agronomy

MDPI

Article

Citrus Tree Canopy Segmentation of Orchard Spraying Robot Based on RGB-D Image and the Improved DeepLabv3+

Xiuyun Xue [1,2,3,4], Qin Luo [1], Maofeng Bu [1], Zhen Li [1,2,3,4,*], Shilei Lyu [1,2,3,4] and Shuran Song [1,3,4]

[1] College of Electronic Engineering (College of Artificial Intelligence), South China Agricultural University, Guangzhou 510642, China; xuexiuyun@scau.edu.cn (X.X.); luoqin@stu.scau.edu.cn (Q.L.); 20223142029@stu.scau.edu.cn (M.B.); lvshilei@scau.edu.cn (S.L.); songshuran@scau.edu.cn (S.S.)
[2] Pazhou Lab, Guangzhou 510330, China
[3] Division of Citrus Machinery, China Agriculture Research System of MOF and MARA, Guangzhou 510642, China
[4] Guangdong Provincial Agricultural Information Monitoring Engineering Technology Research Center, Guangzhou 510642, China
* Correspondence: lizhen@scau.edu.cn

Abstract: The accurate and rapid acquisition of fruit tree canopy parameters is fundamental for achieving precision operations in orchard robotics, including accurate spraying and precise fertilization. In response to the issue of inaccurate citrus tree canopy segmentation in complex orchard backgrounds, this paper proposes an improved DeepLabv3+ model for fruit tree canopy segmentation, facilitating canopy parameter calculation. The model takes the RGB-D (Red, Green, Blue, Depth) image segmented canopy foreground as input, introducing Dilated Spatial Convolution in Atrous Spatial Pyramid Pooling to reduce computational load and integrating Convolutional Block Attention Module and Coordinate Attention for enhanced edge feature extraction. MobileNetV3-Small is utilized as the backbone network, making the model suitable for embedded platforms. A citrus tree canopy image dataset was collected from two orchards in distinct regions. Data from Orchard A was divided into training, validation, and test set A, while data from Orchard B was designated as test set B, collectively employed for model training and testing. The model achieves a detection speed of 32.69 FPS on Jetson Xavier NX, which is six times faster than the traditional DeepLabv3+. On test set A, the mIoU is 95.62%, and on test set B, the mIoU is 92.29%, showing a 1.12% improvement over the traditional DeepLabv3+. These results demonstrate the outstanding performance of the improved DeepLabv3+ model in segmenting fruit tree canopies under different conditions, thus enabling precise spraying by orchard spraying robots.

Keywords: improved DeepLabv3+; attention mechanism; citrus tree canopy; orchard spraying robot; RGB-D detector

Citation: Xue, X.; Luo, Q.; Bu, M.; Li, Z.; Lyu, S.; Song, S. Citrus Tree Canopy Segmentation of Orchard Spraying Robot Based on RGB-D Image and the Improved DeepLabv3+. *Agronomy* **2023**, *13*, 2059. https://doi.org/10.3390/agronomy13082059

Academic Editors: Jun Ni, Lei Feng and Lvhua Han

Received: 3 July 2023
Revised: 27 July 2023
Accepted: 31 July 2023
Published: 3 August 2023

1. Introduction

Citrus is one of the most important agricultural trade commodities worldwide, and China is among the countries with the largest citrus cultivation areas [1]. In citrus orchard management, orchard plant protection is crucial for controlling pests and diseases and ensuring fruit quality. However, traditional manual spraying methods still dominate orchard plant protection, resulting in low efficiency and significant waste of pesticide solution and posing a risk of pesticide poisoning for applicators. With the continuous development of agricultural intelligence technology, orchard spraying robots have begun transitioning from research and development to the experimental stage and will be deployed in large-scale agricultural production processes [2]. Precision spraying is one of the core technologies for orchard spraying robots, and accurate recognition and segmentation of the target are prerequisites for achieving precision spraying [3]. Therefore, an algorithm capable of accurately segmenting target tree canopies in an orchard environment is needed to support efficient pesticide application by orchard spraying robots.

With the development of sensor technology, fruit tree canopy detection methods have become increasingly diverse, including various methods such as LiDAR, ultrasonics, and multispectral imaging [4–8]. However, these methods generally have shortcomings, such as high hardware costs, strict terrain requirements, and susceptibility to environmental factors [9]. To address these issues, some researchers have attempted to use low-cost image processing and machine vision methods for fruit tree canopy recognition and segmentation, achieving certain results. For example, Xiao et al. [10] proposed a precision orchard spraying technology that used color and depth information collected by KinectV1 (Microsoft Corporation, Redmond, WA, USA) sensors to extract the leaf wall area of fruit trees. Experiments on peach trees, apricot trees, and grapevines demonstrated that this technology could improve pesticide application efficiency and reduce environmental pollution. Gao et al. [11] used KinectV1 sensors to collect color and depth image information of grapevines and employed threshold segmentation techniques to accurately extract leaf wall information, calculating leaf wall density with minimal error. However, the performance of these methods is influenced by many factors, such as crop type, weeds or other objects near the canopy, and different lighting conditions. Moreover, the Light Coding technology used by KinectV1 is also easily affected by strong light sources. Therefore, this study adopts the KinectV2 (Microsoft Corporation, Redmond, WA, USA) with a Time-of-Flight (TOF) depth sensor as the image acquisition device and investigates an effective algorithm suitable for citrus orchard environments.

With the rapid development of deep learning theory and the continuous improvement of computer performance, the release of large-scale training datasets such as ImageNet [12] and VSPW [13] has enabled Convolutional Neural Networks (CNNs) to perform remarkably well in tasks such as object detection, image segmentation, and object classification [14]. In recent years, deep learning technology has been widely applied and studied in the agricultural field [15–19]. Research has shown that deep learning methods have high accuracy in the recognition and segmentation of tree canopies in fruit tree images [20–22]. Sun et al. [23] proposed an automatic detection method for flowers in fruit tree canopies, utilizing a pre-trained semantic segmentation network and active contour model, achieving a balanced F-score of 89.6% on an apple dataset. This method can be widely applied to fruit tree cultivation, providing quick and accurate estimates of flowering degrees for fruit growers. Anagnostis et al. [24] proposed an orchard tree segmentation method based on an improved U-Net network model combined with remote sensing technology, achieving an accuracy of 87% in performance tests and successfully segmenting fruit tree canopies. Cao et al. [25] proposed a semantic segmentation model based on an improved Yolo v7 to address the problem of segmenting tree images in simple and complex backgrounds. The model introduced SENet to reduce the acquisition of erroneous features and proposed a weighted loss function to solve the problem of class imbalance in complex backgrounds. In the final performance evaluation, the model achieved a mean intersection over union of 91.17% and 90.23% in simple and complex backgrounds, respectively. Shi et al. [26] introduced a lightweight model, SEMD, based on DeepLabv3+. This model enhances feature extraction capability by introducing the attention mechanism of the SENet module. It achieves a mean intersection over union (*mIoU*) of 91.78% and 86.90% for semantic segmentation of tree images of different varieties and categories under simple and complex backgrounds, respectively. Moreover, the detection speed of this model reaches 160 ms.

In summary, the application of tree crown segmentation in real-time precision spraying in orchards currently faces two challenges: detection speed and accuracy. In terms of speed, it requires a lightweight model that can be deployed on embedded platforms and meet the requirements of real-time detection. However, in terms of accuracy, lightweight models do not achieve precise segmentation of tree crowns in complex orchard backgrounds from RGB images, thus failing to achieve satisfactory performance. In contrast, RGB-D images offer greater advantages. They not only retain the information from RGB images but also contain depth data. By performing threshold segmentation on the depth values, complex orchard backgrounds can be eliminated, thereby improving the adaptability of lightweight models

in this environment. Therefore, this paper proposes a lightweight network segmentation model based on the DeepLabv3+ [27] model, using RGB-D (Red, Green, Blue, Depth) images as input data for citrus tree crown area segmentation to provide technical support for precise pesticide application by orchard spraying robots. The main contributions of this paper are as follows:

1. We collected and prepared a set of RGB-D images of the citrus canopy from two citrus orchards, A and B, in different periods and regions. In Orchard A, we collected 540 groups of images and divided them into training set, validation set, and test set A. Meanwhile, in Orchard B, we collected 100 groups of images as test set B. The data collection and preparation process are described in detail in Section 2. The improved DeepLabv3+ achieved *mIoU* of 95.62% and 92.29% on test sets A and B, respectively, demonstrating its sufficient generalization performance.
2. This paper improved the model's segmentation accuracy in complex backgrounds by using depth information to remove redundant orchard background from RGB-D images. In test set A, the *mIoU* of using RGB-D images increased by 4.67% compared to using only RGB images.
3. Through the introduction of the Depthwise Separable Convolution (DSC) [28] method, the Convolutional Block Attention Module (CBAM) [29], and the Coordinate Attention (CA) [30] mechanism module, we improved the Atrous Spatial Pyramid Pooling (ASPP) [31] module of DeepLabv3+. This not only reduces the computational complexity of the model but also enhances the segmentation accuracy of the network model for citrus tree crowns. In test set A, compared to the original DeepLabv3+ model, the improved DeepLabv3+ model achieved a 2.43% increase in *mIoU*.
4. We replaced the backbone network of DeepLabv3+ with MobileNetV3-Small [32], which significantly accelerated the detection speed of the network model. Furthermore, this model has a parameter count of only 3.11 M, which makes it well-suited for applications on embedded platforms. The improved model achieved a detection speed of 32.69 FPS for citrus tree canopy segmentation tasks on the Jetson Xavier NX device, effectively meeting the segmentation requirements for real-time precision spraying in orchards.

