Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049805

ARTICLE

An Improved UNet Lightweight Network for Semantic Segmentation of Weed Images in Corn Fields

Yu Zuo[1] **and Wenwen Li**[2,*]

[1]School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin, 132022, China

[2]School of Mechanical and Control Engineering, Baicheng Normal University, Baicheng, 137000, China

*Corresponding Author: Wenwen Li. Email: liwenwen1017@126.com

Received: 18 January 2024 Accepted: 28 March 2024 Published: 20 June 2024

ABSTRACT

In cornfields, factors such as the similarity between corn seedlings and weeds and the blurring of plant edge details pose challenges to corn and weed segmentation. In addition, remote areas such as farmland are usually constrained by limited computational resources and limited collected data. Therefore, it becomes necessary to lighten the model to better adapt to complex cornfield scene, and make full use of the limited data information. In this paper, we propose an improved image segmentation algorithm based on unet. Firstly, the inverted residual structure is introduced into the contraction path to reduce the number of parameters in the training process and improve the feature extraction ability; secondly, the pyramid pooling module is introduced to enhance the network's ability of acquiring contextual information as well as the ability of dealing with the small target loss problem; and lastly, Finally, to further enhance the segmentation capability of the model, the squeeze and excitation mechanism is introduced in the expansion path. We used images of corn seedlings collected in the field and publicly available corn weed datasets to evaluate the improved model. The improved model has a total parameter of 3.79 M and miou can achieve 87.9%. The fps on a single 3050 ti video card is about 58.9. The experimental results show that the network proposed in this paper can quickly segment corn weeds in a cornfield scenario with good segmentation accuracy.

KEYWORDS

Semantic segmentation; deep learning; UNet; pyramid pooling module

1 Introduction

In recent years, machine vision technology has been developing rapidly, and vision-based technologies are being applied to agricultural scenarios such as disease and pest identification, weed segmentation, and forest species labeling, which have promoted agricultural progress and reduced labor costs and time consumption at the same time. Image segmentation techniques in vision technology are pivotal in enabling farmers to delineate areas afflicted by weed encroachment within their corn fields. This granular insight empowers farmers to deploy targeted interventions, including herbicide application or manual weeding, with heightened precision and efficiency. By strategically directing resources to areas in need, farmers can mitigate input costs while optimizing crop yield. Weeds pose

a significant threat to corn cultivation in cornfields, as they compete vigorously with corn plants for vital resources such as water, nutrients, and sunlight. The precise segmentation of agricultural imagery facilitates the timely implementation of weed control measures, thereby diminishing competition and augmenting corn yield. Traditional weed management practices, such as blanket herbicide application, often yield harmful environmental consequences. Precise segmentation allows farmers to embrace precision agriculture techniques, reducing overall herbicide usage and ameliorating their adverse effects on soil health, water quality, and biodiversity. Also, there are many challenges in the semantic segmentation of cornfield scenes. Weeds often exhibit similar leaf shape, color, and texture as corn plants, making it arduous for segmentation algorithms to distinguish between them accurately. Agricultural fields can harbor diverse weed species, each with distinct appearance and growth patterns. Additionally, soil and other background elements in agricultural imagery can obscure certain parts of the plants, further complicating segmentation tasks. Shadows, reflections, and variations in lighting conditions can also introduce noise and inconsistencies into the imagery, adding another layer of difficulty.

In recent years, the development of deep learning technology has brought about image segmentation algorithms based on deep learning methods, such as UNet [1], Mask Region-based Convolutional Neural Network (Mask R-CNN) [2], and so on. They achieve better results in complex scenes. Unet shows its unique advantage in dealing with the challenge of image segmentation with severe obscuration by fusing low-level and high-level features, which skillfully maintains the edge information while possessing a relatively small computational burden. Also, many variants of Unet have performed well in different areas [3–6]. DeepLabV3+ model significantly improves performance by extending the convolution and increasing the sensory field [7–9]. Segnet is a lightweight segmentation network for real-time image segmentation tasks in environments with limited computational resources, and variants of it are also used in many applications [10–13]. At the same time, an excellent network architecture can improve the model's ability to understand data, generalize, and adapt, thereby achieving better performance in various tasks [14–16].

Recently, researchers have carried out a lot of research work on weed segmentation and have made significant progress in the study of weed segmentation. You et al. [17] introduced Hybrid dilated convolution and drop block in the classification backbone network based on the Deep Neural Network (DNN) segmentation model to expand the sensory field and regularized the weighted values to learn robust features. They also add generic function approximation blocks to the front end of the backbone network and utilize bridge attention blocks to improve classification performance. Islam et al. [18] used images collected from an Australian pepper field to detect weeds and used Random Forests (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) to test the potential of machine learning algorithms for weed and crop classification tasks. Yu et al. [19] used the Deeply Cascaded Semantic Attention Network (DCSAnet) and the Densely Connected Atrous Convolutional Network (DCA module) as the feature extraction network to test image segmentation of soybean and weed images, and they achieved excellent segmentation results. Khan et al. [20] proposed a deep learning framework for encoding and decoding structures. The framework utilizes Dese-inception networks and Atrous spatial pyramid pooling modules to extract multi-scale features and contextual information while using channels and spatial attention units to help recover spatial information. And it achieves 0.81 miou by challenging the rice-weed dataset. Weyler et al. [21] proposed a domain-generalized semantic segmentation approach for robust crop and weed detection by efficiently extending and diversifying the source domain to achieve high performance under different agricultural domain conditions. Picon et al. [22] proposed a semantic segmentation architecture that helps the model converge by extending the Pyramid Scene Parsing Network (PSPNet) architecture and aiding classification loss.

Yun et al. [23] proposed a method for motion blur image restoration based on the Wide Feeling Wild Attention Network (WRA-Net), and based on this, they investigated how to improve the segmentation accuracy of crops and weeds in motion blur images. Hu et al. [24] proposed a method based on unet++ for detecting sugar beet and weeds. They narrowed the semantic gap between the feature mappings of the encoder and decoder sub-networks by redesigning the skip paths. Janneh et al. [25] proposed a lightweight backbone to reduce the number of channels and adjust the information of the feature blocks using residual joins to improve the model prediction capability. They also proposed a multilevel feature weighted fusion (MFRWF) module and a stable convolutional weighted fusion to enhance the contextual information of crops and weeds. Kim et al. [26] conducted research on low-light image segmentation and proposed a segmentation network applicable to low-light environments (LCW-Net). LCW-Net inputs the feature maps generated by the encoder into two decoders to detect the object region and crops and weeds, respectively. The output feature maps of the object decoder are utilized to apply spatial attention to act on the last layer of the crops and weeds decoder to improve the accuracy of segmentation. Meanwhile, they design a loss function to solve the problem of overfitting during model training. Yang et al. [27] found that the existing semantic segmentation models for weeds often neglect the adaptation of channel dimensions. They proposed a multi-scale convolutional attention network (MSFCA-Net) for crop and weed segmentation. MSFCA-Net consists of multi-scale convolutional attention (MCA) with large kernels and a feed-forward network (FFN). The whole network utilized multi-scale attention and skip connections to effectively integrate local and global context information and improve the model's accuracy. Zou et al. [28] proposed a method for image cutting into sub-images, effectively avoiding overfitting while reducing the labeling workload through pre-training and fine-tuning training.

