IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 May 2023, accepted 4 June 2023, date of publication 8 June 2023, date of current version 28 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284036

RESEARCH ARTICLE

PD-SegNet: Semantic Segmentation of Small Agricultural Targets in Complex Environments

ZHIJIA ZHU[1,2], MINGKUN JIANG[2], JUN DONG[2,3], SHUANG WU[2], AND FAN MA[2]

[1]Science Island Branch, Graduate School of USTC, Hefei 230026, China
[2]Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[3]Anhui Zhongke Deji Intelligence Technology Company Ltd., Hefei 230045, China

Corresponding author: Jun Dong (dong.jun@iim.ac.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2022YFD2001404.

ABSTRACT The agricultural scene is a typical unstructured scene, which is intricate and heavily affected by sunlight, weather, and other factors. Agricultural segmentation targets are generally small in size and heavily obstructed. At the same time, image segmentation in agricultural scenes has strong application scenarios, such as blooming intensity estimation, which refers to the estimation of the density and intensity of blooms, fruit yield estimation, fruit harvesting positioning, and so on. Currently, CNNs dominate semantic segmentation of agricultural scenes due to the significant computational constraints of using the Transformer module. However, CNNs have several disadvantages, such as limited effective receptive fields and the inability to capture global information, which significantly reduce their segmentation accuracy in complex agricultural scenes. In addition, the simple upsampling process used in CNNs can result in blurred segmentation edges and inferior performance. This paper presents a new semantic segmentation algorithm based on SegFormer: PD-SegNet(Powerful Decoder SegFormer Network), which balances accuracy and computational efficiency and combines dynamic kernel self-renewal with edge-aware optimization. The proposed algorithm demonstrates outstanding performance in two typical agricultural scenarios: apple blossom and apple fruit segmentation detection and sets a new state-of-the-art (SOTA) on the MinneApple Apple segmentation dataset. Experimental results demonstrate that the proposed method outperforms the baseline method in the segmentation of complex small targets. This algorithm can optimize the semantics segmentation of small targets in complex scenes and contributes to the development of smart agriculture.

INDEX TERMS Smart agriculture, deep learning, semantic segmentation, object detection, transformer, dynamic kernel, image edge optimization.

I. INTRODUCTION

The maturation of artificial intelligence technology and its application in various agricultural scenarios, along with the increasingly convenient and fast access to images, has resulted in the development of mature AI applications in agriculture, such as image segmentation. Image segmentation has been applied to various agricultural tasks in farming, orchards, and facility horticulture, replacing time-consuming and repetitive agricultural operations, reducing production costs while improving yields and quality. These tasks include autonomous navigation and obstacle avoidance based on semantic segmentation, maturity detection, crop quality assessment, yield estimation, and flowering intensity estimation. These tasks share a common goal, which is the extraction of agricultural elements (such as fruits, flowers, and canopies) from the rest of the agricultural scene (such as leaves, branches, and sky) using segmentation techniques.

In this type of problem, researchers have attempted to detect different visual cues, including texture, color, and shape, by using various sensors such as spectral cameras, near-infrared (NIR) cameras, thermal cameras, and more. They have employed various methods to achieve these tasks, such as clustering, template matching, adaptive thresholding,

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

and others. This process generally involves complex steps, such as manually selecting features for combination and image preprocessing, and the final results are closely related to the selection of these steps [1], [2]. However, the majority of these methods rely on hand-designed features, which are sensitive to the environment and not easily generalized. Furthermore, the features extracted by these methods are specific to the crop object and highly susceptible to weather conditions, lighting, and occlusion. With the development of deep learning, it is now possible to use ordinary RGB images to perform segmentation using abstract features learned autonomously, with strong generalization performance.

Semantic segmentation is essentially an image-to-image prediction task and has undergone significant development since the introduction of fully convolutional networks (FCNs) [3]. Prior to this, the understanding of semantic segmentation was limited to region-level clustering. The introduction of FCNs revolutionized semantic segmentation by enabling pixel-level classification. Subsequent methods can be seen as improvements and refinements of FCNs. Most modern models are based on an Encoder-Decoder architecture, where the encoder is designed to extract image features, and the decoder maps these features to the final segmentation mask. In the field of agricultural target segmentation, several studies have applied segmentation models to tasks such as plant leaf disease segmentation [4], [5], [6], [7], segmentation of specific whole plants [8], [9], [10], segmentation of plant leaves [11], [12], segmentation of plant flowers [13], [14], [15], and segmentation of common fruits or vegetables [16], [17].

The aforementioned works have a common feature: they use CNNs for feature extraction and obtain feature maps. However, CNNs have the following inherent limitations:

1) During feature extraction, it is necessary to reduce the amount of calculation and conduct down-sampling, which inevitably results in a feature map smaller than the original image. When it is finally mapped to the classification results, the upsampling work will affect the final accuracy.
2) The segmentation task is limited to the convolution operation itself, which only allows for local modeling. Research has confirmed that the actual receptive field of the convolution operation is far smaller than its theoretical receptive field. [18].

Theoretically, we could optimize the above problem by reducing downsampling and increasing the convolution kernel size. However, such a design would inevitably result in a doubling of computation and training time for each unit increase in kernel size or reduction in downsampling rate. Therefore, we require a new model to address this issue.

Since the great success of Transformer [19] in the field of NLP, the superior performance has led to its introduction to computer vision tasks, Dosovitskiy et al. proposed Vision Transformer (ViT) [20] for image classification. Building on the success of ViT, Carion et al. proposed DETR [21] for object detection, and Zheng et al. proposed SETR [22] for semantic segmentation. There are also Pyramid Vision Transformer (PVT) [23], Swin Transformer [24], and other vision decoders improved from ViT. Nevertheless, the traditional Transformer architecture, however, faces the following challenges:

1) ViT outputs single-scale, low-resolution features, not multi-scale features;
2) The cost of the Transformer is too high and very inefficient, making it difficult to deploy in real-time application scenarios;
3) For the semantic segmentation task, the position embeddings mentioned in the traditional Transformer architecture are very inefficient, and it is not necessary for semantic segmentation.

Besides the imperfections of the model itself, the complex agricultural scenario presents numerous challenges. In the case of the outdoor orchard dataset MinneApple [25], for example, there are still distinct challenges when compared to other environments, as shown in Figure 1.

1) The small size of the segmentation targets, each image contains 41.2 apple instances on average, but the average size of each apple instance is only 40*40 pixels, accounting for only 0.17% of the original image size;
2) Varying weather and lighting conditions lead to significant variations in saturation, brightness, and contrast across the images;
3) the same semantic (apple) has different instance features, such as variations in color, such as red and yellow;
4) the same features belong to different semantics, such as the abscission of fruits, which needs to be removed from the segmentation results.