2. Materials and Methods

2.1. Image Acquisition

The citrus image acquisition system, consisting of the KinectV2 sensor and Jetson Xavier NX device, was employed in this study. The data collection and creation process is illustrated in Figure 1. The KinectV2 sensor served as the image acquisition sensor for citrus trees, while the Jetson Xavier NX device was utilized for program execution and image storage. The entire data collection process was executed in the Python 3.7.14 environment. To ensure that each image captured the complete canopy of the target fruit tree and also met the spraying range of the spraying robot, the KinectV2 sensor was installed on a stand at a height of 1.30 m above the ground. It was positioned 2.3 m away from the center of the tree row, with the lens parallel to the target fruit tree for image acquisition. The height and distance remained constant throughout the image acquisition process. The KinectV2 sensor can acquire RGB images (at a resolution of 1920 × 1080), depth images (at a resolution of 512 × 424; effective depth range of 0.5–5 m), and point cloud data using the RGB and TOF depth sensors.

In this study, image data was collected from a citrus orchard (referred to as Orchard A) at the Citrus Science Research Institute in Ganzhou, China (Lat: 25°46′46.1136″ N, Long: 114°51′38.7540″ E) from 23 August to 26 August 2022. The acquisition times were from 7:00 am to 10:30 am and 4:00 pm to 6:00 pm local time. The images were captured under varying light intensities (normal light, stronger light, and weaker light) due to changes in sunlight and cloud cover, thereby enhancing the sample diversity of the image data. The citrus tree varieties were Ganzhou navel oranges, including 3-year-old hedge-style, 3-year-old dwarf-style, and 5-year-old conventional high-density planting-style

citrus trees. A total of 540 color images and 540 depth images were collected using the KinectV2 sensor. All images were captured at 540 distinct positions to ensure that the target citrus trees in each image were different. As shown in Figure 2, the images are arranged from left to right, representing trellis-style, dwarf-style, and high-density planted images, respectively. From top to bottom, the images depict normal lighting, strong lighting, and weak lighting conditions. Furthermore, to enhance the evaluation of the model's generalization performance, supplementary test images of the conventional high-density planted citrus trees were collected at the DaJu Fruit Industry Pingtan Plantation (referred to as Orchard B) in Huizhou, Guangdong, China (Lat: 23°2′27.528″ N, Long: 114°36′42.3″ E) from 20–21 March 2023. Compared with Orchard A, Orchard B is located in a coastal region with different temperature and humidity. It was in the flowering period during collection and had more dense canopy branches and leaves. The citrus variety was also different, which was the 4-year-old Hongjiang orange. Therefore, choosing Orchard B as a supplementary test set can effectively test the model's ability to extract and recognize citrus canopy features under different environmental conditions. Moreover, the data collection process in Orchard B was consistent with that in Orchard A, with a total of 100 color images and 100 depth images captured, as shown in Figure 3.

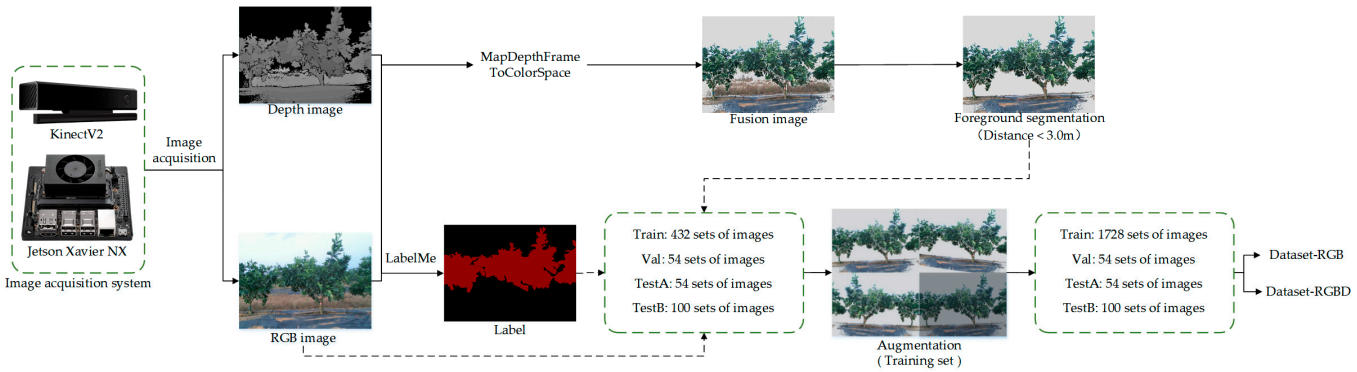

Figure 1. Flowchart of Dataset Collection and Creation Process.

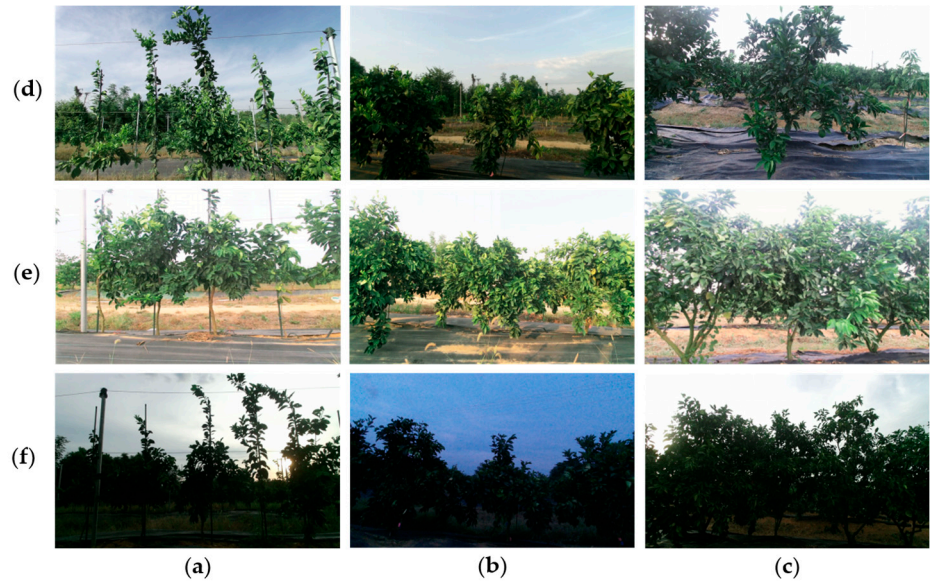

Figure 2. Images of citrus trees in Orchard A under different situations: (**a**) hedge-style; (**b**) dwarf-style; (**c**) dense planting-style; (**d**) normal lighting; (**e**) strong lighting; (**f**) weak lighting.

Figure 3. Images of citrus trees collected in Orchard B.

This paper acquired separate color and depth image data through the KinectV2 sensor; however, due to the different resolutions of the two, the images needed to be aligned before creating the training dataset. Therefore, the MapDepthFrameToColorSpace function from the KinectV2 software development kit (version number: 2.0.1409) was used, and color images were adjusted to match depth images so that each pixel in the depth image corresponds to an RGB value. This process returns an array of colored point clouds, resulting in the final RGB-D image. Since the working area of the orchard spray robot is the nearest part of the tree canopy, the background and distant citrus trees in the collected citrus images are not within the target area range. Thus, it was necessary to threshold-segment the depth data of the RGB-D images and exclude image information outside the target area range according to Equation (1), retaining image information within 3 m for creating the citrus tree foreground segmentation image dataset. Subsequently, we manually annotated the citrus foreground images using the open-source annotation tool LabelMe (version number: 5.1.1) [33] to establish a standard semantic annotation dataset. The citrus canopy in the image is marked as red as the target area, and the rest is marked as black as the background. The generated label images correspond one-to-one with the original images and are saved in the VOC format, providing a reliable data foundation for further research.

$$I_d(x,y) = \begin{cases} D_1(x,y), & if \ \ D_1(x,y) > 3000 \\ 0, & if \ \ D_1(x,y) \leq 3000 \end{cases} \quad (1)$$

$D_1(x,y)$ represents the pixel values $\{x \in (1,W)\}$ and $\{y \in (1,H)\}$ in the depth image; W represents the pixel value corresponding to the image width; H represents the pixel value corresponding to the image height.

In this study, the dataset from Orchard A, consisting of 540 sets of images (each set containing one RGB image, one RGB-D image, and one label image corresponding to each other), was randomly divided into training, validation, and test sets A in an 8:1:1 ratio. The dataset from Orchard B, comprising 100 sets of images, was reserved as test set B to evaluate the generalization performance of the proposed model. In terms of data augmentation, considering the situations that the spraying robot may encounter in practical application, such as differences in the forward and backward movement directions, slight shaking of the acquisition system during motion, and variations in the brightness of the captured images, we applied methods such as image flipping, slight rotation (randomly rotating 5° to 10° left or right), and brightness transformation to augment the training and validation sets. This was done to expand the dataset, improve generalization ability, and enhance the robustness of the model. After augmentation, the training set consisted of 1728 image sets. The validation set, test set A, and test set B were not subjected to augmentation and comprised 54 sets, 54 sets, and 100 sets of images, respectively. Examples of the dataset are shown in Figure 4. From left to right are the original image, foreground segmentation image, and manually annotated image, where black represents the background and red represents the citrus canopy in the manually annotated image.