This article proposes a lightweight Unet-based semantic segmentation network model suitable for cornfield scenes. Firstly, we introduce a lightweight backbone as a contracting path to reduce the number of parameters of the network and improve the extraction of feature information for crops and weeds. Secondly, the decoder uses the Squeeze-and-Excitation (SE) mechanism to improve the model segmentation accuracy. Lastly, we use the pyramid pooling module in skip connections to extract multiscale features and contextual associations in networks to enhance the accuracy of image segmentation.

2 Related Work

2.1 U-Net

Data collection for image segmentation of corn fields is often complex because the resources of remote rural corn fields may be limited to collect enough data information, as well as the influence of factors such as geographic conditions, climate change, and soil type, which may vary from one region to another, and the high cost of manual labeling. Unet is a lightweight network structure with a relatively small number of parameters, which makes it more effective for training on small datasets. Compared to some complex network structures, it is easier to learn an effective feature representation with a small number of training samples. The architecture is shown in Fig. 1. It has a U-shaped symmetric structure consisting of a contraction path and an expansion path. The skip connections help to retain and transfer information at different levels. This structure allows the model to learn better and adapt to small datasets. The contraction path is responsible for gradually extracting the image features and reducing the spatial resolution. In contrast, the expansion path gradually recovers the spatial information of the feature map and finally generates the segmentation result with the same size as the input image. Its up-sampling process maps the low-resolution feature maps back to high-resolution.

The structure of gradual upsampling helps to capture features at different scales without requiring a large amount of training data. For small datasets, this structure of Unet can learn the task more efficiently.

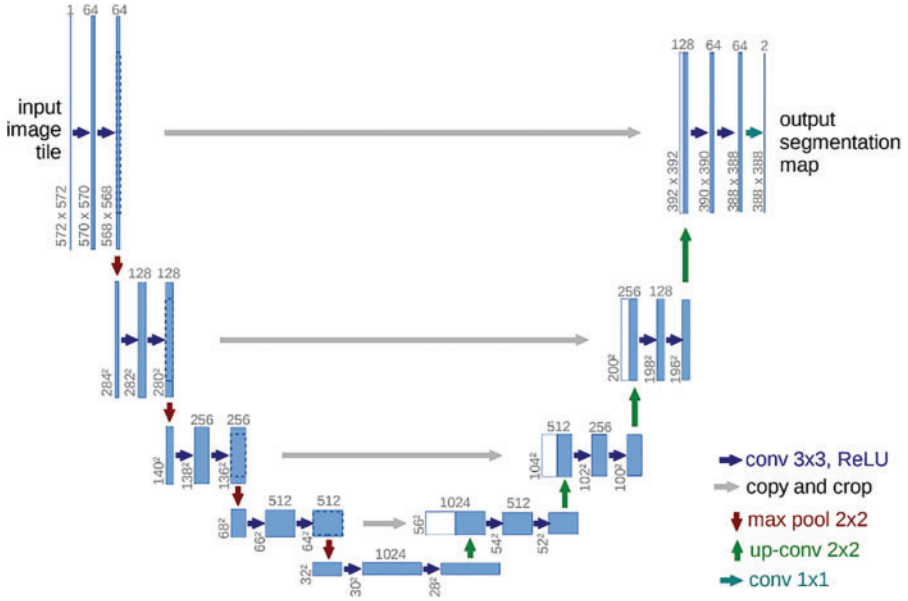

Figure 1: Architecture of U-Net

The contracting path is a classical convolutional neural network consisting of four blocks, each composed of two 3×3 convolutional layers with ReLu activation functions and a max pooling layer. The decoder has the opposite structure to the encoder, using inverse convolutional (or transposed convolutional) layers and jump connections to combine low-level and high-level features. This helps to preserve detailed information and restore spatial resolution. Finally, a 1×1 convolutional layer is used to output the segmented result. Skip connections combine high-resolution features with low-resolution features, which helps to localize the target in the image more accurately. The main operation is first to crop the output of each layer and connect it to the result of the upsampled prediction of the same layer to do the splicing with the features of the deeper layers. Shallow feature maps tend to have more detailed features, whereas deep feature images are greatly condensed due to subsampled prediction; some important feature information is lost and irreversible. The shallow information from the encoder part is fused with the profound information from the decoder to make up for this shortcoming and improve the effect of image segmentation.

2.2 Pyramid Pooling Module

In the early growth stages, corn seedlings and weeds have similar appearances, including color, shape, and texture, making it difficult for algorithms to distinguish between them in complex contexts accurately. Moreover, vegetation and soil in a cornfield may also have similar colors and textures, especially if there is a lack of distinct boundaries between them, making it more difficult for algorithms to distinguish between them accurately.

Most of the traditional scene segmentation networks are designed based on FCNs and null convolutional networks, but such networks often suffer from a lack of ability to collect context; they are also unable to distinguish semantically similar target labels similar to mountains and hills,

buildings, and edifices; and it is similarly a significant challenge for small target object prediction [29]. The primary motivation of the Pyramid Pooling Module (PPM) is to address the problem of capturing contextual information at different scales. In semantic segmentation tasks, understanding the context at multiple scales is essential for accurately identifying object boundaries and capturing the scene's global context. By performing pooling operations on the feature maps at different scales, PPM can catch both local and global contextual information, which enables the model to understand the semantic structure in the image better. The structure is shown in Fig. 2.

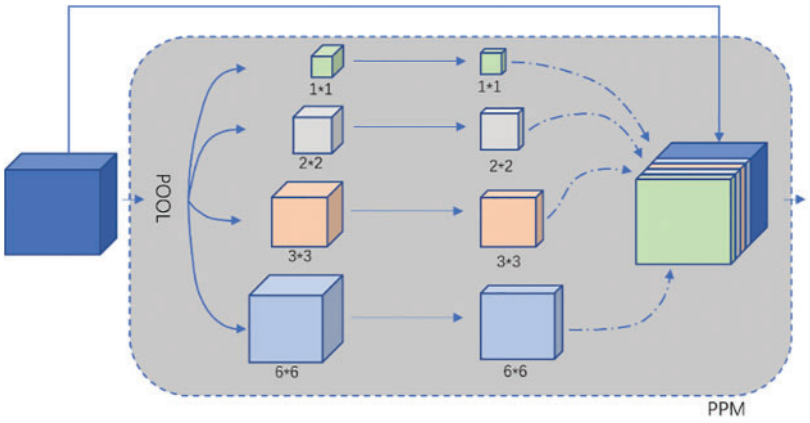

Figure 2: Pyramid pooling structure (PPM)

Skip connection is a structure used in deep learning to build neural networks, also known as residual connections. This type of connection passes input signals directly to the output layer by spanning several layers in the network. Establishing direct connections between different neural network layers allows information to be passed directly from the input layers to the output layers. This approach retains more high-resolution details in the advanced feature maps, providing multi-scale and multi-level information for later image segmentation. Thus, the models can learn complex features better.

2.3 Inverted Residual Structures and the Last Stage in MobileNetV3

The bottleneck structure is a common architectural component in deep neural networks and is especially prevalent in residual networks. This structure reduces the computational complexity and the model's parameters. At the same time, the model's ability to acquire information is maintained. The inverted bottleneck structure is an enhanced structure introduced in MobileNetV2. Compared to the standard bottleneck structure, the inverted bottleneck structure has the reverse order. The inverted residual structure extends the feature map before compressing it. Depthwise separated convolution is also referenced in the inverse residual structure. It can extract spatial features independently for each channel during the convolution process so that the model can capture various local features and patterns in the input image. This structure, with reverse order, helps the model better capture low-level features and is often used for lightweight models to meet the requirements of mobile devices.

