence of small leaves and petioles resulted in reduced segmentation and detection ability of Nottingham and Wageningen (Yin et al., 2014a; Scharr et al., 2016). The poor segmentation ability of MSU may be due to dense leaves (Scharr et al., 2016). Deep coloring may have too many post-processing hyper-parameters, which resulted in limited segmentation ability (Kulikov et al., 2018). The method proposed in this paper combines the discriminative loss function to reconstruct the feature extraction backbone network, the sliding connection, and the decoding backbone network, and the performance in leaf segmentation and leaf number calculation reaches the advanced level.

As far as the network architecture is concerned, traditional encoding-decoding architectures lose some of the semantic information in both image downsampling and upsampling. UNet ++ improves segmentation performance by constructing sliding connections in the form of dense connections, which, however, imposes a large amount of computation. Compared with UNet++, GrotUNet's sliding connection design drastically reduces the number of nodes, computational effort, and number of parameters, and retains sufficient semantic information. The multi-scale upsampling fusion design mechanism fuses the higher-order feature maps into the lower-order sub-networks while using 1×1 convolution for feature aggregation. This not only balances the excessive number of parameters well, but also mitigates the semantic loss in the decoding process.

10.3389/fpls.2025.1378958

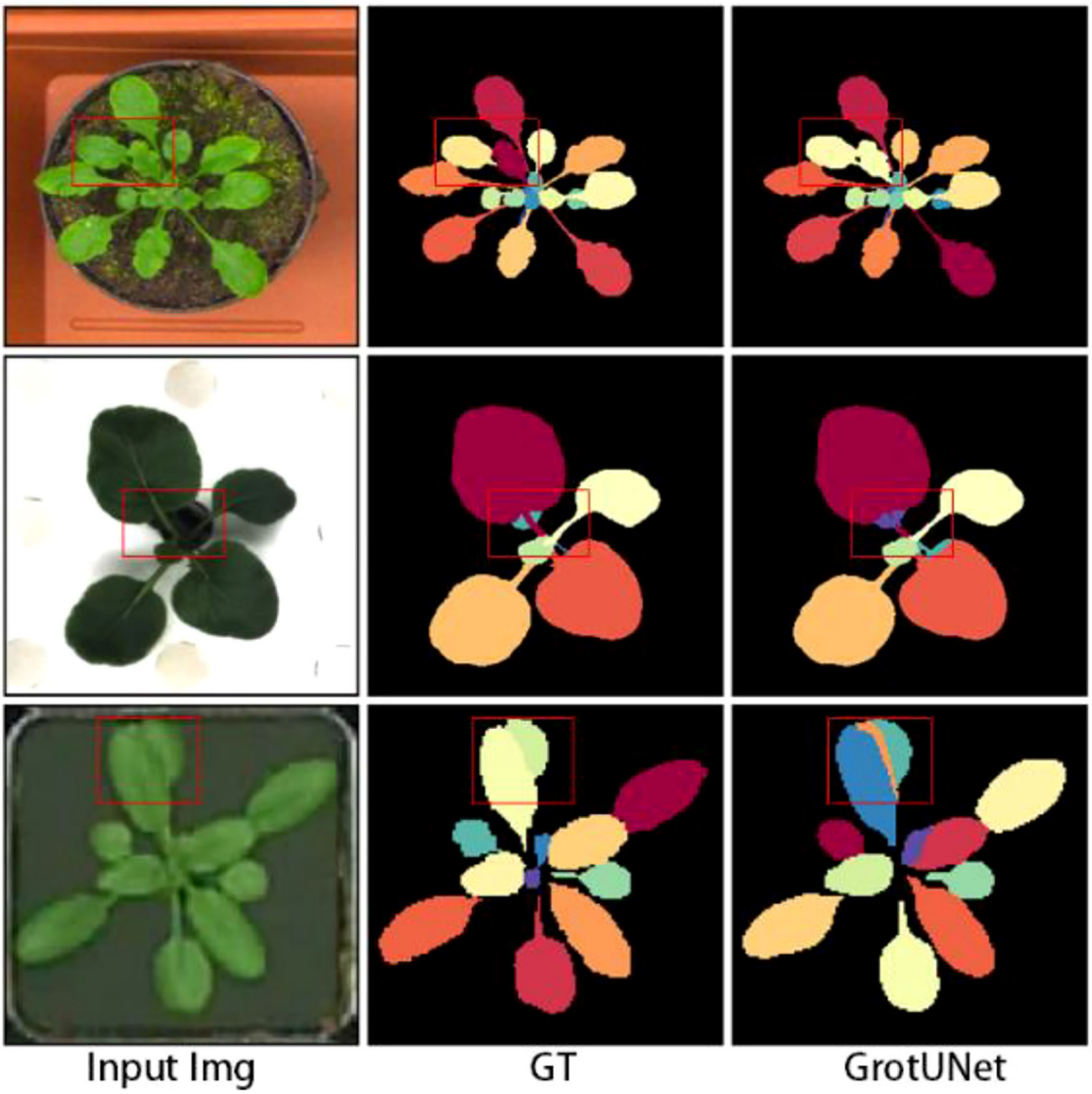

FIGURE 6
Failed cases of GrotUNet segmentation on CVPPP, KOMATSUNA and MSU-PID datasets.

Some cases of segmentation failure of GrotUNet on CVPPP, KOMATSUNA and MSU-PID datasets are shown in Figure 6. It is observed that GrotUNet is unable to accurately detect the number of plant leaves, leaf edge contour, petiole region, etc., and is not sensitive enough to the local area features, which leads to some incorrect segmentation. Meanwhile, this also restricts the further improvement of the model performance, and further research will be done subsequently.

4.4 Ablation study