(**a**) (**b**) (**c**)

Figure 4. Example images from the dataset: (**a**) Original image; (**b**) Foreground segmentation image; (**c**) Manually labeled images.

2.2. Fruit Tree Canopy Segmentation Model

2.2.1. DeepLabv3+ Network Model

DeepLabv3+ [27] is a deep neural network from the DeepLab series specifically designed for image segmentation. It builds upon the improvements made in DeepLabv3 [34] to enhance segmentation accuracy and efficiency. DeepLabv3 improves the Atrous Spatial Pyramid Pooling (ASPP) module by incorporating convolutional kernels, multiple dilation rates, and multiple receptive fields, addressing the multi-scale challenge in image segmentation. Building upon this, the DeepLabv3+ model introduces global average pooling into the ASPP module to capture global semantic information. Additionally, inspired by the encoder–decoder structure, a simple decoder module is introduced to compress the lower-level feature maps. Furthermore, the feature maps processed by the ASPP module undergo upsampling to restore their original resolution. Through a series of operations such as 3×3 convolution and upsampling, the spatial information is gradually recovered, enabling finer boundary detection for semantic segmentation at the pixel level.

2.2.2. Introduction of Depth-Separable Convolution

The main function of convolutional layers is feature extraction. In DeepLabv3+, the ASPP module [31] combines dilated convolutions and spatial pyramid pooling to enable the CNN to capture multi-scale feature information. However, traditional convolutional layers that incorporate dilated convolutions still suffer from high computational complexity and excessive parameter count. To address this issue, this study introduces the method of depthwise separable convolution (DSC) [28]. The method consists of two parts: depthwise convolution and pointwise convolution. Depthwise convolution independently applies a 3×3 kernel to each input channel, realizing depth separation, as shown in Figure 5a. Pointwise convolution employs a 1×1 kernel to mix channel features, as illustrated in Figure 5b. DSC reduces the number of parameters, lowers computation, and improves the balance in spatial and depth directions, making it suitable for lightweight models. Therefore, we replace the convolutional networks in each branch with dilated convolutions in the ASPP module with depthwise separable convolution networks to reduce the computational complexity of dilated convolutions and enhance the detection efficiency of the model.

2.2.3. Introducing Attention Mechanism

Due to the limited richness of information scale in the ASPP module of DeepLabv3+, which only utilizes convolutional blocks with dilation rates of 6, 12, and 18, the extraction of semantic information for edge features is insufficient. Attention mechanisms play a crucial role in enhancing the capability of network models to extract semantic information and have been proven effective in image segmentation methods [35,36].

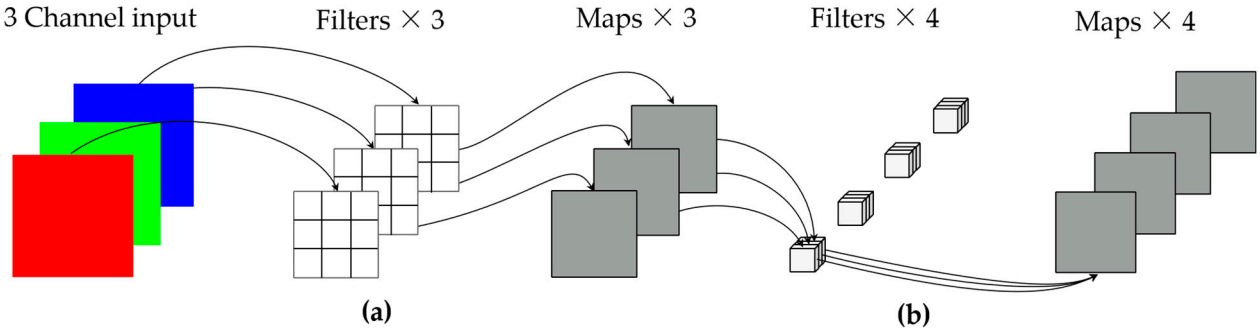

Figure 5. Depthwise Separable Convolution: (**a**) depthwise convolution; (**b**) pointwise convolution.

Woo et al. [29] proposed the Convolutional Block Attention Module (CBAM) mechanism, which fully utilizes channel attention and spatial attention to adjust the weights of feature maps, thereby highlighting important features and suppressing redundant ones. The operational process of the CBAM mechanism is illustrated in Figure 6. Firstly, the channel attention module performs average pooling and max pooling on the input feature map, resulting in two C-dimensional vectors that are fed into a multi-layer perceptron (MLP) with a hidden layer. This process generates two $1 \times 1 \times C$ channel attention maps. Finally, the two maps are added together and passed through an activation function to obtain the final channel attention map. Secondly, the spatial attention module performs average pooling and max pooling on the feature map processed by the channel attention module, resulting in two $H \times W$ matrices. These matrices are then concatenated along the channel dimension, resulting in a $2 \times H \times W$ feature map. The feature map subsequently undergoes convolutional layers and an activation function to obtain the final spatial attention map. Finally, the two attention maps are multiplied by the input feature map to achieve adaptive feature refinement. Therefore, the introduction of the CBAM mechanism enhances the network's ability to extract and focus on edge features, thus playing a positive role in fruit tree canopy segmentation tasks and improving the accuracy of semantic segmentation.

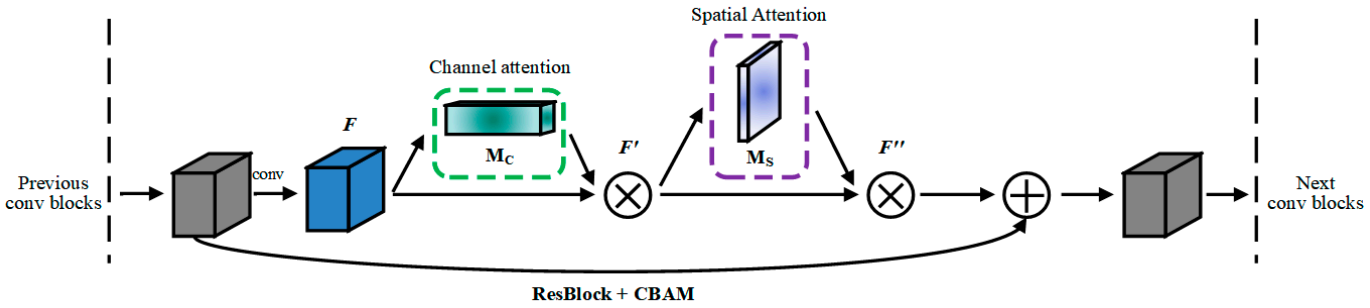

Figure 6. The convolutional block attention module.

Hou et al. [30] proposed the Coordinate Attention (CA) mechanism, which embeds positional information into channel attention to enable the network to capture information from larger regions while avoiding a significant increase in computational costs. The operational process diagram of the CA module is shown in Figure 7. Firstly, the input features undergo global average pooling in the horizontal and vertical directions, followed by channel compression using a 1×1 convolutional layer, batch normalization, and non-linear activation. The features are then split into horizontal and vertical sub-vectors. Subsequently, each sub-vector is processed with a 1×1 convolution and sigmoid activation to obtain the attention maps for the horizontal and vertical directions. Finally, these two attention maps are multiplied with the input feature tensor to obtain the output feature tensor of the CA module. Therefore, the introduction of the CA module enhances the network's focus on the shape and contour of the target, leading to more accurate segmentation of the target boundaries.

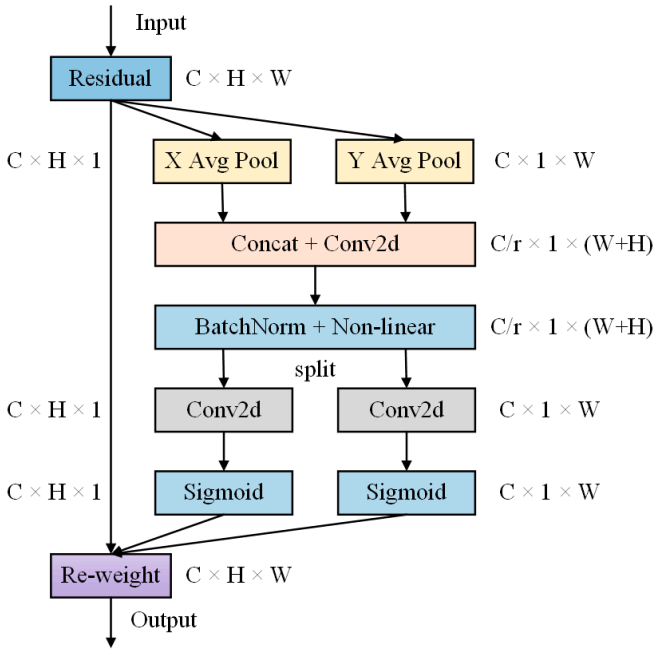

Figure 7. The coordinate attention mechanism.

Both the CBAM and CA modules aid in better capturing semantic feature information. In this study, we incorporate CBAM into the ASPP module, specifically in the branch with dilated convolutions, to enhance the performance of the network. Additionally, a CA module is added after the ASPP module to aid in the segmentation of tree canopies. The improved ASPP module structure is illustrated in Figure 8.