MobileNetV3 [30] inherits the inverted residual structure of MobileNetV2 [31] and improves the Bottleneck structure. The MobileNetV3-Large architecture reduces parameter usage by utilizing 1×1 convolution from the bottleneck residual block while maximizing the performance of depth and point-by-point convolution for better classification accuracy [32]. As shown in Table 1. It introduces the hardswish (unify to H-swish) activation function and the Squeeze-and-Excitation (SE) module

[33]. The SE module is placed after the DW convolution. It performs global average pooling for each channel and outputs it through two fully connected layers (FC) with different activation functions. Lastly, it is multiplied by the input feature map to get the final feature map. The SE module improves feature acquisition by adjusting the channel weights so that the network can focus more on essential features. This operation reduces the training time and improves the model's accuracy. The H-swish function originates from the swish function. It retains the nonlinear nature of Swish, allowing networks to learn more complex patterns and features. The swish and H-swish are defined as follows:

$$\text{swish} = x \cdot \text{sigmod}\,(\beta x) \tag{1}$$

$$\text{hardswish} = \begin{cases} 0 & \text{if} x \leq -3 \\ x & \text{if} x \geq +3 \\ x \cdot \text{ReLu6}(x+3)/6 & otherwise \end{cases} \tag{2}$$

where β represents the learnable parameter, it can be seen from Eq. (1), which is optimized by backpropagation during training. As shown in Eq. (2), H-swish uses a piecewise function, making the computation lighter. Compared to Swish, H-swish has a more straightforward form of computation. Using H-swish can reduce the computational burden on the model, especially in resource-constrained environments such as mobile devices and embedded systems.

Table 1: MobileNetV3_Large network structure

Input	Operator	Exp size	Output	SE	NL	Stride
$224 * 224 * 3$	conv2d	–	16	–	HS	2
$112 * 112 * 16$	bneck, $3 * 3$	16	16	–	RE	1
$112 * 112 * 16$	bneck, $3 * 3$	64	24	–	RE	2
$56 * 56 * 24$	bneck, $3 * 3$	72	24	–	RE	1
$56 * 56 * 24$	bneck, $5 * 5$	72	40	√	RE	2
$28 * 28 * 40$	bneck, $5 * 5$	120	40	√	RE	1
$28 * 28 * 40$	bneck, $5 * 5$	120	40	√	RE	1
$28 * 28 * 40$	bneck, $3 * 3$	240	80	–	HS	2
$14 * 14 * 80$	bneck, $3 * 3$	200	80	–	HS	1
$14 * 14 * 80$	bneck, $3 * 3$	184	80	–	HS	1
$14 * 14 * 80$	bneck, $3 * 3$	184	80	–	HS	1
$14 * 14 * 80$	bneck, $3 * 3$	480	112	√	HS	1
$14 * 14 * 112$	bneck, $3 * 3$	672	112	√	HS	1
$14 * 14 * 112$	bneck, $5 * 5$	672	160	√	HS	2
$7 * 7 * 160$	bneck, $5 * 5$	960	160	√	HS	1
$7 * 7 * 160$	bneck, $5 * 5$	960	160	√	HS	1
$7 * 7 * 160$	conv2d, $1 * 1$	–	960	–	HS	1
$7 * 7 * 960$	pool, $7 * 7$	–	–	–	–	1
$1 * 1 * 960$	conv2d, $1 * 1$, NBN	–	1280	–	HS	1
$1 * 1 * 1280$	conv2d, $1 * 1$, NBN	–	k	–	–	1

MobileNetV3 also redesigns the Original Last Stage, as shown in Fig. 3. The Efficient Last Stage module allows the model to reduce three computationally complex layers at the end of the network without losing accuracy, significantly reducing the number of calculations. The Efficient Last Stage is shown in Fig. 4.

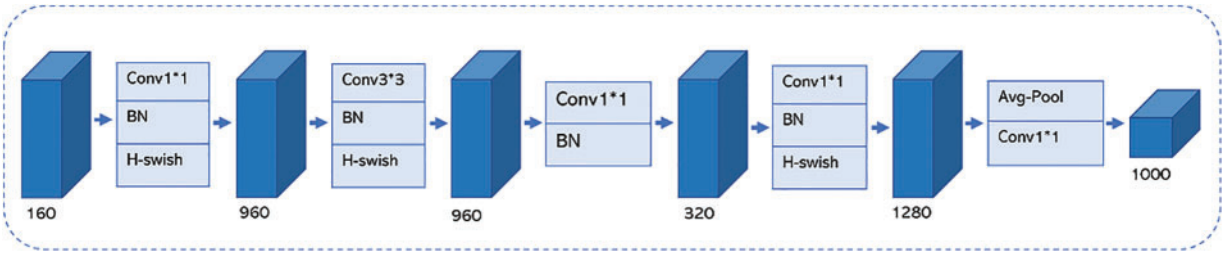

Figure 3: Original last stage

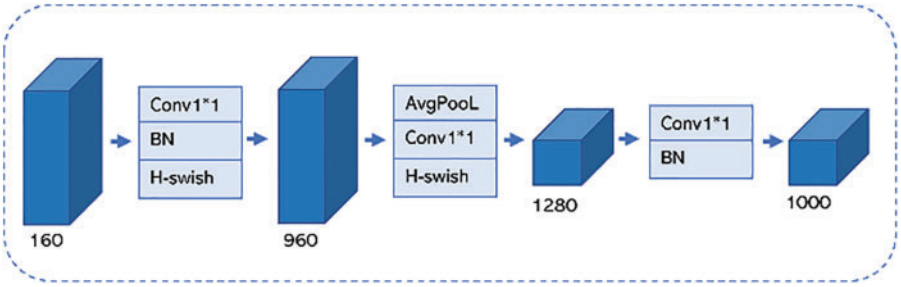

Figure 4: Efficient Last Stage

3 Approach

The architecture of Unet helps learn information from different levels of feature maps and is helpful in better capturing and utilizing limited information of samples on small datasets. Although Unet has a relatively minor number of parameters, it still needs a relatively high amount of storage resources when processing high-resolution images. This is not conducive to the model working on mobile devices or embedded systems. Meanwhile, the original version of UNet needs an explicit attention mechanism, which may lead to a lack of flexibility regarding the importance of different regions when processing cornfield scenes. There may be similar texture, color, or shape features between corn and weeds, making it difficult for Unet to distinguish them in the image. For some tasks, the training time of U-Net is also relatively longer. This is a disadvantage for applications that require fast training and iteration.

We construct an improved Unet model for corns and weeds image segmentation. The improved model is shown in Fig. 5. The overall architecture consists of the compression network, the PPM module, and the extension network. The input image is passed through a compression network to reduce the size of the feature map. The outputs of the different encoding layers are then processed by a pyramid pooling module, thereby capturing the contextual information at various scales. The outputs are then concatenated with the corresponding decoding layers. Finally, the feature map is extended to achieve prediction.