The existing backbone network for feature extraction in segmentation methods mainly uses the ResNet family of architectures, but in the experiments, it is found that these mainstream architectures perform poorly for detail semantic extraction such as petiole, which restricts the performance of the model. Analyzing the reasons, it may be found that the ResNet architecture itself is deficient in the presence of insufficient feature extraction and serious loss of detail semantics when the feature map scale is reduced. Based on this, this paper redesigns the hybrid feature extraction backbone network, and the ablation study is mainly carried out on this basis. The ablation study is carried out on CVPPP, KOMATSUNA and MSU-PID datasets, and the experimental ablation study mainly verifies the performance of R-Skip, Muti-UP, and OTblock modular design on the key evaluation index SBD, so as to verify whether each module contributes to the segmentation performance.

10.3389/fpls.2025.1378958

TABLE 7 Ablation study in different configurations of GrotUNet.

Configuration	GRblock	WGRblock	OTblock	R-Skip	Mti-UP	Flops(G)	Parms(M)	FPS	SBD(%)		
									CVPPP	KOMATSUNA	MSU-PID
I	✓	✓	×	✓	✓	21.52	19.14	74.25	86.42	91.54	82.42
II	✓	✓	✓	×	×	44.18	103.87	38.11	87.00	88.6	81.96
III	✓	✓	✓	✓	×	46.13	104.12	42.08	86.62	91.51	81.96
IV	✓	✓	✓	×	✓	46.03	104.21	35.97	85.50	91.21	83.14
V (GrotUNet)	✓	✓	✓	✓	✓	47.99	104.45	36.82	**89.50**	**92.44**	**85.48**

Table 7 and Figure 7 give the results and visualizations evaluated on the CVPPP, KOMATSUNA and MSU-PID datasets. Observation of the graphs reveals that Configuration II, which uses the same sliding method and decoding structure as UNet and does not employ the R-Skip and Muti-UP modules, performs poorly in segmenting the petiole detail region. Configurations I, III, and IV have different leaf segmentation performances on the CVPPP, KOMATSUNA, and MSU-PID datasets, and the segmentation effect is unsatisfactory in detail regions such as petiole. However, when R-Skip, Muti-UP, and Otblock modules are all applied, the best leaf segmentation performance is realized and SBD is improved significantly. In addition, Table 7 shows that the large number of parameters in the GrotUNet model is mainly due to the application of the OTblock module, which consists of the Outlooker neighborhood attention layer and the Transformer attention layer. Although the performance of GrotUNet is excellent, it increases the computational complexity and reduces the inference speed, which will be further investigated in the future.