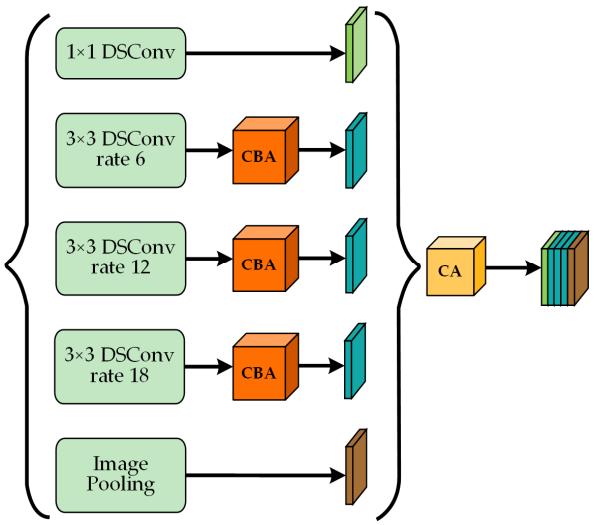

Figure 8. Improved ASPP structure diagram.

2.2.4. Backbone Feature Extraction Network

The traditional DeepLabv3+ model achieves high accuracy in semantic segmentation. Its backbone network, Xception [28], is a structure based on DSC that performs well in semantic segmentation tasks. However, Xception has a large number of network layers and a high number of parameters, which increases the complexity of the model, making it more difficult to train and decreasing the segmentation performance of the network.

For orchard spraying robots, real-time acquisition of crown information and pesticide decision-making require high segmentation efficiency of the network model. Therefore, in order to be deployed on embedded systems, we replace the Xception network with

a lightweight MobileNetV3-Small [32] network with DSC. MobileNetV3 is an efficient convolutional neural network for mobile devices released by Google in 2019. It uses the Squeeze-And-Excite module with a channel attention mechanism to reduce the number of parameters and adopts residual connections and depth separable structure to improve the performance of the network. MobileNetV3-Small is one of the lightweight versions of the MobilNetV3 network structure, with fewer parameters, making it more suitable for mobile vision applications. The network structure of the MobileNetV3-Small model is shown in Table 1.

Table 1. The network structure of the MobileNetV3-Small model.

Input	Operator	Exp Size	Out	SE	NL	S
$224^2 \times 3$	Conv2d, 3×3	-	16	-	HS	2
$112^2 \times 16$	Bneck, 3×3	16	16	√	RE	2
$56^2 \times 16$	Bneck, 3×3	72	24	-	RE	2
$28^2 \times 24$	Bneck, 3×3	88	24	-	RE	1
$28^2 \times 24$	Bneck, 5×5	96	40	√	HS	2
$14^2 \times 40$	Bneck, 5×5	240	40	√	HS	1
$14^2 \times 40$	Bneck, 5×5	240	40	√	HS	1
$14^2 \times 40$	Bneck, 5×5	120	48	√	HS	1
$14^2 \times 48$	Bneck, 5×5	144	48	√	HS	1
$14^2 \times 48$	Bneck, 5×5	288	96	√	HS	2
$7^2 \times 96$	Bneck, 5×5	576	96	√	HS	1
$7^2 \times 96$	Bneck, 3×3	576	96	√	HS	1
$7^2 \times 96$	Conv2d, 1×1	-	576	√	HS	1
$7^2 \times 576$	Pool, 7×7	-	-	-	-	1
$1^2 \times 576$	Conv2d 1×1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	Conv2d 1×1, NBN	-	1000	-	-	1

Note: "Input" denotes the size of the feature map input to each feature layer of MobileNetV3-Small; "Operator" denotes the layer structure through which each feature map will cross; "Exp size" denotes the number of channels after the inverse residual structure in the bottleneck stage; "Out" denotes the number of channels in the feature map after passing through the bottleneck; "SE" denotes whether the SE attention mechanism is introduced at this layer; "NL" denotes the type of activation function used, which can be either HS (h-swish) or RE (ReLU); NBN denotes no batch normalization; "S" denotes the step size used for each layer structure.

2.2.5. Improved Lightweight Design of DeepLabv3+

In this paper, we use MobileNetV3-Small as the backbone network for our proposed model, combined with the improved ASPP module. This network structure greatly reduces the number of parameters while ensuring segmentation accuracy and improves the segmentation efficiency of the model. The improved DeepLabv3+ network model structure is shown in Figure 9.

2.3. Transfer Learning

Transfer learning [37] is a method that applies knowledge and skills learned from one or multiple source tasks to a new target task. Its essence lies in identifying similarities between existing knowledge and new knowledge and leveraging this similarity to enhance learning efficiency and performance. Due to the lack of a large-scale publicly available dataset specifically for fruit tree canopy images and the risk of overfitting when training on small datasets, this study employs a transfer learning-based training approach. The MobileNetV3-Small backbone network model is pre-trained on the ImageNet dataset, allowing the transfer of substantial knowledge learned by MobileNetV3-Small to the dataset used in this study. This facilitates improved extraction of features from citrus tree canopy images.

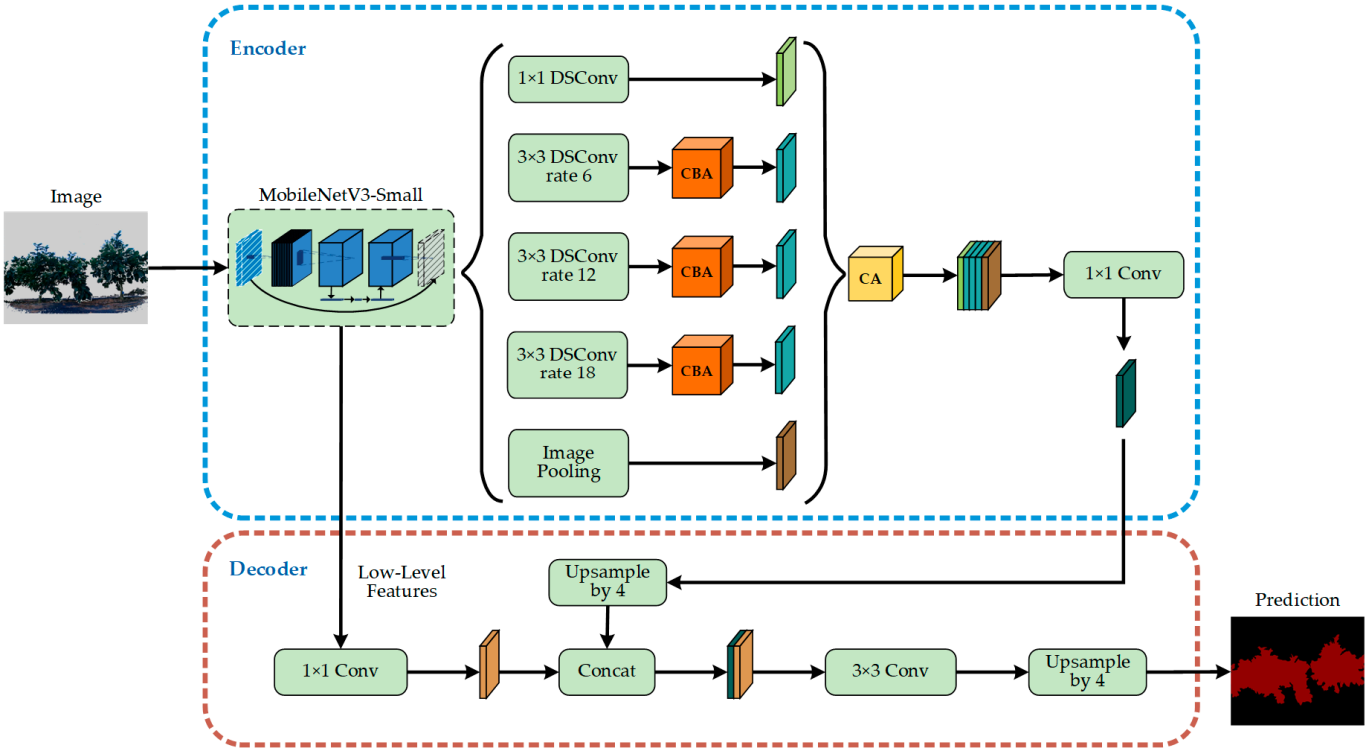

Figure 9. Improved DeepLabv3+ network. "DSConv" denotes depthwise separable convolution; "CBA" denotes convolutional block attention module; "CA" denotes coordinated attention mechanism; "Conv" denotes convolutional layer; "Concat" denotes concatenate layer.

2.4. Experimental Parameters Setup

In this paper, we used two different image datasets, namely, the original RGB images and the RGB-D images processed using the method described in Section 2.1, as the training, validation, and testing datasets for the neural network. Specifically, the training set, validation set, test set A, and test set B consisted of 1728, 54, 54, and 100 sets of images, respectively. The neural network training was conducted on a computer with an Intel(R) Xeon(R) CPU and Tesla T4-16GB GPU, running the Ubuntu 18.04.6 LTS operating system. The Pytorch 1.12.1 deep learning platform was chosen, and the neural network model in this paper was implemented using Python 3.7.14. To ensure real-time detection on the spray robot, experimental tests regarding the detection speed were conducted on the upper computer (Jetson Xavier NX) of the robot.