To address the issue of low accuracy in semantic segmentation caused by the aforementioned model and the complex nature of the scene, we propose a new Transformer-based semantic segmentation system. We use an encoder: Mix Vision Transformer(MiT), proposed in SegFormer gives consideration to both efficiency and accuracy. For the lightweight MLP decoder proposed in SegFormer, we optimize the agricultural scene dataset for its small size, complex scene structure, complex and dense small targets, and optimize the segmentation core of semantic segmentation from the static kernel to the dynamic kernel to increase their learning ability.

At the same time, an edge-aware post-processing module is added, which greatly improves the edge information of segmentation results. It has been verified under two mainstream agricultural orchard scene datasets, and the results show that our new algorithm gives consideration to both efficiency and accuracy, providing a new idea approach for subsequent segmentation and detection of similar scenes.

In section I, we provide an overview of our work. In section II, we present a detailed description of our method and the datasets used, with the most important being the use of the Dynamic Kernel Head in section II-C2 to address inter-kernel communication issues in semantic segmentation

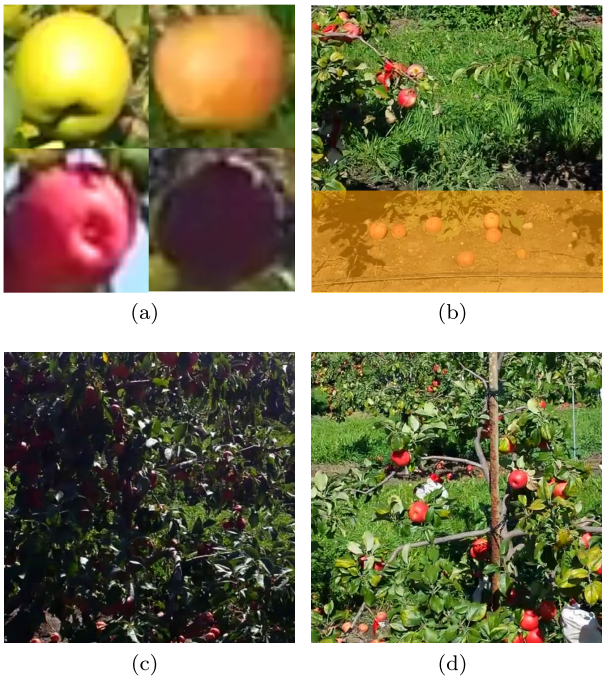

FIGURE 1. Some challenges in MinneApple: (a) Small target apples of different species and characteristics. (b) Abscission of apples with yellow coloration and normal apples in one scene. (c) and (d) Variations in lighting and weather conditions.

for improved accuracy. Additionally, the Complex Points Head is used in section II-C1 to improve segmentation results at object edges. We arrange necessary and rich experiments in section III to demonstrate the effectiveness and real-time performance of our method and summarize our achievements in section IV. The appendix includes some auxiliary experiments and explanations that may be helpful for reading the article.

II. METHODOLOGY

A. OVERALL NETWORK ARCHITECTURE

Nowadays, most of the semantic segmentation models follow the Encoder-Decoder architecture, i.e., the encoder is used to extract the image features, and the decoder is used to decode the above features to complete the image segmentation. One of its distinctive features is that it is an end-to-end learning algorithm. Our method also follows this architecture, so we introduce our method in the order of this architecture. The overall architecture of our network is shown in Figure 2.

B. ENCODER

With the advent of the Transformer era, the most commonly used encoders are the Vision Transformer and the Swin Transformer. As mentioned earlier, they have the drawback of a large number of parameters and computational difficulties.

We use the hierarchical Transformer encoder: Mix Transformer encoder(MiT), proposed in SegFormer, to extract the relevant features from agricultural scenes. It is characterized by its low number of parameters and efficient computation.

The transfer-based encoder usually has the following steps: for an input image $I \in H \times W \times 3$, patch embedding the input image, and convert the original 2D image into a sequence of tokens $I \in 1 \times 1 \times C$, then add the corresponding position embedding to input position information, and after passing the Multi-Head Attention module, the encoder performs Layer Normalization and residual connection.

$$\textbf{Encoder}(I_{out}) = \text{LN}(\text{Position}(\text{E-MSA}(\text{PE})(I_{in}))) + (I_{in}) \tag{1}$$

Among them, the I_{in} refers to the input image, *PE* stands for patch embedding, *E-MSA* refers to an Efficient Multi-Head Attention module, *Position* stands for position information, and *LN* refers to the Layer Normalization, which is commonly used in the Transformer structure.

As mentioned earlier, the positional encoding in the encoder is redundant for semantic segmentation, the Multi-Head Attention layer is the main source of computational effort, MiT mainly makes the following optimizations:

1) Discard positional encoding, only use zero padding to leak location information [26]. The implementation of Mix-FFN used to accomplish this task is as follows:

$$\textbf{Position}(I_{out}) = \text{MLP}\left(\text{GELU}\left(\text{Conv}_{3\times3}\left(\text{MLP}\left(I_{in}\right)\right)\right)\right) + I_{in} \tag{2}$$

Among them, *MLP* is the Multi-Layer Perceptron, $Conv_{3\times3}$ is used to leak location information through zero padding to the main implementation of convolution, *GELU* is the activation function in the process of position coding, and ultimately also through the residuals will be connected to the input and output.

2) In the Multi-Head Attention process, a reduced multiplicity factor R is used to shorten the sequence length of K(Key) and Q(Query), reducing the computational complexity of the entire $attention(Q, K, V)$ by a factor of R.

$$\textbf{Attention}(\frac{Q}{R}, \frac{K}{R}, V) = \text{Softmax}\left(\frac{\frac{Q}{R}\frac{K}{R}^{\top}}{\sqrt{d_{\text{head}}}}\right)V. \tag{3}$$

The final multi-scale feature map contains both shallow, high-resolution basic semantic information and deep, low-resolution abstract semantic information. The rich multi-level information is important and useful for the merits of our next decoding and segmentation work.

C. DECODER

The original decoder in SegFormer is very minimalist, consisting of only MLPs. Since the Transformer encoder has a larger perceptual field than the traditional CNNs, it can be designed without redundant manual components, and the MLP can be used for uniform sizing, upsampling, feature fusion, and final prediction:

$$\textbf{Decoder}(I_{out}) = \text{Linear}(\text{Linear}(\text{Upsample}(\text{Linear})\text{I}_{in})) \tag{4}$$