The method in this paper is really experimentally set up for 1 × 1 convolutional feature aggregation operation in sliding joins. In the experiments, it is found that the performance of segmentation using 1×1 convolution is better than 3×3 convolution, which is beneficial for reducing the number of parameters. In addition, in this paper, the higher-order feature maps are up-sampled by multiscale bilinear interpolation and fused into the shallow decoding sub-networks, and feature aggregation is achieved using the 1×1 convolution operation. Through experiments, it is proved that the multi-scale up-sampling fusion mechanism aggregates the higher-order features with the lower-order features, which effectively improves the quality of the feature maps and preserves more semantic information. Overall, GrotUNet achieves 89.50%, 92.44%, and 85.48% SBD for the evaluation metrics on the CVPPP, KOMATSUNA, and MSU-PID datasets, respectively, which are superior to most of the existing research methods.

5 Conclusions

Since plant leaves have overlapping, occluded, and tiny petioles, it is difficult to capture key features using traditional segmentation methods, resulting in inaccurate leaf and petiole detection and poor segmentation performance phenotype. In order to solve the above problems, this paper proposes a novel, end-to-end training leaf segmentation algorithm, GrotUNet. the main contributions of this algorithm are an improved feature extraction encoder, a reconstructed jump connection, and a multiscale upsampling fusion decoder. The encoder consists of three parts: the GRblock, the WGRblock, and the OTblock. The former two utilize the ideas of Resnet and GoogLeNet residual connectivity and parallel branching to fully exploit the semantic features of the image. The latter OTblock, on the other hand, performs one-step mining and encoding of fine-grained image semantic information to obtain finer features. Combining the three effectively extracts the features of local key regions of the instance object. The reconfigured sliding connection module employs a convolutional block at the intermediate node to aggregate semantic information from

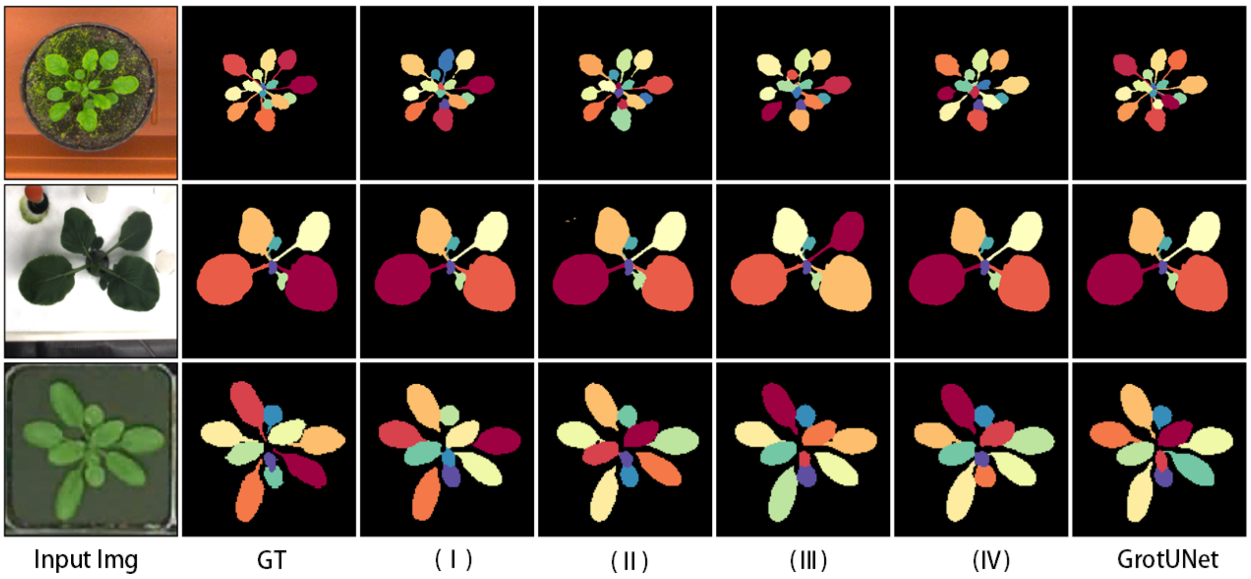

FIGURE 7
Ablation study visualization results on CVPPP, KOMATSUNA and MSU-PID datasets.

different scales, which can make the feature representation of each spatial location richer. The decoding backbone network adopts a multi-scale upsampling fusion design to incorporate the outputs of high-level sub-networks into each low-level sub-network, effectively mitigating the loss of semantic information. Experimental evaluations on CVPPP, KOMATSUNA and MSU-PID datasets show that the proposed method GrotUNet outperforms state-of-the-art methods such as UNet, ResUNet, DeepV3+, UNet++, Perspective + UNet. In the future, GrotUNet will be migrated to the fields of crop disease detection and agricultural product quality inspection to further verify its outstanding performance, aiming to provide a strong contribution to the green development of agriculture (Zhenye et al., 2024).