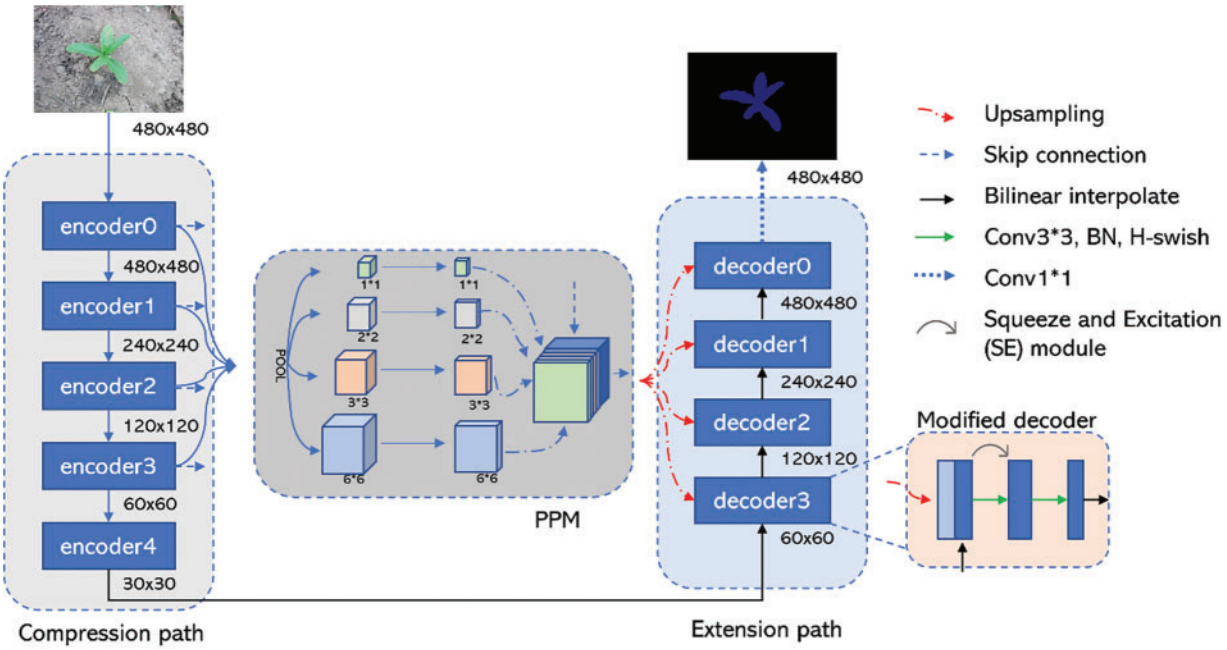

Figure 5: Network structural diagram

3.1 Compression Network Based on MobileNetV3_Large

Under the premise of guaranteeing the model's performance, a well-performing backbone network can effectively extract representative features from the original input data while reducing the model's parameters and computational complexity. In the compression path of the original Unet, the size of the feature map is reduced by applying the max pooling operation, which reduces the spatial resolution. In some cases, the compression paths may be more computationally intensive due to the use of large sensory fields. The above issues may lead to slower or unsuitable operation of the model in resource-constrained environments. MobileNetV3 better captures image semantics by dividing the network into multiple stages and repeating multiple basic blocks within each stage, thus gradually extracting and combining feature information at different levels. By increasing the number of repetitions and the number of basic blocks, the depth and expressiveness of the network are improved, making it more capable of learning complex features from the input data. This multi-layer design allows the network to represent the input data more comprehensively and accurately at different levels of abstraction, improving the performance and accuracy of the model. Given the above issues, this paper introduces MobileNetV3_Large as a compression network, the structure of which is shown in Table 1. The lightweight block design, attention mechanism, and maximized message passing in MobileNetV3 can help improve the learning ability of the model and help it perform better on some complex tasks. Here, layers 1st, 3rd, 6th, and 12th of MobileNetV3_Large are chosen as the outputs of encoder 0∼3, and the production of layer 15th is connected to the expansion path after up-sampling. By retaining output data at different levels of abstraction, a more comprehensive and accurate representation is provided for deeper levels, thus improving the performance and accuracy of the model. Migration learning can utilize features and weights learned on large datasets to accelerate model convergence on small datasets and improve performance. The weight parameters obtained from pretraining of MobileNetV3_Large

on the ImageNet-1K dataset are introduced and migrated to this task to improve the speed of model training and the accuracy of model segmentation.

3.2 Modified Decoder

The advantages of H-swish in terms of nonlinear characteristics, reducing gradient vanishing, high computational efficiency, and compatibility with hardware make it an activation function that achieves good performance in lightweight models. In this paper, we use H-swish as the decoder's activation function. The SE attention mechanism has fewer parameters and does not significantly increase the model's size. The SE mechanism can improve the feature representation by learning the weights of each channel, which makes the network pay more attention to the essential features. This is very helpful in improving the performance of lightweight models. This paper introduces the SE attention mechanism in the decoder after the first convolution. The SE attention mechanism performs global average pooling on the input feature map and compresses the feature values of each channel. Next, through two fully connected layers, the weights of each channel are learned. In the first fully connected layer, the number of nodes in its fully connected layer equals 1/4 of the input feature matrix channel, and the channel of the second fully connected layer is consistent with the channel of our feature matrix. After average pooling and two fully connected layers, the output feature vector can be interpreted as a weight relationship for each channel of the feature matrix before SE, and the more critical channels will be given more significant weight. In contrast, the less essential channel dimensions correspond to a smaller weight. Lastly, the weighted feature maps are generated by multiplying the original input feature maps with the learned excitation weights.

The overall operation is as follows: First, the upsampled output is merged with the output of the PPM module. The merged results are processed by a 3×3 convolution with BN and H-swish to reduce the number of channels. The number of channels will be reduced to half of the original. Then, the channel importance of the feature map is dynamically adjusted by processing through the SE mechanism to enhance the characterization of the network. The number of channels is reshaped by performing the same 3×3 convolution as BN and H-swish; the number of channels here will be the same as the number of channels of the feature map entered into the next decoder. Finally, the output is upsampled by bilinear interpolation to further reduce the size of the feature map in preparation for the subsequent stitching. The modified decoder is shown in Fig. 6.

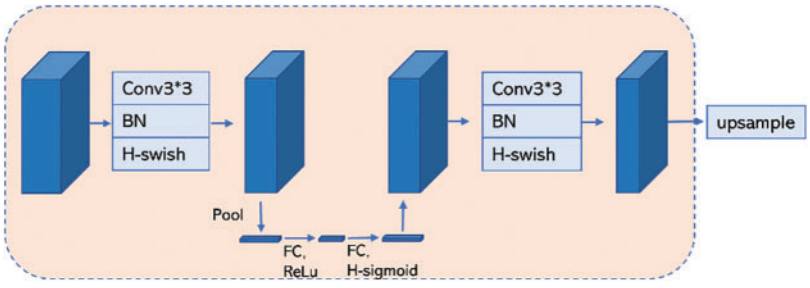

Figure 6: Modified decoder

3.3 Skip Connection and Pyramid Pooling Module

Skip connection allows direct information transfer between shallow and deep layers, which helps maintain the detail of the original image. The direct, cross-layer transfer of information not only helps alleviate the situation where the model becomes difficult to train as depth increases but also makes

it easier for the model to learn identity mapping and thus converge more easily. However, incorrect selection of the number of layers to connect and how to connect them can lead to performance degradation. Researchers must carefully tune and validate this to choose the right design. The original skip connection that directly connects the shallow and deep layers may transfer noise or irrelevant information from the shallow layer to the deeper layer, and this case may lead to a possible semantic gap between the two combined feature sets, which in turn negatively affects the performance of the model [34]. As shown in Fig. 5, the PPM module is applied to the original skip connections to enable the network to acquire the semantic information of the context more comprehensively, thus reducing the roughness of image segmentation and improving the accuracy of image segmentation.

