agronomy

MDPI

Article

Method for Segmentation of Litchi Branches Based on the Improved DeepLabv3+

Jiaxing Xie [1,2,3], Tingwei Jing [1], Binhan Chen [1], Jiajun Peng [1], Xiaowei Zhang [1], Peihua He [1], Huili Yin [1], Daozong Sun [1,3], Weixing Wang [1,3], Ao Xiao [1], Shilei Lyu [1,4] and Jun Li [1,2,5,*]

[1] College of Electronic Engineering (College of Artificial Intelligence), South China Agricultural University, Guangzhou 510642, China
[2] Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510640, China
[3] Guangdong Engineering Research Center for Monitoring Agricultural Information, Guangzhou 510642, China
[4] Pazhou Lab, Guangzhou 510330, China
[5] College of Engineering, South China Agricultural University, Guangzhou 510642, China
* Correspondence: autojunli@scau.edu.cn

Abstract: It is necessary to develop automatic picking technology to improve the efficiency of litchi picking, and the accurate segmentation of litchi branches is the key that allows robots to complete the picking task. To solve the problem of inaccurate segmentation of litchi branches under natural conditions, this paper proposes a segmentation method for litchi branches based on the improved DeepLabv3+, which replaced the backbone network of DeepLabv3+ and used the Dilated Residual Networks as the backbone network to enhance the model's feature extraction capability. During the training process, a combination of Cross-Entropy loss and the dice coefficient loss was used as the loss function to cause the model to pay more attention to the litchi branch area, which could alleviate the negative impact of the imbalance between the litchi branches and the background. In addition, the Coordinate Attention module is added to the atrous spatial pyramid pooling, and the channel and location information of the multi-scale semantic features acquired by the network are simultaneously considered. The experimental results show that the model's mean intersection over union and mean pixel accuracy are 90.28% and 94.95%, respectively, and the frames per second (FPS) is 19.83. Compared with the classical DeepLabv3+ network, the model's mean intersection over union and mean pixel accuracy are improved by 13.57% and 15.78%, respectively. This method can accurately segment litchi branches, which provides powerful technical support to help litchi-picking robots find branches.

Keywords: improved DeepLabv3+; litchi branches; semantic segmentation; litchi; attention mechanism

check for updates

Citation: 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. https://doi.org/10.3390/agronomy12112812

Academic Editors: Jun Ni, Lei Feng and Lvhua Han

Received: 24 September 2022
Accepted: 9 November 2022
Published: 11 November 2022

1. Introduction

China is the country with the widest planting area, the largest yield, and the largest production area of litchi in the world [1]. At present, manual picking is the most important litchi picking method, but it has low efficiency and poor timeliness. The litchi picking period is short, so if the fruit cannot be picked in time, it will be damaged to a certain extent [2]. Therefore, it is necessary to develop automatic picking technology. The use of automatic picking technology can improve the picking efficiency and minimize the damage of litchi fruit caused by not picking in time. It can also reduce labor costs and improve the returns of fruit farmers.

In recent years, many scholars have conducted substantial research on fruit and vegetable recognition and segmentation performed by picking robots [3,4]. The accurate segmentation of fruit branches is the key that allows picking robots to complete their task. The picking point of litchi is located on its branches, so to realize automatic picking, the accurate identification of litchi branches is one of the first problems that need to be solved.

At present, there are two method types for fruit branch segmentation: traditional methods and deep learning methods.

Of the traditional image processing methods, the most common are the Otsu algorithm, fuzzy clustering method, and K-means. Xiong et al. [5] used the fuzzy clustering method (FCM) and Otsu algorithm to segment litchi fruits and stems from images taken at night. The litchi regions were segmented by combining red/green chromatic mappings and using Otsu thresholding in the RGB color space [6]. By incorporating both local spatial and local hue-level intensity relationships, the hue component in the HSV color space was reconstructed to filter the noise. Xiong et al. [7] used an improved fuzzy C-means clustering method for image segmentation to obtain litchi fruits and branches, and Luo et al. [8] extracted the HSI color space component H of grapes and segmented the images using the improved artificial bee colony optimization fuzzy clustering method.

In practical application scenarios, traditional image processing methods are easily affected by the complex environment of orchards. Therefore, the use of deep learning methods can improve the accuracy and robustness of segmentation algorithms.

By using the deep learning method, models gain stronger robustness and better generalization ability by learning the features of a large number of samples to extract deeper features. Therefore, research on fruit and branch recognition and segmentation using deep learning methods has become mainstream. Image segmentation is further divided into semantic segmentation and instance segmentation. Semantic segmentation networks mainly use models such as the FCN [9], U-Net [10], SegNet [11], PSPNet [12], DeepLabv3 [13], and DeepLabv3+ [14], while instance segmentation often uses the Mask R-CNN [15] and YOLACT [16] models.