In this study, a transfer learning approach was employed to pre-train the MobileNetV3-Small backbone network on the ImageNet dataset. While keeping the pre-trained weights unchanged, adjustments were made to the parameters of the ASPP module in the DeepLabv3+ model to adapt to the citrus tree canopy image dataset. During the network training phase, we selected Adam as the optimizer and used the "Cross Entropy Loss Function" as the loss function of the model. Additionally, the input image size was uniformly adjusted to 512 × 512. The initial learning rate was set to 0.001, the minimum learning rate to 0.0001, the weight decay coefficient to 0.0001, and the batch size to 16. The experiment was conducted through 50 epochs of iterative training.

2.5. Canopy Segment Network Performance Evaluation

In this paper, the model recognition performance evaluation uses pixel accuracy (*Acc*), recall (*Re*), precision (*Pr*), mean Intersection over Union (*mIoU*), and balanced F-score (F_1) as evaluation metrics for experimental results. Additionally, to test the model's detection speed, it was deployed on the Jetson Xavier NX (NVIDIA, Santa Clara, CA, USA). The

detection speed was measured in frames per second (FPS). The specific calculation formulas for each evaluation metric are as follows:

$$Acc = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FN + \sum FP} \times 100\% \quad (2)$$

$$R_e = \frac{\sum TP}{\sum TP + \sum FN} \times 100\% \quad (3)$$

$$P_r = \frac{\sum TP}{\sum TP + \sum FP} \times 100\% \quad (4)$$

$$mIoU = \frac{1}{k+1} \times \sum \frac{\sum TP}{\sum TP + \sum FN + \sum FP} \times 100\% \quad (5)$$

$$F_1 = \frac{2 \times P_r \times R_e}{P_r + R_e} \times 100\% \quad (6)$$

where *TP* represents the number of pixels correctly segmented as citrus canopy regions; *TN* represents the number of pixels correctly labeled as other unrelated regions; *FP* represents the number of pixels mistakenly labeled as citrus canopy regions; *FN* represents the number of pixels of citrus canopy that are mistakenly labeled as other unrelated regions; $k + 1$ represents the number of image categories (including background), where $k = 1$ in this paper.

3. Results

3.1. Contrast with Transfer Learning

In this study, a comparative experiment was conducted between the improved DeepLabv3+ model utilizing MobileNetV3-Small as the backbone network with and without transfer learning. The training was performed using the RGB-D dataset, and the *mIoU* curve of the model training is shown in Figure 10. The performance evaluation of the models was conducted on test set A and test set B, and the results are presented in Table 2.

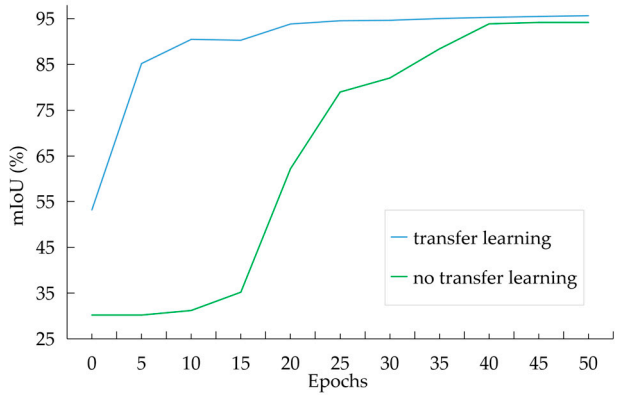

Figure 10. A comparison of the *mIoU* curves for transfer learning.

Table 2. The effect of applying transfer learning on the segmentation results of two different datasets.

DataSet	Transfer Learning	*Acc* (%)	*Re* (%)	*Pr* (%)	*mIoU* (%)	F_1 (%)
Test A	No	97.01	95.87	96.79	94.18	96.33
	Yes	97.79	97.98	97.06	95.62	97.52
Test B	No	91.68	88.98	93.47	84.60	91.17
	Yes	95.99	98.81	93.28	92.29	95.97

Note: The model used in the table is the modified DeepLabv3+ and the backbone network is MobileNetV3-Small. Test A and Test B represent test set A and test set B, respectively.

It can be seen from Figure 10 that without employing transfer learning, the initial value of *mIoU* is remarkably low, and its growth rate is relatively slow until the 15th epoch. Subsequently, between the 40th epoch, *mIoU* undergoes a phase of rapid growth and eventually stabilizes. On the other hand, it can be seen from Figure 10 that the MobileNetV3-Small network with transfer learning demonstrates relatively good feature extraction capability right from the initial stages due to the transfer of a substantial amount of knowledge from the ImageNet dataset to the dataset used in this study. It exhibits rapid growth even within the first 10 epochs and stabilizes after the 20th epoch. The results from test set A in Table 2 indicate that the model with transfer learning outperforms the model without transfer learning, showing improvements of 0.78% in pixel accuracy, 2.11% in recall, 0.27% in precision, 1.44% in *mIoU*, and 1.19% in F_1 score. Moreover, the evaluation results on test set B reveal a 7.69% increase in *mIoU* for the model with transfer learning compared to the model without transfer learning. The experimental results demonstrate that transfer learning effectively addresses the issue of slow model convergence caused by small sample datasets, enhances the generalization performance of the model, and slightly improves the segmentation accuracy.

3.2. Comparative Analysis of RGB and RGB-D Detectors Based on Improved DeepLabv3+

In this study, we trained the improved DeepLabv3+ canopy segmentation model using RGB and RGB-D datasets separately to compare the detection performance of RGB and RGB-D detectors. The performance evaluation results of the trained model on test set A are presented in Table 3. For the RGB and RGB-D detectors based on the improved DeepLabv3+, we calculated pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed. On the DeepLabv3+ canopy detectors for fruit trees, better detection performance was achieved using RGB-D compared to RGB. The RGB-D detector exhibited improvements of approximately 2.67%, 0.63%, 5.19%, 4.67%, and 2.99% in pixel accuracy, recall, precision, *mIoU*, and F_1, respectively, compared to the RGB detector. The detection speeds of the RGB and RGB-D detectors on embedded devices were approximately 32.58 FPS and 32.69 FPS, respectively, with negligible differences in detection speed. Therefore, both RGB and RGB-D detectors are suitable for real-time detection applications in precision spraying in orchards.

Table 3. Experimental results of RGB and RGB-D detectors based on an improved DeepLabv3+ model.

Type of Detector	*Acc* (%)	*Re* (%)	*Pr* (%)	*mIoU* (%)	F_1 (%)	Speed (FPS)
RGB	95.12	97.35	91.87	90.95	94.53	32.58
RGB-D	97.79	97.98	97.06	95.62	97.52	32.69

Note: The model used in the table is the modified DeepLabv3+, the backbone network is MobileNetV3-Small, and the test set A is used for the performance evaluation. The RGB detector uses the model trained from the RGB dataset, and the RGB-D detector uses the model trained from the RGB-D dataset.

We used RGB and RGB-D detectors for target detection and analyzed some examples. As shown in Figure 11, the red mask portion represents the segmentation results for the target area. The figure displays, from top to bottom, the segmentation of the target area using the improved DeepLabv3+ feature extractor under normal, low, and strong illumination conditions. Both RGB and RGB-D achieved good segmentation accuracy. However, when detecting in more complex background environments, the RGB-D detector had better segmentation accuracy than the RGB detector. This is because the RGB-D detector eliminated background environments before performing the segmentation task, making it less susceptible to interference from complex backgrounds compared to the RGB detector. As a result, under normal and low illumination conditions, the RGB detector is more likely to misidentify background parts as target areas, such as a wheel in the background of the images in the first row of Figure 11b,c, and a person and a portion of the back row of citrus tree canopies in the background of the images in the second row of Figure 11b,c. Under strong illumination conditions, although depth data may be affected

by interference, the segmentation results of the RGB-D detector are still superior to those of the RGB detector (e.g., the third row of Figure 11b,c). This indicates that the RGB-D detector can be used under different lighting conditions, and it consistently demonstrates better detection performance compared to the RGB detector. Therefore, considering detection accuracy and time performance, we will further analyze and study the RGB-D detector based on DeepLabv3+.