Specifically, pooling kernels or windows of different sizes are applied to the input feature map to capture features at different scales. Typically, this involves multiple pooling operations, such as average pooling or max pooling, each corresponding to a different scale. In this step, we use global average pooling kernels of 1×1, 2×2, 3×3, and 6×6 sizes. Since the pooling operation at each scale generates feature maps with potentially different spatial dimensions, subsequent scaling is required. To achieve feature fusion, we employ a 1×1 convolution kernel and bilinear interpolation to adjust the feature maps at each scale to have the same height and width. This way, information from different scales can work more synergistically and provide consistent inputs for subsequent processing. The adjusted feature maps are formed into a pyramid feature map by stitching them by channel dimension. This fusion strategy ensures that features at each scale are preserved, providing more comprehensive and richer information for the model. The final pyramid feature map can be fed into downstream neural networks for semantic segmentation tasks. Due to the pyramid pooling module, the model becomes more robust and can better adapt to targets of different sizes and complex scenes, improving the overall performance.

4 Experiments and Analysis

4.1 Experiments Settings

The dataset is partly images of corn at the seedling stage collected from a corn field in Tai'an, Shandong Province, China. The other part is from a publicly available dataset (corn weed datasets) containing corn seedlings, bluegrasses, cirsium setosums, sedges, and chenopodium albums. The images are taken from fields with complex backgrounds and light intensities. The images in the dataset were taken from natural environments using the digital camera and contain 1024 images. The dataset contains about 209 images of common corn, about 209 images of bluegrasses, about 205 images of cirsium setosums, about 184 images of sedges, and about 207 images of five species of chenopodium albums. Also, images in the dataset were photographed at multiple angles, as shown in Fig. 7, and under different light conditions, as shown in Fig. 8. To increase the diversity of the dataset and enhance the segmentation ability of the model for corn and weeds, a preprocessing operation of one-half probability of horizontal and vertical flipping was performed on the images during model training. Each image was normalized and cropped to 480×480 pixels to reduce computation and training time. The dataset is divided into training and validation sets in the ratio of 9:1. Some images in the dataset are shown in Fig. 9. LabelMe is a widely used image annotation tool that provides a user-friendly interface and allows users to label objects in an image manually. This tool is commonly used for dataset annotation in computer vision. In this experiment, the labelMe annotation tool manually annotates plant images with a polygonal labeling pattern to support model training and evaluation. Some of the images are on display in Fig. 10.

Figure 7: Corn seedlings photographed at different angles

Figure 8: Corn seedlings photographed under different light conditions

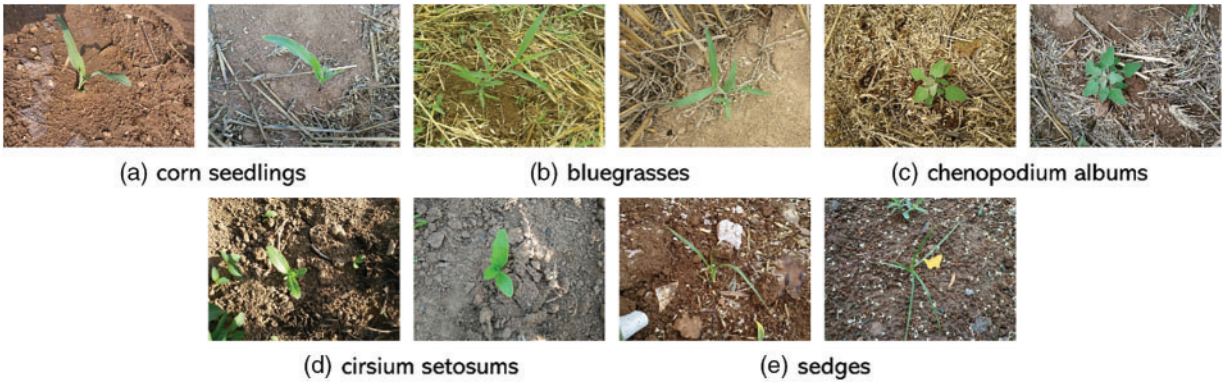

Figure 9: Part of the dataset

Figure 10: Part of the manually labeled segmented images

The experimental setup was as follows: Intel (R) × (R) Platinum 8255C @ 2.50 GHz CPU, an RTX 3080, 10 G of video memory, Cuda 11.8, Python 3.8, PyTorch 2.0.0 deep learning framework, and ubuntu 20.04 operating system. The model was trained for 200 epochs at a learning rate of 0.01, and SGD was chosen as the optimization algorithm. The weight decay coefficient was set to 1×10^{-4}, the batch size was set to 4, and the momentum factor was set to 0.9. The weights were pre-trained using MobileNetV3_Large provided by PyTorch. Finally, the trained model was evaluated using validation images. The experiment was implemented on pycharm software.

4.2 Evaluation Metrics

To validate whether the model is used with corn and weed image segmentation, this paper uses dice coefficient, mean intersection-over-union (miou), global accuracy, and the number of parameters as evaluation metrics. The dice coefficient is a statistical metric used to measure the similarity of two samples and is commonly used in image segmentation tasks. The higher degree of overlap between predicted and accurate labels, the dice coefficient is closer to 1, indicating the more accurate segmentation results. On the contrary, if the degree of overlap is low, the dice coefficient will be close to 0. The formula is as follows:

$$\text{Dice\_}Cofficient = \frac{2\,|X \cap \text{Y}|}{|X| + |Y|} \tag{3}$$

where X and Y denote the actual foreground and predicted foreground, respectively, and |X∩Y| denotes the portion of X that intersects Y. Global accuracy represents the ratio of correctly predicted pixels to the total number of pixels in the image. The formula is as follows:

$$Global\_accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where TP represents the actual class of the sample as positive and the prediction as positive. TN means that the proper category of the sample is negative and that the model predicts a negative category. FP represents a sample whose accurate category is negative, but which the model recognizes as positive. And FN represents a sample whose accurate category is positive, but which the model identifies as negative. Iou is a measure of segmentation accuracy that calculates the ratio of intersection and concatenation between the predicted segmentation results and the actual labels. Miou is the value obtained by averaging the ious of multiple samples or categories. Higher values of iou and miou indicate a better match between the segmentation result and the actual label. The formulas are as follows:

$$IOU = \frac{TP}{TP + FN + FP} \tag{5}$$

$$MeanIOU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{TP}{TP + FN + FP} \tag{6}$$

The number of floating point operations (FLOPs) depends on the model and can be used to evaluate model complexity. It was used as a criterion to assess the complexity of the model. Frames Per Second (FPS) refers to how many frames per second can be processed by the model, you can use the number of images that can be processed in a second or the time needed to process an image to evaluate the speed of the model to segment images.

4.3 Experimental Results and Analysis

First, we test the segmentation performance of the original Unet on the created dataset. Then, we modify the corresponding improved modules in turn and test their effectiveness. The models were tested twice and the average value was taken as the result. The results of segmentation are displayed in Table 2.