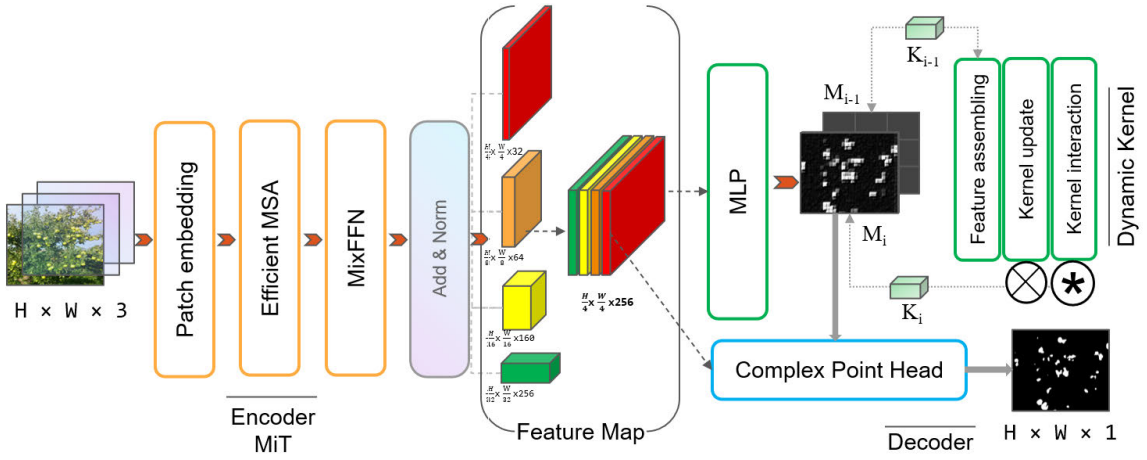

FIGURE 2. The entire network architecture consists of an orange encoder, which generates four different sizes of feature maps. The green decoder includes MLP that produces a coarse segmentation result, and the Dynamic Kernel Head that updates the convolutional kernel parameters and segmentation result in real-time to obtain a fine segmentation result. The blue post-processing module of the decoder uses the Complex Points Head to classify error-prone points in the segmentation boundary, resulting in a single-channel image with a completed classification.

Among them, the underlying implementation of both *Linear* and *Upsample* are *MLPs*, from inside to outside:

1) Linear: Implements channel size unification;
2) Upsample: Upsamples the features to 1/4 of the original size and stitches them together based on channel size;
3) Linear: Fuse the stitched features;
4) Linear: Use the final fused features obtained in the previous step for segmentation prediction.

However, the original decoder performs well for the segmentation of large targets, but it produces unsatisfactory results for small targets. We trained on Cityscapes [27], a dataset with various scale target types. Some training results are presented in Table 9. Therefore, there is room for further optimization of the MLP decoder for our agricultural scenario.

1) DYNAMIC KERNEL HEAD

The reasoning process of semantic segmentation can be summarized as follows: a set of masks is generated by a set of convolutional kernels, with each mask segmenting only one class of objects in the image and different kernels being responsible for generating masks for different objects. With the original decoder in II-C, a set of masks is generated, which is essentially a prediction of the kernel on whether each pixel belongs to its corresponding group. However, since there are differences in appearance and scale among each instance in the corresponding group, our convolutional kernels need to have a stronger discriminative capability.

We were inspired by the dynamic kernel updating mechanism in K-Net [28]. After obtaining rough segmentation results, we dynamically update the kernels based on the semantic information within different segmentation kernels in an attempt to enhance the information exchange between the background segmentation kernel and the foreground

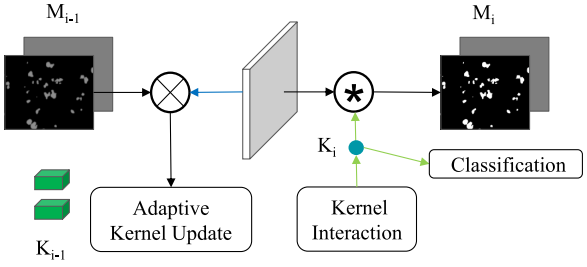

FIGURE 3. Dynamic Kernel Head: Enhanced convolutional kernels for more accurate segmentation results.

segmentation kernel during the segmentation process and obtain more outstanding segmentation results and accuracy. As shown in Figure 3, dynamically augmenting the convolutional kernels with the content in the feature map. The operation of the Dynamic Kernel Head can be outlined in two steps as follows:

1) Kernel Update Head: Kernel dynamicization based on the mask and feature map. The input feature map generated by Encoder:$F \in R^{B\times C\times H\times W}$,mask prediction generated by MLPs:$M \in R^{B\times N\times H\times W}$, create K kernels such that each kernel corresponds to a pixel group: $K \in R^{N\times C}$, Firstly, the assembled feature is obtained by multiplying the input feature map F with the mask prediction M_{i-1} generated by the MLP. Secondly, the kernel is adaptively and dynamically updated through a kernel update method that weights and sums the kernel K_{i-1} and the group features obtained by multiplying the assembled feature with K_{i-1} elements:

$$\mathbf{K_{\tilde{i}}}, \mathbf{M_{\tilde{i}}} = f_i\left(M_{i-1}, K_{i-1}, F\right) \tag{5}$$

Among them, F represents the input feature map, K represents the kernel used for classification, and M

represents the mask obtained after segmentation, F stands for the kernel update method.

2) Kernel Interaction: Globalize the information in the kernel, i.e., communicate information between kernels. This is implemented through Multi-Head Self-Attention & Feed-Forward Neural Network, The final updated segmentation kernel Ki is obtained, and after passing the activation function, normalization layer, and fully connected layer, the new mask prediction is generated through interaction with F:

$$\begin{aligned} \mathbf{K_i} &= \mathrm{MSA}(\mathrm{FFN}(K_{\tilde{i}})) \\ \mathbf{M_i} &= \text{FC-LN-RELU}(K_i) * F \end{aligned} \tag{6}$$

Among them, *MSA* and *FFN* stand for Self-Attention & Feed-Forward Neural Network, *FC* denotes Fully Connected Layer, *LN* represents Layer Normalization, *RELU* is the activation function used during kernel interaction, and F refers to the input feature map. Kernel Interaction enables different kernels to exchange information with each other, i.e., to provide contextual information that allows kernels to implicitly exploit the relationships between groups of images.

This process can be iterated enough times, based on the actual computation volume, to generate kernels with an enhanced ability to differentiate between front and back views. With this component, we finally obtain the updated semantic masks M, $shape(B, C, \frac{H}{4}, \frac{W}{4})$, and the updated classification convolution kernel K.

2) COMPLEX POINTS HEAD

It has been shown that in semantic segmentation, the pixels most likely to be misclassified by the model are typically located at the edges of objects [29]. The main reason for the blurred boundaries is attributed to the upsampling process, which is used to restore the final semantic masks in II-C1) to the size of the original image. The upsampling process leads to poor edge effects.

After passing through the Dynamic Kernel Head, we obtain updated semantic masks that are only one-quarter the size of the input image. Directly upsampling these masks would inevitably result in blurred boundaries and high segmentation errors. Therefore, we abandon the traditional upsampling approach and instead utilize a new upsampling method optimized for accurately segmenting object edges [30], providing better performance on the challenging-to-segment edge regions of objects. The specific component structure is shown in the Figure 4.