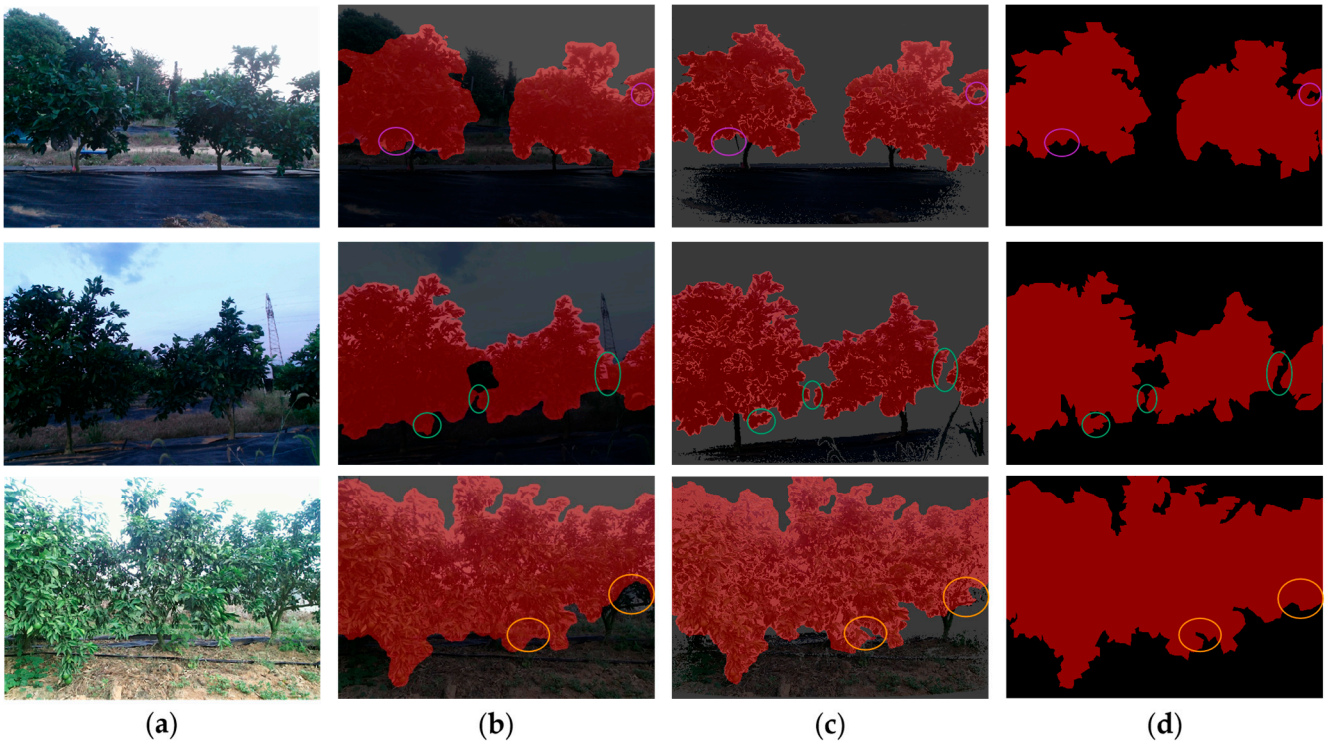

Figure 11. Experimental results of different detectors: (**a**) Original image; (**b**) RGB detector; (**c**) RGB-D detector; (**d**) Manually labeled images. The circles of each color in the figure indicate the parts where the segmentation results of the RGB detector and the RGB-D detector are significantly different in the corresponding example result images, and the corresponding parts in the manual annotation images are also circled for comparison.

3.3. Ablation Experiment

In this section, we conducted ablation experiments on the DeepLabv3+ model to demonstrate the effectiveness of our proposed improvements. We trained the model using an RGB-D dataset and compared different enhancement methods for the backbone network and ASPP module in the DeepLabv3+ model. The DeepLabv3+ model with ResNet50 as the backbone network is denoted as ResNet50. The DeepLabv3+ model with MobileNetV3-Small as the backbone network is denoted as MobileNetV3. The DeepLabv3+ model with the DSC network replacing the convolutional network in the ASPP module is denoted as MobileNetV3-DSC. MobileNetV3-DSC-SAM and MobileNetV3-DSC-CA represent MobileNetV3-DSC with the SAM module and CA module added, respectively. MobileNetV3-DSC-SAM-CA represents MobileNetV3-DSC with both the SAM module and CA module added. The performance evaluation results of the ablation models on the RGB-D test set A are presented in Table 4.

Table 4. A comparison of the results of ablation experiments.

Methods	DSC	SAM	CA	*Acc* (%)	*Re* (%)	*Pr* (%)	*mIoU* (%)	F_1 (%)	Speed (FPS)	Parameters (M)
ResNet50				97.64	97.59	96.58	95.27	97.08	5.31	40.34
MobileNetV3				95.61	97.04	94.75	93.19	95.88	26.06	6.24
MobileNetV3-DSC	√			96.40	96.67	95.73	93.74	96.20	34.98	3.05
MobileNetV3-DSC-SAM	√	√		97.48	97.27	96.49	94.99	96.88	33.35	3.11
MobileNetV3-DSC-CA	√		√	97.42	96.56	97.06	94.87	96.81	31.75	3.05
MobileNetV3-DSC-SAM-CA	√	√	√	97.79	97.98	97.06	95.62	97.52	32.69	3.11

Note: "√" indicates that the marked module is used in the experiment, and no "√" indicates that this module is not added.

It can be seen from Table 4 that when the DeepLabv3+ model, using ResNet50 as its backbone network, is employed for citrus canopy segmentation, it achieves pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed of 97.64%, 97.59%, 96.58%, 95.27%, 97.08%, and 5.31 FPS, respectively, on test set A. The model demonstrates high segmentation performance. However, due to its large parameter count of 40.34 M, it is not suitable for embedded platform systems. By replacing the backbone network of DeepLabv3+ with the lightweight MobileNetV3-Small model, the parameter count can be reduced to 6.24 M, making it suitable for embedded platform usage. Nevertheless, the decrease in computational parameters leads to a reduction in detection accuracy, with pixel accuracy and *mIoU* of the model reaching only 95.61% and 93.19%, respectively. The proposed MobileNetV3-DSC-SAM-CA model in this study achieves pixel accuracy, *mIoU*, and detection speed of 97.79%, 95.62%, and 32.69 FPS, respectively, on the test dataset. Compared to MobileNetV3, the MobileNetV3-DSC-SAM-CA model achieves an improvement of 2.18% and 2.43% in pixel accuracy and *mIoU*, respectively, without compromising the model's detection speed. Compared to ResNet50, the MobileNetV3-DSC-SAM-CA model achieves a six times increase in detection speed while reducing the model's parameter count by 92.3% to only 3.11 M. Experimental results demonstrate that the DSC network effectively reduces the model's computational load. Additionally, incorporating appropriate attention modules in the ASPP module enhances the model's ability to extract semantic information. Therefore, the performance of the MobileNetV3-DSC model is superior to that of the MobileNetV3 model. Furthermore, the MobileNetV3-DSC-SAM-CA model, with the inclusion of the attention module, effectively enhances the model's capability to extract semantic information while maintaining a lightweight parameter count, slightly outperforming the segmentation performance of the ResNet50 model.

3.4. Comparative Results and Analysis of Different Models

In this paper, we employed commonly used machine learning methods such as Image features + SVM [38], as well as state-of-the-art deep learning-based image segmentation algorithms, including U-Net, PSPNet, DeepLabv3+, and the proposed network in this paper. These models were trained on an RGB-D dataset, with all deep learning-based models using the same training parameters and employing transfer learning for model training. The trained models were evaluated for their performance on test set A and test set B. Additionally, we generated segmented result images by applying threshold segmentation to the foreground images obtained from the two test sets. These segmented result images serve as reference comparison images for the image segmentation algorithms. The segmentation results for all models are presented in Table 5 and Figure 12.

Table 5. Comparison of performance of different models.

	Segmentation Methods	Backbone	*Acc* (%)	*Re* (%)	*Pr* (%)	*mIoU* (%)	F_1 (%)	Speed (FPS)	Parameters (M)
Test A	Threshold segmentation	—	78.69	98.98	70.65	69.87	82.45	—	—
	Image features + SVM	—	75.17	54.12	75.81	44.75	63.15	—	—
	PSPNet	ResNet50	94.25	94.94	90.94	88.78	92.90	8.81	46.71
	Unet	VGG16	97.74	96.21	97.95	95.36	97.07	2.63	24.89
		ResNet50	97.88	96.94	97.56	95.55	97.25	6.06	43.93
	DeepLabv3+	ResNet50	97.64	97.59	96.58	95.27	97.08	5.31	40.34
	Proposed network	MobileNetV3-Small	97.79	97.98	97.06	95.62	97.52	32.69	3.11
Test B	Threshold segmentation	—	80.08	99.40	71.10	71.16	82.90	—	—
	Image features + SVM	—	75.24	67.49	73.74	51.80	70.48	—	—
	PSPNet	ResNet50	93.12	94.64	87.75	86.11	91.06	8.94	46.71
	Unet	VGG16	94.19	98.67	91.95	90.82	95.19	2.58	24.89
		ResNet50	95.48	98.78	92.39	91.35	95.48	5.98	43.93
	DeepLabv3+	ResNet50	94.73	95.71	**93.57**	91.17	94.63	5.28	40.34
	Proposed network	MobileNetV3-Small	95.99	98.81	93.28	92.29	95.97	32.52	3.11

Note: Test A and Test B represent test set A and test set B, respectively.

Figure 12. Experimental results of different algorithms: (**a**) Original images; (**b**) Foreground segmentation images; (**c**) Threshold segmentation; (**d**) Image features + SVM; (**e**) PSPNet_ResNet50; (**f**) Unet_VGG16; (**g**) Unet_ResNet50; (**h**) DeepLabv3+; (**i**) Proposed network; (**j**) Manually labeled images.

Based on the results, threshold segmentation of the RGB-D depth values effectively eliminates complex background information while preserving the relatively complete canopy information of fruit trees. The pixel accuracy on test set A reaches 78.69%. However, the precision and *mIoU* values are only 70.65% and 69.87%, respectively, indicating that the foreground images still contain some non-canopy information. On test set B, the pixel accuracy and *mIoU* values reach 80.08% and 71.16%, respectively, showing that the effectiveness of the threshold segmentation of depth values is consistent across different test sets. Among the compared segmentation algorithms, Image features + SVM, as a common machine learning method, performs poorly on both test set A and test set B, with *mIoU* values of 44.75% and 51.80%, respectively. This is because the color, texture, and shape features of fruit tree canopies are complex, and the capability of this algorithm to extract image features is limited. Based on the image (Figure 12d), it can be observed that some darker canopy areas are labeled as non-canopy regions, while the ground is labeled as canopy regions, resulting in certain errors. Therefore, this algorithm does not meet the requirements.