Table 2: Comparison of segmentation results

	Global accuracy (%)	mIOU (%)	Dice coefficient (%)
Unet	99.2	84.2	81.2
MN_unet	99.3	87.2	88.2
MN_PPM_unet	99.3	87.5	90.2
MN_PPM_MD_unet	99.3	87.9	92.5

After the model adopts MobileNetV3_Large as the backbone network (MN_unet), the miou and dice coefficient of the model are greatly improved. The miou is improved by about three, and the Dice coefficient is improved by about 7. After adding the PPM module, the model (MN_PPM_unet) has a rise in miou by 0.3 and a rise in dice coefficient by 2. This is because the Pyramid Pooling Module (PPM) can perform multi-scale pooling operations on the input feature maps to capture contextual information at different scales. It fuses the result after multi-scale pooling with the original feature map, effectively synthesizing the contextual information of different scales. This improves the accuracy of the model when classifying at the pixel level. We modified the decoder, and the miou and dice coefficient of the model (MN_PPM_MD_unet) are improved by 0.4 and 2.3, respectively. The modified decoder enhances the model by focusing on different feature channels, further improving the segmentation performance. The model training process was accelerated by utilizing the parameters of the pre-trained model on the ImageNet-1K. From Fig. 11a, we can see that the dice coefficients of the improved model can stabilize faster in the first 50 epochs compared to the original Unet model. And it is easy to see from Fig. 11b that the iou of the models will be more stable after the 150th epoch. This may be because migration learning accelerates the model's learning process, making it possible to achieve better performance even in the case of data scarcity.

We also tested the performances of using different convolutional networks as the backbone of the Unet and two common models in the field of image segmentation, fcn [35] and deeplabv3 [36], trained on this corn weed dataset. fcn and deeplabv3 both use resnet50 [37] as the backbone network. And the Unet using different backbones all use the number of channels interaction place as the input for skip connections, use the same parameter settings for training, and use the pretraining weights trained on ImageNet-1K for migration learning. The models were tested twice, and the average value was taken as the result. By comparing MobileNetV3_Large [32], squeezenet [38], convenxt_tiny [39], efficientnetv2_s [40] as backbone networks. From Table 3, we can see that convnext_unet has the highest miou on this dataset, but has the highest number of parameters. MobileNetV3_unet has the lowest number of parameters and has a higher miou than squeezenet_unet and efficientnetv2s_unet. Using ImageNet-1K as the backbone network can reduce the complexity of the model while ensuring the level of segmentation accuracy.

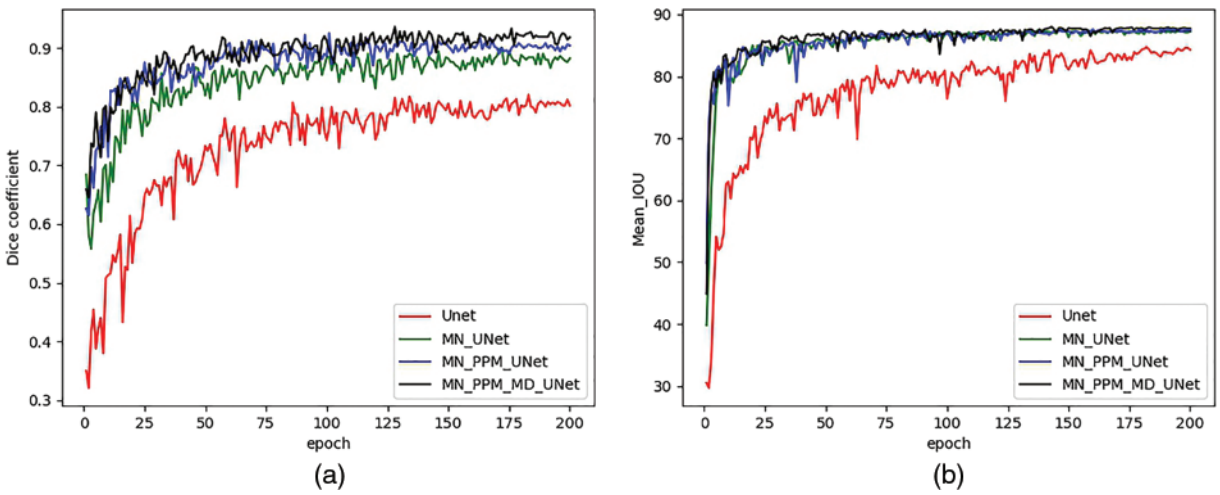

Figure 11: Dice coefficients and mIOUs of models

Table 3: Segmentation results of MobileNetV3_unet, unet, squeezenet_unet, efficientnetv2s_unet, convnext_unet and ours

	Global accuracy (%)	mIOU (%)	Number of parameter (M)	FPS	FLOPs (G)
unet	99.2	84.3	4.32	25.76	7.76
MobileNetV3_unet	99.3	87.2	3.45	81.71	0.64
squeezenet_unet	99.3	86.9	9.69	22.00	12.85
convnext_unet	99.4	88.1	38.27	21.41	9.16
efficientnetv2s_unet	98.9	79.3	21.73	33.50	3.94
fcn_resnet50	99.2	84.7	32.95	12.16	26.58
deeplabv3_resnet50	99.2	85.2	39.64	10.87	31.41
ours	99.3	87.9	3.79	58.90	0.79

Meanwhile, it can be found that the final optimized model has fewer parameters than the original Unet's, and miou is second only to convnext_unet. It shows that our model ensures that the results of the model segmentation are more consistent with the actual image in the case of fewer parameters. Fig. 12 shows that our network converges faster than the other five Unet models using different backbones. The dice coefficient of our model is second only to convnext_unet and higher than the other four Unet models using different backbones. This means that our model can accurately segment the corn and weed from the background, and the predicted boundaries overlap well with the actual boundaries. Regarding the computational complexity of the model, our model has the second lowest GFLOPs after MobileNetV3_unet, with a value of 0.79, which indicates that our model is easy to deploy and run in resource-constrained environments. We conducted our tests on a computer with a single 3050 ti graphics card and Win11 operating system to evaluate the frames-per-second (FPS) performance of different models with that card. The results show that our model can achieve a transmission rate of about 58.9 frames per second when using a 3050 ti graphics card.

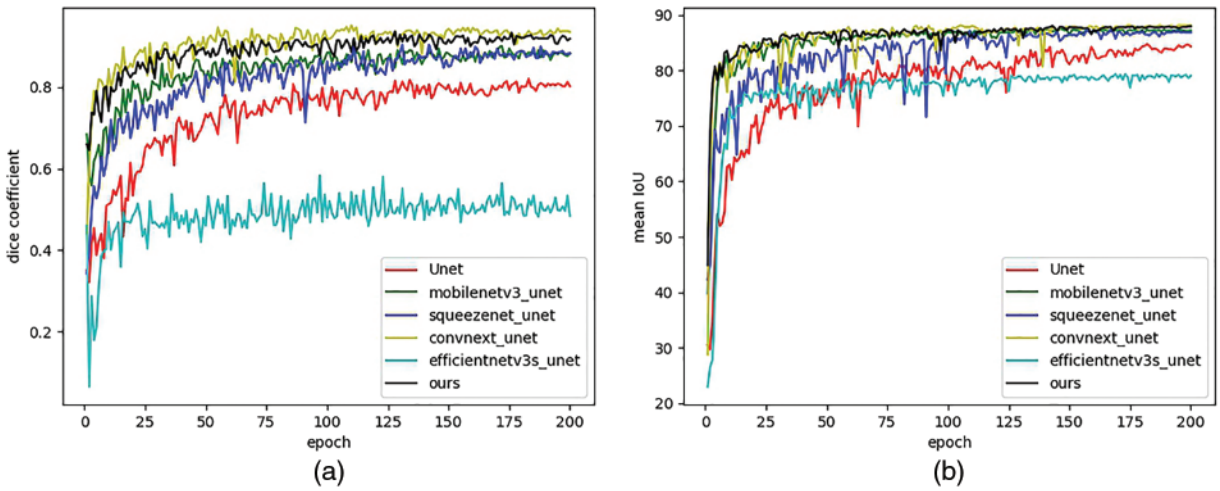

Figure 12: Dice coefficients and mIOUs of six networks

The trained model was used to segment images taken in a natural environment. Fig. 13 shows the segmented images of five plants. Among them, the sedges are very similar to the hay in the background, which causes the model to mistake the parts of sedges for other plants when segmenting. When segmenting the image of bluegrasses, the model may also mistake some of the pixels of the occluders for those of the same plant due to the occlusion of other plants in the background. The optimization of the model and the preprocessing of the data during training both enable the network to learn more about the corn weed images. From Fig. 13, we can find that the segmentation effect of the optimized model is better than that of the unet. We found that since the chenopodium albums have large leaves and small stems, the models may not be able to segment the hollows between the leaves. We can see from the figure that the optimized model can segment the hollows between the leaves of the chenopodium album better than the unet. In Table 4, we give the values of iou for each plant. In contrast, our model can efficiently segment the images of each plant while keeping the number of parameters and computational complexity low. Overall, our model can meet the performance and resource requirements of the model in the cornfield scenario.