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: CVPPP: https://www.plant-phenotyping.org/datasets-home, KOMATSUNA: https://limu.ait.kyushu-u.ac.jp/~agri/komatsuna/, MSU-PID: https://cvlab.cse.msu.edu/category/downloads.html.

Author contributions

HD: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Resources, Writing – original draft, Writing – review & editing. BW: Supervision, Writing – original draft, Writing – review & editing. CG: Investigation, Visualization, Writing – original draft. YF: Investigation, Visualization, Writing – original draft.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The research is supported by the Key Laboratory of Education Informalization for Nationalities of Ministry of Education, Yunnan Key Laboratory of Smart Education, Key Laboratory of Digital Learning Supporting Technology, Department of Education of Yunnan Province, and Yunnan International Joint R&D Center of China-Laos-Thailand Educational Digitalization.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Correction note

This article has been corrected with minor changes. These changes do not impact the scientific content of the article.

Publisher's note

References

Bert De, B., Davy, N., and Luc Van, G. (2017). Semantic instance segmentation with a discriminative loss function. *Computing Res. Repository*. abs/1708.02551.

Bhagat, S., Kokare, M., Haswani, V., Hambarde, P., and Kamble, R. (2022). Eff-UNet ++: A novel architecture for plant leaf segmentation and counting. *Ecol. Inf.* 68, 101583. doi: 10.1016/j.ecoinf.2022.101583

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Eur. Conf. Comput. Vision* 11211, 833–851.

Cruz, J. A., Yin, X., Liu, X., Imran, S. M., Morris, D. D., Kramer, D. M., et al. (2016). Multi-modality imagery database for plant phenotyping. *Mach. Vision Appl.* 27, 735–749. doi: 10.1007/s00138-015-0734-6

Diakogiannis, F. I., Waldner, F., Caccetta, P., and Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogrammetry Remote Sens.* 162, 94–114. doi: 10.1016/j.isprsjprs.2020.01.013

Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., et al. (2019). "SSAP: single-shot instance segmentation with affinity pyramid," in *IEEE international conference on computer vision*, vol. 1. , 642–651.

Guo, Z., Bian, L., Wei, H., Li, J., Ni, H, and Huang, X. (2024). "DSNet: a novel way to use atrous convolutions in semantic segmentation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, pp. 3679–3692. doi: 10.1109/TCSVT.2024.3509504

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 770–778.

Hu, J., Chen, S., Pan, Z., Zeng, S., and Yang, W. (2024). "Perspective+Unet:enhancing segmentation with bi-path fusion and efficient non-local attention for superior receptive fields," in *In International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cham: Springer Nature Switzerland) p. 499–509.

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., et al. (2020). "UNet 3 +: A full-scale connected unet for medical image segmentation," in *IEEE international conference on acoustics. Speech, and signal processing*, 1055–1059. abs/2004.08790.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys* 54.10s, 1–41. doi: 10.1145/3505244

Kirillov, A., Wu, Y., He, K., and Girshick, R. (2020). PointRend: image segmentation as rendering. *Comput. Vision Pattern Recognition*, 9796–9805. doi: 10.1109/CVPR42600.2020

Kulikov, V., and Lempitsky, V. (2020). "Instance segmentation of biological images using harmonic embeddings," in *Proceedings - IEEE computer society conference on computer vision and pattern recognition*. p. 3842–3850.