The Complex Points Head accepts a feature map with C channels: $F \in R^{B\times C\times H\times W}$. It first picks locations where the values are likely to be significantly different from their neighbors, makes higher resolution predictions from the most uncertain points that are on a small number of possible object boundaries (red points), and performs the normal upsampling method on the other points to finally obtain the labels $M \in R^{B\times N\times H\times W}$.When the feature map is smaller than the original image resolution:

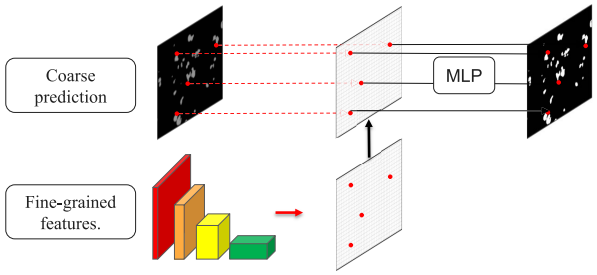

FIGURE 4. The figure shows the network structure of Complex Points Head, which optimizes the segmentation results of difficult-to-separate edge points through this structure.

1) Perform direct 2-fold bilinear interpolation upsampling to obtain coarse prediction;
2) Point Selection: Randomly oversample KN points ($K > 1$) from the uniform distribution, and select the most uncertain βN points ($\beta \in [0, 1]$) from the KN candidate points by interpolating the coarse prediction values of all KN points and calculating the task-specific uncertainty estimate. The remaining $(1-\beta N)$ points are sampled from a uniform distribution. Finally, obtain N "difficult pixels" ;
3) Point-wise feature representation: Obtain the feature representation of N difficult points, which consists of two parts, low-level features: fine-grained features, obtained by bilinear interpolation on the feature map, and high-level features: coarse prediction, obtained by step 1;
4) Use MLP to calculate the representation vector and obtain new predictions.

In the end, we used this up-sampling method, we obtained a single-channel segmentation result that has the same size as the input image. Furthermore, since the additional computation is focused only on the "difficult pixels" we selected instead of being applied globally, the work efficiency of this module is exceptionally high. Subsequent experiments demonstrate that the computational cost of this up-sampling method is extremely low, while significantly improving the edge area, particularly in the AppleA Flower dataset which has complex edges.

D. DATASET

We selected two benchmark datasets as the subjects of our experiments: the MinneApple apple fruit dataset [25] and the AppleA apple flower dataset [13]. These datasets correspond to different growth stages of the same agricultural product, as shown in Figure 5.

The Minneapple dataset contains approximately 1000 images, with over 40,000 accurately segmented individual apples. Each individual has unique features and is influenced by the environment, and the extreme class imbalance between object instances and background pixels is also one of the challenges we face in our work.

FIGURE 5. Two datasets with different characteristics: (1st line)MinneApple: Detection of dense targets and more environmental occlusions. (2nd line)AppleA Flower: Detection of target shape irregularity and edge complexity.

TABLE 1. Dataset details.

Dataset	No.train	No.val	Resolution	Camera model	Camera support	Proportion
MinneApple	670	331	1280×720	Galaxy S4	Hand-held	6.8%[1]
AppleA Flower	100	30	5184×3456	Canon EOS 60D	Hand-held	2.5%

[1] According to MinneApple [25], the average number of strengths per image is 41.2, and the average size of each instance is 40*40 pixels, and the proportion of apples to the whole image is calculated.

TABLE 2. Comparison before and after AppleA Flower changes.

Dataset	No.train	No.val	Resolution
AppleA Flower	100	30	5184×3456
AppleA Flower	2400	720	864×864

In the AppleA Flower dataset with ultra-high resolution, less than 5% of the image area contains the flowers we need to detect. Compared to apple fruits, the boundaries of apple flowers are more complex and variable, making them difficult to segment. This poses a challenging task for feature extraction in our segmentation method.

More detailed information about the dataset is given in the following Table 1.

Since the AppleA dataset has high data accuracy and requires large computational resources, we followed traditional remote sensing image processing methods [31] and split the dataset images into segments of size 864 × 864. To avoid cropping the original image and introducing artificial borders that may impact the final training results, we set the sliding step to be smaller than the split size, which was 432 × 432. The dataset before and after splitting is shown in Table 2.

III. RESULTS

A. TRAINING DETAILS

We implemented our dataset and network structure code on the open-source platform MMSegmentation [34]. By using pre-trained model weights from ImageNet [35], we were able to accelerate our training process. Additionally, we applied a consistent image pre-processing process to all comparison networks, which included:

1) Riseze: Change the image size;
2) RandomCrop: Randomly crop the image size;
3) RandomFlip: Random flip images and their annotations;
4) PhotoMetricDistortion: Optically distort the current image and its annotations using a number of methods;
5) Normalize: Normalize the current image;
6) Padding: Padding the image to the specified size.

We used four NVIDIA GeForce RTX 3090 GPUs for the training computation and trained for 30k iterations per experiment. We used the software versions listed in Table 3.

B. EFFECTIVE RECEPTIVE FIELD

As mentioned earlier, an advantage of the Transformer architecture over traditional CNNs is its larger receptive field,

TABLE 3. Table of software frameworks and their versions.

Ubuntu	Pytorch	CUDA [32]	MMCV-full [33]	MMSegmentation [34]
18.04 LTS	1.10.2	V11.6	V1.6.1	v0.27.0

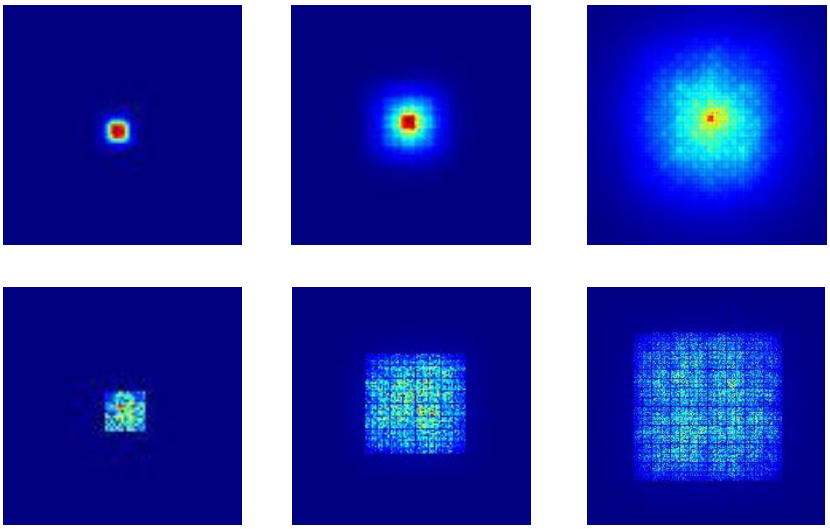

FIGURE 6. Effective receptive field visualization, the first row is the visualization result of DeepLab V3, the second row is the visualization result of our method, and from left to right is the initial layer, middle layer and end layer of the network feature map.

which is beneficial for achieving better segmentation results by incorporating global information during segmentation. In this section, we conduct experiments to verify that the Transformer-based encoder has a larger receptive field than traditional CNNs. We visualize the effective receptive field using a representative DeepLab V3 network with our method, building on the concept first proposed in [18]. The results are shown in Figure 6.