Table 4: IOUs of different plants

	Corn	Bluegrass	Chenopodium album	Cirsium setosum	Sedge
unet	78.6	78.1	93.4	90.5	65.6
MobileNetV3_unet	89.1	82.1	93.7	93.3	65.8
squeezenet_unet	87.0	79.4	93.7	92.1	69.6
convnext_unet	87.7	83.2	94.3	94.7	69.2
efficientnetv2s_unet	71.5	75.4	84.5	88.1	57.2
fcn_resnet50	88.8	78.4	92.9	92.0	57.2
deeplabv3_resnet50	90.0	78.9	92.6	92.5	58.1
ours	90.0	82.2	93.4	93.7	68.5

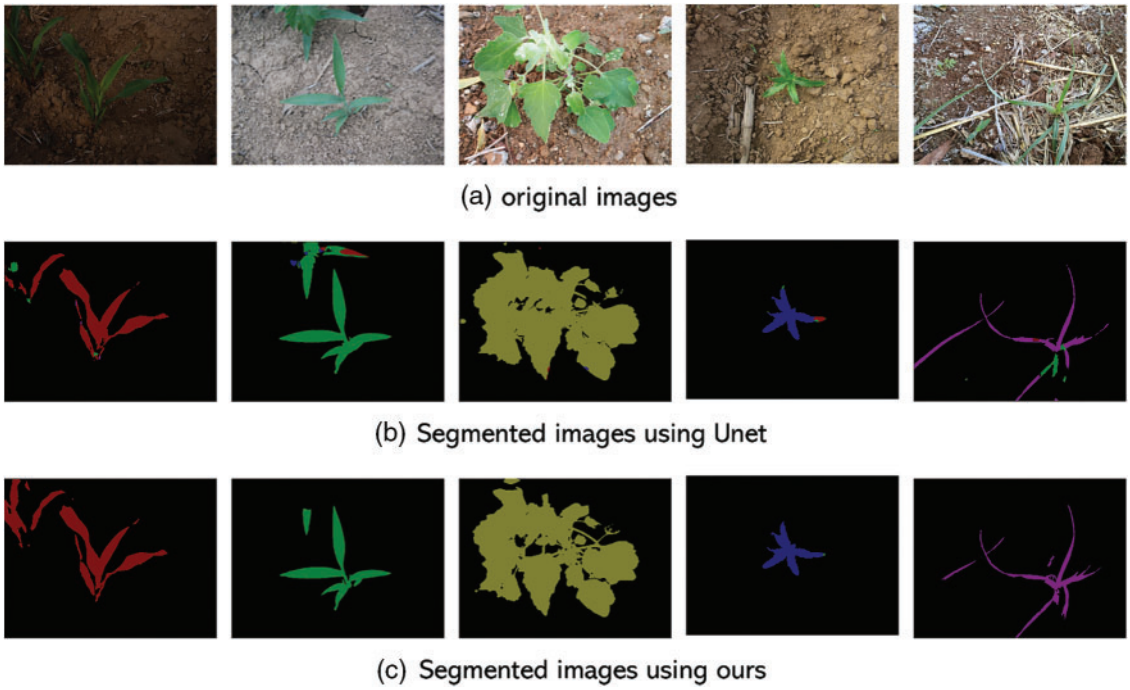

Figure 13: Segmented images of the original unet and ours

5 Conclusion

In this paper, we optimize the Unet model to make it more suitable for the semantic segmentation of corn and weed scene images. We hope this improvement will positively impact operation automation by enhancing the perception of agricultural machinery and robots in the farm environment to perform various tasks more accurately and efficiently. Part of MobileNetV3_Large is introduced as the compression network to enhance the extraction of image features. The SE mechanism is mapped into the encoder, and the encoder's activation function is modified to make the model more efficient. The skip connection introduces the PPM module to reduce the semantic error generated when connecting directly, improve the extraction of local detail information of the model, and use the bottleneck structure in the tail instead of the original upsampling. We preprocessed and tested in the small dataset collected; experimental results show that the improved model outperforms the original unet for corn weed image segmentation. This paper aims to investigate lightweight image segmentation methods applicable to corn fields. However, the experiment also has some drawbacks, such as the model not being able to segment the occluded plants wholly in the case of occlusion and not being able to segment the target well in the case of dim light. The results have only been validated on a single small dataset. It is crucial to adjust the network's training hyperparameters based on the specific dataset being processed, so the transferability of model weight parameters to other datasets cannot be fully guaranteed, and exchanging the backbone can yield good results. In future research, the model needs to be further validated and optimized.

Acknowledgement: The authors would like to express their gratitude to all the anonymous reviewers and the editorial team for their valuable feedback and suggestions.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Y. Zuo, W. W. Li; data collection: W. W. Li; analysis and interpretation of results: Y. Zuo, W. W. Li; draft manuscript preparation: Y. Zuo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Part of the publicly available weed dataset for this study can be available at the following url: https://github.com/zhangchuanyin/weed-datasets.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.-Ass. Intervent.*, Munich, Germany, 2015, pp. 234–241.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[3] H. Wang *et al.*, "MFBP-UNet: A network for pear leaf disease segmentation in natural agricultural environments," *Plants*, vol. 12, no. 18, pp. 3209, 2023. doi: 10.3390/plants12183209.

[4] Z. Sun, J. Yao, and Z. Jiang, "U-net with residual network and mask multi-task learning for plant leaf segmentation," in *IEEE Int. Conf. Sens., Electron. Comput. Eng.*, 2023, pp. 786–790.

[5] Y. Liu, J. Shen, L. Yang, G. Bian, and H. Yu, "ResDO-UNet: A deep residual network for accurate retinal vessel segmentation from fundus images," *Biomed. Signal Process. Control*, vol. 79, no. 3, pp. 104087, 2023. doi: 10.1016/j.bspc.2022.104087.

[6] Y. Gao, H. Cao, W. Cai, and G. Zhou, "Pixel-level road crack detection in UAV remote sensing images based on ARD-Unet," *Measurement*, vol. 219, no. 11045, pp. 113252, 2023. doi: 10.1016/j.measurement.2023.113252.

[7] C. Liu, J. Su, L. Wang, S. Lu, and L. Li, "LA-DeepLab V3+: A novel counting network for pigs," *Agriculture*, vol. 12, no. 2, pp. 284, 2022. doi: 10.3390/agriculture12020284.