Kulikov, V., Yurchenko, V., and Lempitsky, V. (2018). Instance segmentation by deep coloring. *arXiv preprint arXiv* 1807, 10007.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., and Sun, J. (2018). Detnet: design backbone for object detection. *Eur. Conf. Comput. Vision* 11213, 339–354.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. *Comput. Vision Pattern Recognition*, 8759–8768. abs/1803.01534. doi: 10.1109/CVPR.2018.00913

Mengye, R., and Richard, S. Z. (2017). End-to-End instance segmentation with recurrent attention. *Comput. Vision Pattern Recognition* 1, 293–301.

Minervini, M., Fischbach, A., Scharr, H., and Tsaftaris, S. A. (2016). Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition Lett.* 81, 80–89. doi: 10.1016/j.patrec.2015.10.013

Pape, J. M., and Klukas, C. (2015). 3-D histogram-based segmentation and leaf detection for rosette plants. *Lecture Notes Comput. Sci.* 8928, 61–74.

Romera-Paredes, B., and Torr, P. H. S. (2016). Recurrent instance segmentation. *Lecture Notes Comput. Sci.* 9910, 312–329.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. *Lecture Notes Comput. Sci.* 9351, 234–241.

Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., et al. (2016). Leaf segmentation in plant phenotyping: a collation study. *Mach. Vision Appl.* 27, 585–606. doi: 10.1007/s00138-015-0737-3

Sheng, X., Zhenye, L., Rui, W., Tingting, Z., Yangchun, Y., and Chao, N. (2023). "Few-shot learning for small impurities in tobacco stems with improved YOLOv7,", vol. 11. (IEEE), 48136–48144.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Computer vision and pattern recognition*, 1–9.

Tuan, T. A., Khoa, N. T., Quan, T. M., and Jeong, W. K. (2021). ColorRL: reinforced coloring for end-to-end instance segmentation. *Comput. Vision Pattern Recognition*, 16727–16736. doi: 10.1109/CVPR46437.2021.01645

Uchiyama, H., Sakurai, S., Mishima, M., Arita, D., Okayasu, T., Shimada, A., et al. (2017). "An easy-to-setup 3d phenotyping platform for komatsuna dataset," in *IEEE international conference on computer vision*, vol. 1. 2038–2045.

Van Houdt, G., Mosquera, C., and Nápoles, G. (2020). A review on the long short-term memory model. *Artif. Intell. Rev.* 53, 5929–5955. doi: 10.1007/s10462-020-09838-1

Ward, D., and Moghadam, P. (2020). Scalable learning for bridging the species gap in image-based plant phenotyping. *Comput. Vision Image Understanding* 197, 103009. doi: 10.1016/j.cviu.2020.103009

Ward, D., Moghadam, P., and Hudson, N. (2018). Deep leaf segmentation using synthetic data. *arXiv preprint arXiv* 1807, 10931.

Xingyou, L., Sheng, X., Zhenye, L., Xiaodong, F., Tingting, Z., and Chao, N. (2024). "A candy defect detection method based on styleGAN2 and improved YOLOv7 for imbalanced data," in *FOODS*, vol. 13. .20.

Yin, X., Liu, X., Chen, J., and Kramer, D. M. (2014a). "Multi-leaf alignment from fluorescence plant images," in *Winter conference on applications of computer vision*, 437–444.

Yin, X., Liu, X., Chen, J., and Kramer, D. M. (2014b). "Multi-leaf tracking from fluorescence plant videos," in *IEEE international conference on image processing*, 408–412.

Yin, X., Liu, X., Chen, J., and Kramer, D. M. (2017). Joint multi-leaf segmentation, alignment, and tracking for fluorescence plant videos. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 1411–1423. doi: 10.1109/TPAMI.2017.2728065

Yuan, L., Hou, Q., Jiang, Z., Feng, J., and Yan, S. (2022). Volo: Vision outlooker for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 6575–6586.

Zhang, G., Lu, X., Tan, J., Li, J., Zhang, Z., Li, Q., et al. (2021). "RefineMask: towards high-quality instance segmentation with fine-grained features," in *Proceedings - IEEE computer society conference on computer vision and pattern recognition*, 6861–6869. abs/2104.08569.

Zhenye, L., Dongyi, W., Tingting, Z., Yang, T., and Chao, N. (2024). Review of deep learning-based methods for non-destructive evaluation of agricultural products. *Biosyst. Eng.* 245, 56–83.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2019). UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* 39, 1856–1867. doi: 10.1109/TMI.42