We selected all 331 validation images from the MinneApple dataset as our experimental data. For each image, we selected the center point of the picture as the object of the experiment and calculated the degree to which information from all other points in the image affects it. By obtaining the gradient information for all points with respect to the center point and normalizing the results after accumulating them for all images, we obtained a visually clear final visualization image.

The results showed that conventional CNNs had a Gaussian distribution of influence on the centroids, with the area concentrated near the centroids. In contrast, our method had a wider area of influence with a more uniform degree of influence on the centroids. These results demonstrate that our method has more effective receptive fields.

C. MODEL ACCURACY

To demonstrate the superiority of our method, we conducted a comparative experiment with several excellent classic and advanced algorithms. Among them, FCN and DeepLab V3 are traditional CNN segmentation algorithms, which are still vital, and FCN is the pioneer of traditional

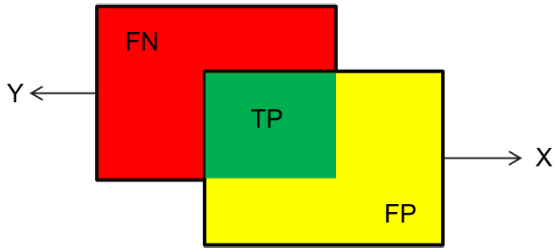

FIGURE 7. TP, FP and FN.

CNN algorithms. NLNet [36] and CCNet [37] are typical representatives of introducing global attention mechanism in the decoder. NLNet is also a pioneer in capturing long-range dependencies. In addition, we also selected UPerNet [38], which uses the powerful Swin Transformer [24] as an encoder, to participate in our comparative experiment.

Due to the limitation of hardware computing power, we followed the principle of similar computational complexity. For the CNN encoder, we used ResNet 50-d8 [39] as its backbone, where, compared with default ResNet [40], ResNet-d replaces the 7×7 conv in the input stem with three 3×3 convs. And for the Swin Transformer encoder, we used Swin-Tiny as its backbone. For MiT, we chose MiT-B2 as its backbone.

We chose Intersection over Union (IoU) and Pixel Accuracy (Acc) as the evaluation criteria for semantic segmentation, as they are commonly used. Since all the agricultural segmentation scenes in this experiment are

TABLE 4. Comparison results on MinneApple.

Method	Backbone	IoU	CIoU[1]	aAcc	CAcc	mDice	CDice
FCN	Res-50-d8	83.10	67.57	98.73	85.24	89.97	80.60
DeepLab V3	Res-50-d8	82.99	67.35	98.67	88.58	89.90	80.49
NLNet	Res-50-d8	83.98	69.15	98.84	84.05	90.68	81.76
CCNet	Res-50-d8	83.90	69.14	98.81	85.92	90.57	81.76
UperNet	Swin-Tiny	84.94	70.97	98.93	84.38	91.24	83.02
Ours	MiT-B2	85.24	71.53	98.98	83.11	91.44	83.40

[1]"C" stands for the category of the segmented object. This notation is used consistently throughout the text.

TABLE 5. Comparison results on AppleA Flower.

Method	Backbone	IoU	CIoU	aAcc	CAcc	mDice	CDice
FCN	Res-50-d8	77.67	56.73	98.64	66.72	85.85	72.39
DeepLab V3	Res-50-d8	78.06	57.46	98.68	66.75	86.15	72.98
NLNet	Res-50-d8	78.48	58.32	98.67	69.73	86.50	73.67
CCNet	Res-50-d8	79.34	58.79	99.16	71.18	87.01	74.05
UperNet	Swin-Tiny	80.11	63.38	98.87	73.28	87.96	77.59
Ours	MiT-B2	81.45	64.03	98.89	76.94	88.75	78.07

TABLE 6. Results of ablation experiments.

Method	Dataset	IoU	CIoU	aAcc	CAcc	mDice	CDice
SegFormer	Apple	84.42	69.99	98.88	81.92	90.89	82.35
SegFormer+KHead	Apple	84.52	70.14	98.94	80.72	90.95	82.45
Ours	Apple	85.24	71.53	98.98	83.11	91.44	83.40
SegFormer	Flower	79.55	60.33	98.61	68.59	87.32	75.26
SegFormer+KHead	Flower	80.39	62.05	98.76	76.13	87.97	76.58
Ours	Flower	81.45	64.03	98.89	76.94	88.75	78.07

binary classification, we also used the common evaluation index in medical image binary segmentation, Dice Similarity Coefficient(Dice), as a supplementary evaluation. Figure 7 illustrates the definitions of TP, FP, and FN, and the formula below explains the difference and connection between IoU and Dice.

$$\begin{aligned} \text{Dice} &= \frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|} \\ \text{IoU} &= \frac{TP}{TP + FP + FN} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \end{aligned} \tag{7}$$

where X is our segmentation result, and Y is the ground truth.

Table 4 presents the results of each method on the MinneApple dataset. Our excellent data pre-processing and network design, combined with optimization of the framework for semantic segmentation, resulted in our metrics achieving state-of-the-art (SOTA) scores compared to the results of the MinneApple Fruit Segmentation Challenge competition [41] released by the University of Minnesota.

Table 5 shows the results of each method on the AppleA Flower dataset.

D. ABLATION EXPERIMENT

Ablation experiments are a common method used to evaluate the importance of different components in a model or method. By systematically removing or modifying each component, we can measure the impact on the overall performance. In the field of computer vision, ablation experiments can help us understand how algorithms work, test hypotheses, and find directions for optimization. Through ablation experiments, we can quantitatively evaluate the performance of each component and provide valuable guidance for further improvement of the algorithm.

To demonstrate the effectiveness of each module in our method, we conducted ablation experiments on two datasets, as shown in Table 6. The experimental results demonstrate that each of our modules is useful and essential for improving the final overall segmentation accuracy.

The Dynamic Kernel Head and Complex Points Head upsampling modules effectively improve the accuracy of the segmentation metrics.

In our task, each semantic kernel corresponds to a unique semantic class, allowing it to learn to segment the same

TABLE 7. Results of computational volume experiments.

Method	Backbone	GFLOPs	Params	FPS
FCN	Res-50-d8	1581.80	49.50M	15.19
DeepLab V3	Res-50-d8	2157.43	68.11M	12.28
NLNet	Res-50-d8	1603.32	50.02M	12.65
CCNet	Res-50-d8	1599.01	49.83M	13.93
UperNet	Swin-Tiny	1879.77	59.94M	11.28
SegFormer	MiT-B2	144.01	24.73M	29.18
SegFormer+KHead	MiT-B2	169.32	31.02M	17.09
Ours	MiT-B2	175.82	31.79M	14.29

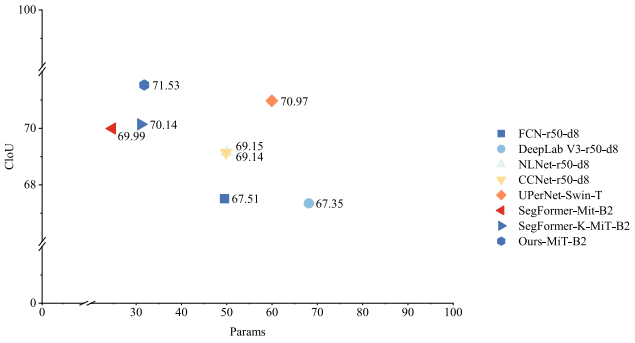

FIGURE 8. Scatter chart for MinneApple: The x-axis represents the parameter quantity in millions, while the y-axis represents CIoU.