The deep learning-based image segmentation algorithm performs well, and the PSPNet model exhibits good performance on segmentation tasks, but its use of global pooling in the final layer results in a loss of detailed information that impacts its ability to handle edge segmentation. On test set A, PSPNet achieves pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed of 94.25%, 94.94%, 90.94%, 88.78%, 92.90%, and 8.81 FPS, respectively. On test set B, the *mIoU* is 86.11%. As shown in Figure 12e, it can be seen that the segmentation results obtained using PSPNet lack fine recognition of the edge details of all citrus tree canopies in both test sets. The precision and mIoU of the segmentation are lower compared to other segmentation models. Furthermore, due to this model's complexity and large number of parameters, it requires more computational resources, making it unsuitable for real-time canopy detection tasks in citrus orchards. The Unet-VGG16 model employs upsampled layer-by-layer and convolution operations for decoding and introduces skip connections between the encoder and decoder, connecting the feature maps extracted from the encoding part with the feature maps obtained from the upsampling layer. Then, the feature maps are decoded through two convolution layers, preserving more detailed information. Therefore, Unet-VGG16 performs better in handling details than the PSPNet. The pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed of Unet-VGG16 on the test set A are 97.74%, 96.21%, 97.95%, 95.36%, 97.07%, and 2.63 FPS, respectively, with the model having 24.89 M parameters. On test set B, it achieves an *mIoU* of 90.82% and an F_1 of 95.19%. As shown in Figure 12f, it can be observed that the segmentation accuracy of this model is sufficiently fine on test set A. However, on test set B, some weeds are misclassified as canopy regions. Moreover, due to its complex structure with multiple convolution layers and parameters, its detection speed is relatively slow, meaning it cannot meet the real-time detection requirements for precision spraying in citrus orchards. Compared to the Unet model with VGG16, the Unet-ResNet50 model introduces residual connections to alleviate the vanishing gradient problem, maintaining segmentation performance while improving the segmentation rate. As shown in Figure 12g, in test set B, the model is able to label the majority of weed portions as non-canopy regions. The pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed of Unet-ResNet50 on the test set A are 97.88%, 96.94%, 97.56%, 95.55%, 97.25%, and 6.06 FPS, respectively. Furthermore, on test set B, it achieves an *mIoU* of 91.35% and a balanced F_1 score of 95.48%. The model has 43.93 M parameters. The DeepLabv3+ model uses an ASPP module on top of the encoder–decoder structure to capture multi-scale contextual information and improve segmentation accuracy. The Xception backbone network used by this model exhibits good performance on complex segmentation tasks. However, for simple segmentation tasks with limited training data, overfitting can easily occur. The ResNet50 network model uses residual connections to improve the depth and generalization ability of the network when dealing with deep networks, effectively avoiding overfitting. Therefore, in this study, the DeepLabv3+ model used for comparative experiments adopts the ResNet50, which is more suitable for tree

canopy segmentation tasks, as the backbone network. On test set A, the model achieved pixel accuracy, recall, precision, *mIoU*, F_1, and detection speed of 97.64%, 97.59%, 96.58%, 95.27%, 97.08%, and 5.31 FPS, respectively. However, on test set B, the model's *mIoU* and F_1 are 91.17% and 94.63%, respectively. The model has a parameter count of 40.34 M. As shown in Figure 12h, it can be observed that the model misclassifies a small portion of tree canopies as non-canopy regions on test set B, resulting in a recall rate of only 95.71%.

The proposed improved DeepLabv3+ model optimizes the ASPP module based on the image features of tree crowns in segmentation tasks. The improved ASPP module maintains the same image feature extraction capability while consuming fewer computational resources. Additionally, MobileNetV3-Small is used as the backbone network, which, compared to other network models, maintains high accuracy while having a smaller parameter count and faster detection speed, making it more suitable for segmentation tasks on mobile platforms. Under the proposed approach, this model demonstrates excellent segmentation accuracy, surpassing other network models in terms of recall, *mIoU*, and F_1 scores. As shown in Figure 12i, it can be observed that the model accurately segments canopy regions in both test sets. On test set A, the model achieves pixel accuracy, recall, precision, *mIoU*, and F_1 of 97.79%, 97.98%, 97.06%, 95.62%, and 97.52%, respectively. Compared to the threshold segmentation baseline, significant improvements are observed in recognition performance, particularly in precision and *mIoU*, which increase by 26.41% and 25.75%, respectively. On test set B, the model achieves precision, *mIoU*, and F_1 of 93.28%, 92.29%, and 95.97%, respectively, surpassing the threshold segmentation baseline by 22.18%, 21.13%, and 13.06%, respectively. Hence, it demonstrates the feasibility of using threshold segmentation as an image data preprocessing method and employing suitable semantic segmentation algorithms to segment fruit tree canopy regions. Furthermore, this network achieves significant improvements in detection speed on embedded devices, reaching 32.69 FPS, which is more than four times faster than other models. With a parameter count of only 3.11 M, this model has a substantially smaller parameter size compared to other models.

4. Discussion

In this paper, we combined color and depth information and proposed a citrus canopy segmentation model based on an improved DeepLabv3+ model, which was trained and tested on two datasets collected from orchards in different regions. The Orchard A dataset contained 540 groups of images, which were divided into the training set, validation set, and test set A, and data augmentation was performed on the training set to improve the generalization and robustness of the model. The Orchard B dataset contained 100 groups of images, which were used as test set B to verify the generalization performance of the model.

RGB-D images, which contain both color and depth information, have been proven to effectively reduce the interference of image background in semantic segmentation. In this study, a method of fusing depth information with RGB images was used to pre-exclude background information from citrus tree images, improving the segmentation accuracy of the model. Compared to semantic segmentation network models trained on RGB images alone, the proposed method is more accurate in segmenting target tree crowns and demonstrates better robustness under varying lighting conditions.

In semantic segmentation tasks, the ASPP serves as a crucial component of the encoder, primarily responsible for extracting multi-scale feature information through multiple parallel dilated convolutions. These features play a significant role in determining the location and boundaries of segmented targets, endowing the model with a stronger perception of targets of various sizes and shapes and improving segmentation accuracy. In the DeepLabv3+ network, the ASPP module mainly extracts high-level features from the images. Therefore, when applying the model on embedded platforms, it is possible to optimize the structure of the ASPP module by replacing convolutional networks with the DSC network, effectively reducing the model's computational burden and alleviating the workload of semantic segmentation tasks [39]. Additionally, an attention mechanism

module is introduced, enabling the model to simultaneously focus on channel, spatial, and positional information, thus enhancing the accuracy of semantic segmentation and improving the stability of the model's performance [35,40]. The primary function of the backbone network in the encoder is to extract image features layer by layer through multiple convolutional and pooling layers. Generally, the deeper the backbone network, the more abstract the extracted image features, which play a crucial role in segmenting complex image information. Consequently, when image information is not complex and higher detection speed is required, using a lightweight convolutional neural network as the backbone structure is feasible. In this study, DeepLabv3+ was chosen as the base model, and the model's computational efficiency, generalization capability, and feature extraction ability were improved by adding the attention mechanism module in the ASPP module and introducing the DSC method. Additionally, MobilNetV3-Small was employed as the backbone network, reducing the model's parameter and computational load and further enhancing the model's detection speed. The experimental results show that the improved DeepLabv3+ model achieves pixel accuracy and *mIoU* of 97.79% and 95.62%, respectively, on test set A, and pixel accuracy and *mIoU* of 95.66% and 92.29%, respectively, on test set B, and the detection speed reaches 32.69 FPS on Jetson Xavier NX. These results indicate that the model has strong generalization ability and high detection speed.

The proposed approach in this study focuses on optimizing and simplifying the model to achieve high-speed operation on embedded devices (such as Jetson Xavier NX, Jetson Orin Nano, and Raspberry Pi) without relying on remote servers. This approach ensures algorithm accuracy while meeting the real-time detection requirements in orchard environments. Therefore, the algorithm can be deployed in orchard spraying robots to perform precise spraying according to the canopy information of the target fruit trees. This can not only improve the savings rate of pesticides but also reduce environmental pollution caused by the liquid in non-target areas.

The lightweight model presented in this paper has the primary advantage of rapid operation. However, it also has some limitations. Despite achieving high-level segmentation performance, there are still a few instances in test set B where weeds are incorrectly labeled as canopy areas. This is because most of the images in the training set were covered with weed cloth, resulting in limited feature information about weeds. In contrast, test set B contains orchards without weed cloth, leading to more weed features that serve as interference. Furthermore, the model is trained only on canopy depth and color information and can only output fruit tree canopies. It is unable to differentiate fruits within the canopy or detect the health status and yield of the trees. In future research, we will consider training the model using data from different orchards to enhance its generalization capability. We will also explore secondary classification of canopy information and fusion of spectral data to better detect the health status of fruit tree canopies and segment canopy areas affected by different pests and diseases.