[8] K. Li, L. Zhang, B. Li, S. Li, and J. Ma, "Attention-optimized DeepLab V3+ for automatic estimation of cucumber disease severity," *Plant Methods*, vol. 18, no. 1, pp. 109, 2022. doi: 10.1186/s13007-022-00941-8.

[9] H. Zhang, H. Luo, W. Li, and R. Qin, "A rapid image semantic segment method based on Deeplab V3+," in *Proc. 3rd Int. Conf. Artif. Intell.*, China, 2022, pp. 191–197.

[10] J. Sun *et al.*, "MASA-SegNet: A semantic segmentation network for polsar images," *Remote Sens.*, vol. 15, no. 14, pp. 3662, 2023. doi: 10.3390/rs15143662.

[11] S. Ahmed and M. K. Hasan, "Twin-SegNet: Dynamically coupled complementary segmentation networks for generalized medical image segmentation," *Comput. Vis. Image Understanding*, vol. 240, no. 1, pp. 103910, 2024. doi: 10.1016/j.cviu.2023.103910.

[12] V. N. Pattwakkar, S. Kamath, M. Kanabagatte Nanjundappa, and R. Kadavigere, "Automatic liver tumor segmentation on multiphase computed tomography volume using SegNet deep neural network and K-means clustering," *Int. J. Imaging Syst. Technol.*, vol. 33, no. 2, pp. 729–745, 2023. doi: 10.1002/ima.22816.

[13] Y. Xia, Y. Li, Q. Ye, and J. Dong, "Image segmentation for blind lanes based on improved SegNet model," *J. Electron. Imaging*, vol. 32, no. 1, pp. 013038, 2023. doi: 10.1117/1.JEI.32.1.013038.

[14] X. Li *et al.*, "Soybean leaf estimation based on RGB images and machine learning methods," *Plant Methods*, vol. 19, no. 1, pp. 1–16, 2023. doi: 10.1186/s13007-023-01023-z.

[15] X. Xue, Q. Luo, M. Bu, Z. Li, S. Lyu and S. Song, "Citrus tree canopy segmentation of orchard spraying robot based on RGB-D image and the improved DeepLabv3+," *Agronomy*, vol. 13, no. 8, pp. 2059, 2023. doi: 10.3390/agronomy13082059.

[16] Z. Luo, W. Yang, R. Gou, and Y. Yuan, "TransAttention U-Net for semantic segmentation of poppy," *Electronics*, vol. 12, no. 3, pp. 487, 2023. doi: 10.3390/electronics12030487.

[17] J. You, W. Liu, and J. Lee, "A DNN-based semantic segmentation for detecting weed and crop," *Comput. Electron. Agric.*, vol. 178, pp. 105750, 2020. doi: 10.1016/j.compag.2020.105750.
[18] N. Islam *et al.*, "Early weed detection using image processing and machine learning techniques in an Australian chilli farm," *Agriculture*, vol. 11, no. 5, pp. 387, 2021. doi: 10.3390/agriculture11050387.
[19] H. Yu, M. Che, and Y. Ma, "Research on weed identification in soybean fields based on the lightweight segmentation model DCSAnet," *Front. Plant Sci.*, vol. 14, pp. 1268218, 2023. doi: 10.3389/fpls.2023.1268218.
[20] S. D. Khan, S. Basalamah, and A. Lbath, "Weed-Crop segmentation in drone images with a novel encoder-decoder framework enhanced via attention modules," *Remote Sens.*, vol. 15, no. 23, pp. 5615, 2023. doi: 10.3390/rs15235615.
[21] J. Weyler, T. Läbe, F. Magistri, J. Behley, and C. Stachniss, "Towards domain generalization in crop and weed segmentation for precision farming robots," *IEEE Rob. Autom. Lett.*, vol. 8, no. 6, pp. 3310–3317, 2023. doi: 10.1109/LRA.2023.3262417.
[22] A. Picon, M. G. San-Emeterio, A. Bereciartua-Perez, C. Klukas, T. Eggers and R. Navarra-Mestre, "Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets," *Comput. Electron. Agric.*, vol. 194, pp. 106719, 2022. doi: 10.1016/j.compag.2022.106719.
[23] C. Yun, Y. H. Kim, S. J. Lee, S. J. Im, and K. R. Park, "WRA-Net: Wide receptive field attention network for motion deblurring in crop and weed image," *Plant Phenomics*, vol. 5, pp. 0031, 2023. doi: 10.34133/plantphenomics.0031.
[24] X. Z. Hu, W. S. Jeon, and S. Y. Rhee, "Sugar beets and weed detection using semantic segmentation," *Int. Conf. Fuzzy Theory its Appl.*, vol. 7, pp. 1–4, Nov. 2022. doi: 10.1109/iFUZZY55320.2022.9985222.
[25] L. L. Janneh, Y. Zhang, Z. Cui, and Y. Yang, "Multi-level feature re-weighted fusion for the semantic segmentation of crops and weeds," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, pp. 101545, Jun. 2023. doi: 10.1016/j.jksuci.2023.03.023.
[26] Y. H. Kim, S. J. Lee, C. Yun, S. J. Im, and K. R. Park, "LCW-Net: Low-light-image-based crop and weed segmentation network using attention module in two decoders," *Eng. Appl. Artif. Intell.*, vol. 126, no. 11, pp. 106890, Nov. 2023. doi: 10.1016/j.engappai.2023.106890.
[27] Q. Yang, Y. Ye, L. Gu, and Y. Wu, "MSFCA-Net: A multi-scale feature convolutional attention network for segmenting crops and weeds in the field," *Agriculture*, vol. 13, no. 6, pp. 1176, Jun. 2023. doi: 10.3390/agriculture13061176.
[28] K. Zou, H. Wang, T. Yuan, and C. Zhang, "Multi-species weed density assessment based on semantic segmentation neural network," *Precis. Agric.*, vol. 24, no. 2, pp. 458–481, Apr. 2023. doi: 10.1007/s11119-022-09953-9.
[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.
[30] A. Howard *et al.*, "Searching for MobileNetV3," in *IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[32] D. A. Pramudhita, F. Azzahra, I. K. Arfat, R. Magdalena, and S. Saidah, "Strawberry plant diseases classification using CNN based on MobileNetV3-large and EfficientNet-B0 architecture," *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, vol. 9, no. 3, pp. 522–534, 2023.
[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[34] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imaging.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020. doi: 10.1109/TMI.2019.2959609.
[35] K. L. Lam, A. Abdullah, and D. Albashish, "Ensemble of fully convolutional neural networks with end-to-end learning for small object semantic segmentation," *Lect. Notes Networks Syst.*, vol. 642, pp. 125–135, 2023. doi: 10.1007/978-3-031-26889-2.

[36] Y. Heryadi, E. Irwansyah, E. Miranda, H. Soeparno, and K. Hashimoto, "The effect of resnet model as feature extractor network to performance of DeepLabV3 model for semantic satellite image segmentation," in *IEEE Asia-Pacific Conf. Geosci., Electron. Remote Sens. Technol.*, 2020, pp. 74–77.

[37] A. V. Ikechukwu, S. Murali, R. Deepuand, and R. C. Shivamurthy, "ResNet-50 *vs.* VGG-19 *vs.* training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Global Trans. Proc.*, vol. 2, no. 2, pp. 375–381, 2021. doi: 10.1016/j.gltp.2021.08.027.

[38] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size," arXiv:1602.07360, 2016.

[39] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, "A convnet for the 2020s," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.

[40] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *ICML*, Jul. 2021, pp. 10096–10106.