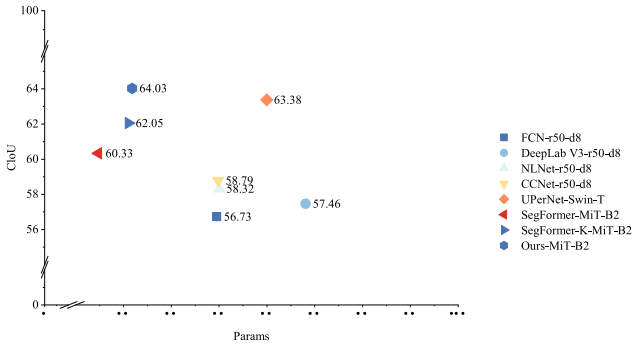

FIGURE 9. Scatter chart for AppleA Flower: The x-axis represents the parameter quantity in millions, while the y-axis represents CIoU.

category in each image. We enhance the kernel by utilizing image information, allowing each segmented kernel to obtain pixel group information corresponding to the kernel through the preliminary mask. This enhances its discriminative ability, and the kernel interaction operation allows the segmented kernel to obtain new global information. Experimental results demonstrate that this module improves the performance of the original segmentation method, resulting in more accurate mask predictions. The Dynamic Kernel Head increased CIoU by 0.15% and 1.72% on two datasets, respectively.

Additionally, to confirm that our method is more accurate at segmenting object edges, we presents some segmentation results in Figure 10 and 11. It can be seen that our upsampling module is very effective for processing difficult segmentation points in the edge region, by detecting and re-segmenting difficult segmentation points, improving both accuracy and the perception of the segmented image edges. These optimizations are not only reflected in the observation of the resulting images but also in the data. The Complex Points Head increased CIoU by 1.39% and 1.98% on both datasets, respectively.

Our two methods did not show a significant improvement in the MinneApple dataset compared to the FlowerA dataset. This is because, compared to flowers, the boundaries of fruits are less complex and easier to be extracted from the surrounding environment. On the other hand, flowers have complex contour shapes and are more difficult to be segmented from the surrounding branches and other complex environmental factors.

E. COMPUTATIONAL VOLUME EXPERIMENTS

For a good semantic segmentation algorithm, high segmentation accuracy alone is not enough. The algorithm should also have a small computation and number of parameters, making it easy to deploy in real-time. We will perform the following experiment: we will use an image of size $I/[3\times2048\times1024]$, run it through the entire network process, and calculate the number of floating-point operations (FLOPs) and parameters (Params) required to process the image.

In practical work, factors other than FLOPs and Params can also affect the training and inference speed, such as memory read and write speed and frequency. Therefore, we took the training data in the MinneApple dataset as an example and calculated the number of images processed per second (FPS) for various networks to more intuitively reflect the computational speed of the model. We used one GPU and explicitly set the number of training samples to 1 and the number of data-loading subprocesses to 2. Since the first several iterations may be very slow, we skipped them and calculated the average value of 200 iterations. The results of the above three experiments are shown in Table 7.

Compared to the baseline, our algorithm has an acceptable decrease in computational cost while yielding excellent performance improvements compared to other algorithms.

TABLE 8. Results of comparison with previous studies.

Method	Dataset	IoU	CIoU	aAcc	CAcc	mDice	CDice
Semi-supervised GMM	Apple	63.50	34.10	96.80	45.50	-	-
User-supervised GMM	Apple	64.90	45.50	95.90	63.40	-	-
U-Net(no pretraining)	Apple	67.80	39.70	96.00	81.80	-	-
U-Net(pretrained)	Apple	68.50	41.00	96.20	84.80	-	-
Mask R-CNN	Apple	76.60	-	98.80	-	-	-
SE-Mask R-CNN	Apple	79.40	-	98.40	-	-	-
Previous best-performing result	Apple	82.40	66.00	98.70	87.40	-	-
Ours	Apple	85.24	71.53	98.98	83.11	91.44	83.40
SPPX+CLARIFAI	Flower	51.30	-	63.10[1]	-	67.80[2]	-
DeepLab+RGR	Flower	71.40	-	79.40	-	83.30	-
DeepLab+SCL	Flower	81.10	-	87.30	-	89.60	-
SSL	Flower	76.20	-	84.80	-	86.10	-
SSL+RGR	Flower	79.60	-	88.10	-	88.60	-
Ours	Flower	81.45	64.03	98.89	76.94	88.75	78.07

[1] Due to the large size of the original image and inconsistent pre-processing methods, the ratio of foreground and background in the final generated image has changed significantly. Furthermore, because the foreground and background have significantly different weights, the aAcc results show a large discrepancy. The same applies to other methods.

[2] The original text uses "F1" to refer to the metric calculated using the Dice coefficient: $\text{dice} = \frac{2\text{TP}}{2\text{TP}+\text{FN}+\text{FP}} = \frac{2*\text{ precision } * \text{ recall}}{\text{precision } + \text{ recall}} = \text{F1}$. The same applies to other methods.

Our method also has an absolute advantage in terms of computational and parametric numbers.

The experimental results indicate that the added computation and parameters are primarily concentrated in the dynamic kernel self-updating module. Moreover, our method has evident advantages in terms of the number of parameters compared to mainstream methods. Our method employs more complex decoder and processing modules, but compared to other state-of-the-art methods, our processing speed still ranks among the top and is even faster than methods with similar accuracy. Consequently, the backbone could be enhanced to MiT-B3 or even MiT-B4, which has a more robust feature extraction capability, in future work.

We have plotted two scatterplots Figure 8 and Figure 9 with the number of parameters for each model as the horizontal coordinate and the CIoU as the vertical coordinate, which shows that our method achieves excellent accuracy compared to the mainstream model while still having a large advantage in the number of parameters. This is highly advantageous for later deployment and real-time inference. We believe that the proposed network will perform well on other tasks with similar complexity.

F. COMPARISON WITH PREVIOUS STUDIES

We have consulted numerous other excellent works and compared our results with them to demonstrate the superior accuracy of our network model. It should be noted that the comparison results below are for reference only, as the pre-processing and hardware conditions of each work vary and cannot rigorously reflect the advantages and disadvantages of various methods.

1) PREVIOUS STUDIES ON MINNEAPPLE DATASET

In paper [25], the authors conducted experiments using four different methods: UNet without preprocessing, UNet with ImageNet preprocessed weights, the semi-supervised method based on Gaussian Mixture Models, and the user-supervised method based on Gaussian Mixture Models. In paper [42], the authors proposed an effective method based on Mask R-CNN for segmenting apples in the MinneApple dataset. For competition [41], we included the previously best-performing result in our comparison.