5. Conclusions

This study utilized a data acquisition system primarily based on the KinectV2 sensor to capture images of citrus trees in orchards. An improved DeepLabv3+ model is proposed for citrus tree canopy segmentation. The model effectively segments the canopy regions in the images, achieving *mIoU* of 95.62% and 92.29% on test set A and test set B, respectively. Furthermore, the detection speed on embedded devices reaches 32.69 FPS. Compared to traditional machine learning methods (image features + SVM), the deep semantic segmentation approach yields higher recognition accuracy. Compared to other advanced semantic segmentation models, the improved DeepLabv3+ model in this study significantly improves detection speed while maintaining segmentation performance and model generalization. The results indicate that the prediction accuracy and speed of this model reached a high level, making it suitable for precision spraying in orchards. This lightweight model can operate independently on embedded systems. Additionally, the canopy segmentation results of this model preserve the depth data from RGB-D images,

enabling real-time calculation of canopy area, volume, and other information, which can serve as references for the precise operation of orchard spraying robots. However, the current model still has limitations in that it cannot classify crown regions according to the health status of the fruit trees. In the future, we will focus on developing more practical and valuable models that can recognize and segment crown regions of different health statuses. These models will provide more accurate canopy information for precision operations, including spraying, fertilization, and yield measurement performed by orchard robots.

Author Contributions: Conceptualization, X.X., M.B., Z.L., S.L. and S.S.; methodology, Q.L., M.B., Z.L., S.L. and S.S.; validation, Q.L., M.B., Z.L. and S.S.; formal analysis, Q.L.; investigation, X.X.; resources, M.B., Z.L., S.L. and S.S.; data curation, Q.L. and M.B.; writing—original draft preparation, Q.L.; writing—review and editing, X.X.; visualization, Q.L.; supervision, X.X.; project administration, S.L.; funding acquisition, X.X. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (31971797, 32271997); Key-Area Research and Development Program of Guangdong Province (2023B0202090001); China Agriculture Research System of MOF and MARA (CARS-26); General Program of Guangdong Natural Science Foundation (2021A1515010923); Special Projects for Key Fields of Colleges and Universities in Guangdong Province (2020ZDZX3061); and the Guangdong Provincial Special Fund For Modern Agriculture Industry Technology Innovation Teams (2023KJ108).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy of the organization.

Acknowledgments: The authors would like to thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. FAO. *FAO Statistical Yearbook 2013: World Food and Agriculture*; FAO: Rome, Italy, 2013; p. 169. Available online: https://www.fao.org/3/i3107e/i3107e.PDF (accessed on 10 June 2023).
2. Lan, Y.; Yan, Y.; Wang, B.; Song, C.; Wang, G. Current status and future development of the key technologies for intelligent pesticide spraying robots. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 30–40. (In Chinese) [CrossRef]
3. Patil, S.; Patil, Y.; Patil, S. Review on Automatic Variable-Rate Spraying Systems Based on Orchard Canopy Characterization. *Inform. Autom.* **2023**, *22*, 57–86. [CrossRef]
4. Ampatzidis, Y.; Partel, V. UAV-Based High Throughput Phenotyping in Citrus Utilizing Multispectral Imaging and Artificial Intelligence. *Remote Sens.* **2019**, *11*, 410. [CrossRef]
5. Maghsoudi, H.; Minaei, S.; Ghobadian, B.; Masoudi, H. Ultrasonic sensing of pistachio canopy for low-volume precision spraying. *Comput. Electron. Agric.* **2015**, *112*, 149–160. [CrossRef]
6. Wang, J.; Chen, X.; Cao, L.; An, F.; Chen, B.; Xue, L.; Yun, T. Individual rubber tree segmentation based on ground-based LiDAR data and faster R-CNN of deep learning. *Forests* **2019**, *10*, 793. [CrossRef]
7. Wu, B.; Yu, B.; Wu, Q.; Huang, Y.; Chen, Z.; Wu, J. Individual tree crown delineation using localized contour tree method and airborne LiDAR data in coniferous forests. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 82–94. [CrossRef]
8. Mahmud, M.S.; Zahid, A.; He, L.; Martin, P. Opportunities and Possibilities of Developing an Advanced Precision Spraying System for Tree Fruits. *Sensors* **2021**, *21*, 3262. [CrossRef]
9. Abbas, I.; Liu, J.; Faheem, M.; Noor, R.S.; Shaikh, S.A.; Solangi, K.A.; Raza, S.M. Different sensor based intelligent spraying systems in Agriculture. *Sens. Actuators A Phys.* **2020**, *316*, 112265. [CrossRef]
10. Xiao, K.; Ma, Y.; Gao, G. An intelligent precision orchard pesticide spray technique based on the depth-of-field extraction algorithm. *Comput. Electron. Agric.* **2017**, *133*, 30–36. [CrossRef]
11. Gao, G.; Xiao, K.; Ma, Y. A leaf-wall-to-spray-device distance and leaf-wall-density-based automatic route-planning spray algorithm for vineyards. *Crop Prot.* **2018**, *111*, 33–41. [CrossRef]
12. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
13. Miao, J.; Wei, Y.; Wu, Y.; Liang, C.; Li, G.; Yang, Y. VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4133–4143. [CrossRef]

14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
15. Alves, A.N.; Souza, W.S.R.; Borges, D.L. Cotton pests classification in field-based images using deep residual networks. *Comput. Electron. Agric.* **2020**, *174*, 105488. [CrossRef]
16. Azizi, A.; Abbaspour-Gilandeh, Y.; Vannier, E.; Dusséaux, R.; Mseri-Gundoshmian, T.; Moghaddam, H.A. Semantic segmentation: A modern approach for identifying soil clods in precision farming. *Biosyst. Eng.* **2020**, *196*, 172–182. [CrossRef]
17. Barros, T.; Conde, P.; Gonçalves, G.; Premebida, C.; Monteiro, M.; Ferreira, C.S.S.; Nunes, U.J. Multispectral vineyard segmentation: A deep learning comparison study. *Comput. Electron. Agric.* **2022**, *195*, 106782. [CrossRef]
18. Majeed, Y.; Zhang, J.; Zhang, X.; Fu, L.; Karkee, M.; Zhang, Q.; Whiting, M.D. Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* **2020**, *170*, 105277. [CrossRef]
19. Zou, K.; Liao, Q.; Zhang, F.; Che, X.; Zhang, C. A segmentation network for smart weed management in wheat fields. *Comput. Electron. Agric.* **2022**, *202*, 107303. [CrossRef]
20. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* **2020**, *171*, 105302. [CrossRef]
21. Majeed, Y.; Zhang, J.; Zhang, X.; Fu, L.; Karkee, M.; Zhang, Q.; Whiting, M.D. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. *IFAC-PapersOnLine* **2018**, *51*, 75–80. [CrossRef]
22. Sun, Q.; Zhang, R.; Chen, L.; Zhang, L.; Zhang, H.; Zhao, C. Semantic segmentation and path planning for orchards based on UAV images. *Comput. Electron. Agric.* **2022**, *200*, 107222. [CrossRef]
23. Sun, K.; Wang, X.; Liu, S.; Liu, C. Apple, peach, and pear flower detection using semantic segmentation network and shape constraint level set. *Comput. Electron. Agric.* **2021**, *185*, 106150. [CrossRef]
24. Anagnostis, A.; Tagarakis, A.C.; Kateris, D.; Moysiadis, V.; Sørensen, C.G.; Pearson, S.; Bochtis, D. Orchard Mapping with Deep Learning Semantic Segmentation. *Sensors* **2021**, *21*, 3813. [CrossRef]
25. Cao, L.; Zheng, X.; Fang, L. The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm. *Electronics* **2023**, *12*, 929. [CrossRef]
26. Shi, L.; Wang, G.; Mo, L.; Yi, X.; Wu, X.; Wu, P. Automatic Segmentation of Standing Trees from Forest Images Based on Deep Learning. *Sensors* **2022**, *22*, 6663. [CrossRef] [PubMed]
27. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818. [CrossRef]
28. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [CrossRef]
29. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19. [CrossRef]
30. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [CrossRef]
31. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122. [CrossRef]
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
33. Torralba, A.; Russell, B.C.; Yuen, J. Labelme: Online image annotation and applications. *Proc. IEEE* **2010**, *98*, 1467–1484. [CrossRef]
34. Chen, L.-C.; Papandreou, G.; Schroff, F.; Hartwig, A. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef]
35. Liu, R.; Tao, F.; Liu, X.; Na, J.; Leng, H.; Wu, J.; Zhou, T. RAANet: A Residual ASPP with Attention Framework for Semantic Segmentation of High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 3109. [CrossRef]
36. Xie, J.; Jing, T.; Chen, B.; Peng, J.; Zhang, X.; He, P.; Yin, H.; Sun, D.; Wang, W.; Xiao, A.; et al. Method for Segmentation of Litchi Branches Based on the Improved DeepLabv3+. *Agronomy* **2022**, *12*, 2812. [CrossRef]
37. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
38. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
39. Yang, G.; Wang, J.; Nie, Z.; Yang, H.; Yu, S. A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy* **2023**, *13*, 1824. [CrossRef]
40. Wan, T.; Rao, Y.; Jin, X.; Wang, F.; Zhang, T.; Shu, Y.; Li, S. Improved U-Net for Growth Stage Recognition of In-Field Maize. *Agronomy* **2023**, *13*, 1523. [CrossRef]