2) PREVIOUS STUDIES ON APPLEA DATASET

In paper [13], the authors achieved precise flower segmentation by using the Clarifai CNN architecture to classify individual superpixels. We refer to this model as SPPX+CLARIFAI. In paper [14], the authors proposed a novel end-to-end residual convolutional neural network that clusters pixels using Monte Carlo region growing based on seed points provided by the semantic segmentation network. This method further improved the accuracy of flower segmentation and is referred to as DeepLab+RGR. In paper [43], the authors implemented a new post-processing module, where the contours of recognized objects were extracted through energy minimization of the original image and recognition result using the method of level set evolution. This model is referred to as DeepLab+SCL. In paper [44], the authors proposed a self-supervised learning strategy to improve the sensitivity of the segmentation model to different flower species by using automatically generated pseudo-labels. This approach, referred to as SSL (self-supervised

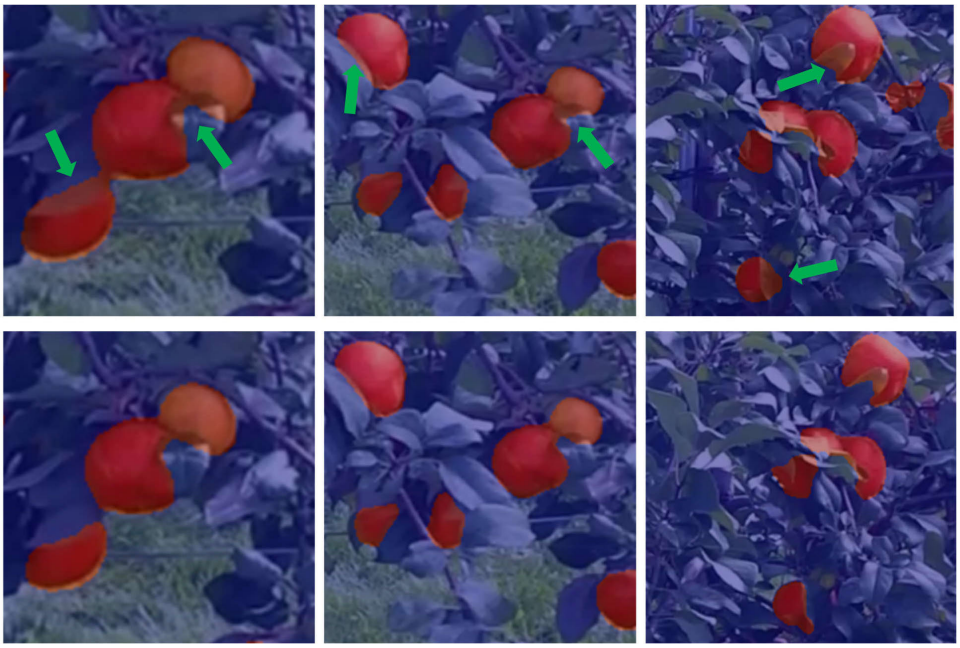

FIGURE 10. The first row shows the segmentation result without boundary optimization, while the second row shows the segmentation result after boundary optimization.

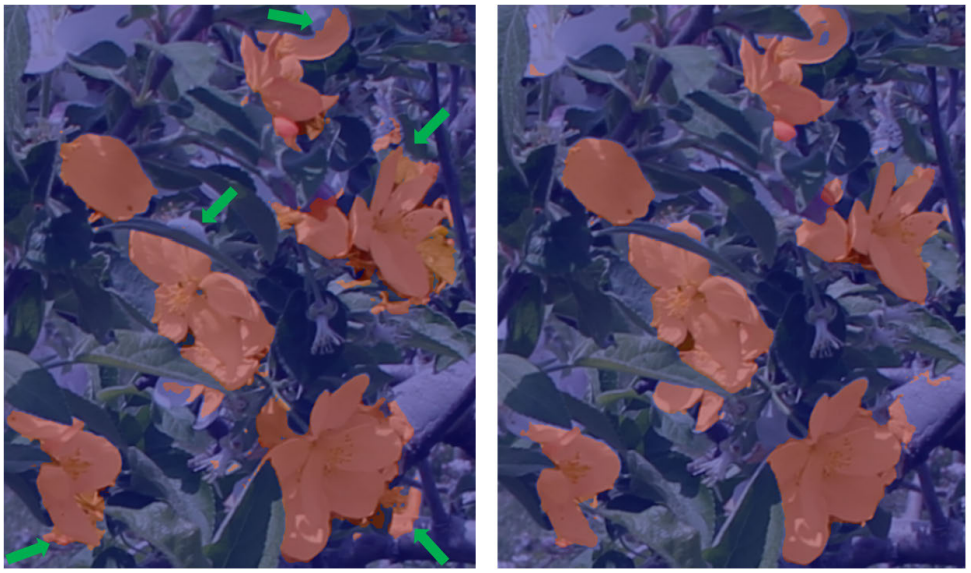

FIGURE 11. The first line shows the segmentation result without boundary optimization, while the second line shows the segmentation result after boundary optimization.

learning), was experimentally combined with the RGR post-processing module. The specific results are shown in Table 8.

IV. CONCLUSION

The paper proposes a Transformer-based semantic segmentation network structure that achieves good results in two segmentation tasks involving unstructured agricultural scenes. These scenes are greatly affected by environmental and lighting conditions. The network achieves these results by accurately extracting global and local contextual information from multi-scale feature maps and by employing a clever decoding and upsampling process. According to the experimental results, we can clearly see that our method takes into account both efficiency and accuracy. Compared with the baseline network SegFormer-B2, we only increased 7.06M parameters but improved 3.7% CIoU on the AppleA Flower dataset, 1.54% CIoU on the MinneApple dataset. Moreover, our method still has a significant advantage over mainstream models in terms of the number of parameters, which is very beneficial for post-deployment and real-time inference. In the future, we plan to conduct experiments using this network for instance segmentation of crops.

TABLE 9. Partial results of SegFormer (MiT-B0) network on Cityscapes dataset.

Class	IoU	Acc
Car	94.51	97.52
Bus	83.99	89.65
Motorcycle	64.02	75.61
Rider	57.19	69.29

APPENDIX A
BOUNDARY OPTIMIZATION

We have compared the results of the two datasets before and after adding the Complex Points Head module and put them in the appendix for reference.See Figure 10, Figure 11 for details.

APPENDIX B
CITYSCAPES RESULTS

We provide a partial score of the SegFormer (MiT-B0) network on the Cityscapes dataset to illustrate its limitations for small target segmentation, see Table 9 for details.

APPENDIX C
ABBREVIATIONS

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
E-MSA	Efficient Multi-Head Self-Attention
FC	Fully Connected Layer
FFN	Feed Forward Networks
FLOPs	floating-point operations
FN	False Negative
FP	False Positive
GELU	Gaussian Error Linear Unit
IOT	Internet of Things
IoU	Intersection over Union
LN	Layer Normalization
MSA	Multi-Head Self-Attention
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
PD-SegNet	Powerful Decoder Segformer Network
PE	Patch Embdding
RELU	Rectified Linear Unit
TP	True Postive

APPENDIX D
CODE

The code is available at https://github.com/plainzzj/PD-SegFormerNetwork.

REFERENCES

[1] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer vision for fruit harvesting robots–state of the art and challenges ahead," *Int. J. Comput. Vis. Robot.*, vol. 3, nos. 1–2, pp. 4–34, 2012.

[2] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Comput. Electron. Agricult.*, vol. 116, pp. 8–19, Aug. 2015.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[4] C. Wang, P. Du, H. Wu, J. Li, C. Zhao, and H. Zhu, "A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net," *Comput. Electron. Agricult.*, vol. 189, Oct. 2021, Art. no. 106373.

[5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018.

[6] U. Afzaal, B. Bhattarai, Y. R. Pandeya, and J. Lee, "An instance segmentation model for strawberry diseases based on mask R-CNN," *Sensors*, vol. 21, no. 19, p. 6565, Sep. 2021.

[7] L. M. Tassis, J. E. Tozzi de Souza, and R. A. Krohling, "A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images," *Comput. Electron. Agricult.*, vol. 186, Jul. 2021, Art. no. 106191.

[8] A. Abdalla, H. Cen, L. Wan, R. Rashid, H. Weng, W. Zhou, and Y. He, "Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure," *Comput. Electron. Agricult.*, vol. 167, Dec. 2019, Art. no. 105091.

[9] S. Kolhar and J. Jagtap, "Convolutional neural network based encoder–decoder architectures for semantic segmentation of plants," *Ecol. Informat.*, vol. 64, Sep. 2021, Art. no. 101373.

[10] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2229–2235.

[11] L. C. Ngugi, M. Abdelwahab, and M. Abo-Zahhad, "Tomato leaf segmentation algorithms for mobile phone applications using deep learning," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105788.

[12] B. R. Hussein, O. A. Malik, W.-H. Ong, and J. W. F. Slik, "Automated extraction of phenotypic leaf traits of individual intact herbarium leaves from herbarium specimen images using deep learning based semantic segmentation," *Sensors*, vol. 21, no. 13, p. 4549, Jul. 2021.

[13] P. A. Dias, A. Tabb, and H. Medeiros, "Apple flower detection using deep convolutional networks," *Comput. Ind.*, vol. 99, pp. 17–28, Aug. 2018.

[14] P. A. Dias, A. Tabb, and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3003–3010, Oct. 2018.

[15] S. Talasila, K. Rawal, and G. Sethi, "PLRSNet: A semantic segmentation network for segmenting plant leaf region under complex background," *Int. J. Intell. Unmanned Syst.*, vol. 11, no. 1, pp. 132–150, Jan. 2023.

[16] Q. Li, W. Jia, M. Sun, S. Hou, and Y. Zheng, "A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment," *Comput. Electron. Agricult.*, vol. 180, Jan. 2021, Art. no. 105900.

[17] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. V. Henten, "Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset," *Comput. Electron. Agricult.*, vol. 144, pp. 284–296, Jan. 2018.

[18] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv: 2010.11929*.

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[22] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6877–6886.

[23] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.
[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
[25] N. Häni, P. Roy, and V. Isler, "MinneApple: A benchmark dataset for apple detection and segmentation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 852–858, Apr. 2020.
[26] M. A. Islam, S. Jia, and N. D. B. Bruce, "How much position information do convolutional neural networks encode?" 2020, *arXiv:2001.08248*.
[27] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[28] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-Net: Towards unified image segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10326–10338.
[29] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6459–6468.
[30] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9796–9805.
[31] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-S. Xia, and X. Bai, "iSAID: A large-scale dataset for instance segmentation in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 28–37.
[32] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?" *Queue*, vol. 6, no. 2, pp. 40–53, 2008.
[33] MMCV Contributors. (2018). *MMCV: OpenMMLab Computer Vision Foundation*. [Online]. Available: https://github.com/open-mmlab/mmcv
[34] MMCV Contributors. (2020). *MMSegmentation: Openmmlab Semantic Segmentation Toolbox and Benchmark*. [Online]. Available: https://github.com/open-mmlab/mmsegmentation
[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[36] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," 2017, *arXiv:1711.07971*.
[37] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "CCNet: Criss-cross attention for semantic segmentation," 2018, *arXiv:1811.11721*.
[38] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 418–434.
[39] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[41] (2019). *Minneapple Fruit Segmentation Challenge*. [Online]. Available: https://competitions.codalab.org/competitions/21694
[42] Y. Liu, G. Yang, Y. Huang, and Y. Yin, "SE-mask R-CNN: An improved mask R-CNN for apple detection and segmentation," *J. Intell. Fuzzy Syst.*, vol. 41, no. 6, pp. 6715–6725, Dec. 2021.
[43] K. Sun, X. Wang, S. Liu, and C. Liu, "Apple, peach, and pear flower detection using semantic segmentation network and shape constraint level set," *Comput. Electron. Agricult.*, vol. 185, Jun. 2021, Art. no. 106150.
[44] A. Siddique, A. Tabb, and H. Medeiros, "Self-supervised learning for panoptic segmentation of multiple fruit flower species," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12387–12394, Oct. 2022.

ZHIJIA ZHU was born in 1994. He received the bachelor's degree in vehicle engineering from the Hefei University of Technology, in 2017. He is currently pursuing the master's degree with the University of Science and Technology of China. His research interests include computer vision and smart agriculture, focusing on how computer vision technology can be applied to improve agricultural production efficiency and quality.

MINGKUN JIANG was born in 1996. He received the master's degree in computer science and technology from Anhui University. He is currently with the Hefei Institutes of Physical Science, Chinese Academy of Sciences. His research interest includes pattern recognition. He is currently engaged in research on 3D object detection and segmentation.

JUN DONG was born in 1973. He received the Ph.D. degree in engineering from the Nanjing University of Posts and Telecommunications, in 2010. He is currently a specially appointed researcher at the Hefei Institutes of Physical Science, Chinese Academy of Sciences. His research interests include computer vision, artificial intelligence and pattern recognition, information networks, and agricultural IoT applications. He has presided over and participated in more than ten national and local natural fund projects, national 863 projects, national science and technology support projects, and provincial and municipal projects.

SHUANG WU was born in 1993. She received the master's degree from Temple University. She is currently with the Institute of Intelligent Machines, Chinese Academy of Sciences. Her research interest includes smart agriculture.

FAN MA was born in 1994. She received the master's degree in agricultural engineering and information technology from Anhui Agricultural University. She is currently with the Institute of Intelligent Machines, Chinese Academy of Sciences. Her research interest includes smart agriculture.