Li et al. [17] used the improved U-Net model to segment images of green apples and added edge structures to obtain the target image's edge information. Then, an atrous spatial pyramid pooling structure was applied to merge the edge features with the high-level features. The recognition rates for single fruits, covered fruits, and overlapping fruits were 97.15%, 93.58% and 94.46%, respectively. Yu et al. [18] used Mask-RCNN to segment strawberries and introduced the feature pyramid network to improve the backbone network's feature extraction ability. The average detection precision rate was 95.78%, the recall rate was 95.41%, and the mean intersection over union of the instance segmentation was 89.85%. Cai et al. [19] introduced the attention mechanism into the DeepLabv3+ model to segment strawberries and added two serial attention modules, the efficient channel attention module and the simple, parameter-free attention module, to the backbone network, Xception, to improve its feature extraction ability. They also integrated the convolutional block attention module into the atrous spatial pyramid pooling module to reduce the interference of environmental factors. The model's average pixel accuracy and mean intersection over union were 90.9% and 83.05%, respectively. Ning et al. [20] used the Mask-RCNN model to roughly segment grape fruit stems and used the regional growth algorithm to finely segment fruit stems. The mean intersection over union was 85.73%, 83.51%, and 81.75% in sunny and sunlight, sunny and overshadow shaded, and cloudy light, respectively, and the average detection time was 4.9 s. Xue et al. [21] proposed a multi-scale feature extraction module, which was added to the FCN-8S model for the segmentation of the Lingwu long jujube. The model's mean intersection over union was 96.41%. Yang et al. [22] used the SegNet model to segment rice panicles and extract their phenotypic characteristics. The mean intersection over union and mean accuracy of the model were 83.8% and 97.0%, respectively.

In recent years, the research on the recognition and segmentation of litchi fruits and branches using deep learning methods has gradually become a hot topic. Li et al. [23] used an RGB-D camera to collect images of litchi, and the DeepLabv3 model was employed to segment images of litchi into three categories: litchi fruit, branches, and background. By combining morphological processing, density-based spatial clustering of application with noise clustering, and principal component analysis, the spatial location information of the fruits and branches belonging to the same litchi cluster was obtained. The mean intersection over union of the model was 79.46%. Peng et al. [24] used the DeepLabv3+ model with Xception_65 as the backbone network for the semantic segmentation of litchi branches, and the mean intersection over union of the model was 76.5%. Peng et al. [25] used the DeepLabv3+ model for the semantic segmentation of litchi branches, and used focal loss as loss function to solve the problem of data imbalance. The model's mean intersection over union was 79.7%. Zhong et al. [26] used the YOLACT model to segment litchi clusters and main fruit-bearing branches and determined their location according to the pixel differences between them. The accuracy of locating picking points was 89.7%, and the F1-score was 83.8%. Liang et al. [27] used YOLOv3 to detect litchi fruits in a natural environment at night, determined the regions of interest on the litchi branches, and used the U-Net model to segment litchi fruit stems, thus realizing the detection of litchi fruit stems at night. The mean intersection over union of the fruit stem segmentation model was 79.00%, 84.33%, and 78.60% under the high-brightness, normal brightness, and low-brightness settings, respectively. Qi et al. [28] used YOLOv5 to detect the main stem and litchi clusters, extracted the detected images of litchi stems, and used the semantic segmentation model, PSPNet, to segment the images and obtain exact images of the litchi stems. The mean intersection over union of the model was 74.08%.

The segmentation accuracy for the litchi branches in previous studies initially meets the needs of subsequent picking robots, but there is still room for improvement in terms of the segmentation accuracy. To further improve the model's segmentation effect on litchi branches, this paper proposes an improved DeepLabv3+ semantic segmentation model that can improve the model's feature extraction ability by replacing the backbone network of the classical DeepLabv3+ model and adding the attention mechanism to the atrous spatial pyramid pooling module. During the training process, the loss function is modified to cause the model to pay more attention to mining the foreground region, which can improve the model's segmentation effect.

2. Materials and Methods

2.1. Construction of the Litchi Image Dataset

The experimental data were collected from the litchi standard orchard (orderly production management according to standard specifications) in the Germplasm Resource Nursery of Southern Characteristic Fruit Trees of the State Key Laboratory of South China Agricultural University and the litchi orchard of the Shenma Ecological Agriculture Development Company Limited, Maoming City, Guangdong Province, China. Collected between June and July 2022, the dataset contains three weather conditions: sunny, cloudy, and rainy. It also contains two lighting conditions: natural light and backlight. The varieties of litchi include Guiwei and Nuomici. From the collected data, 333 pieces were randomly selected as experimental data. To increase the richness of the dataset, litchi images in various conditions were found on the web via crawling, and 155 images were selected as experimental data, resulting in a total of 488 litchi images. The dataset's images contain litchi fruits, litchi branches, and complex orchard backgrounds.

LabelMe was used to construct the semantic segmentation dataset. The litchi branches in the images were annotated, and the rest of the images were used as background. The annotated data were saved in the VOC format.

To improve the model's generalization ability and prevent overfitting, it was necessary to enhance the sample data and expand the number of samples. In this paper, we used offline data augmentation methods, including rotating 90 degrees, rotating 270 degrees, flipping left and right, and adjusting the brightness, chroma, and contrast. The total number of images after the expansion was 3416.

2.2. Overview of Litchi Branch Segmentation Models

2.2.1. DeepLabv3+ Model

DeepLab is a series of semantic segmentation models developed by the Google team. DeepLabv3 [13] consists of a serial structure in which dilated convolutions with dilated rates of 2, 4, and 8 were added to the backbone network, followed by the addition of a modified atrous spatial pyramid pooling (ASPP) module to obtain multiscale semantic features. Finally, upsampling is used to recover the original graph size and obtain the semantic segmentation results. The dilated convolutions added to the DeepLabv3 network cause gridding artifacts, which can affect the accuracy for a pixel-level prediction task like semantic segmentation. Therefore, although the segmentation accuracy is improved compared to DeepLabv1 [29] and DeepLabv2 [30], the segmentation effect is still limited.

Inspired by the encoder-decoder structure, the DeepLabv3+ [14] network adds a simple decoder structure to DeepLabv3; this structure is used to recover the target objects' boundary information. The DeepLabv3+ network is divided into an encoder and a decoder. The encoder consists of a backbone feature extraction network and an atrous spatial pyramid pooling module. The feature extraction network in the classical DeepLabv3+ network is Xception. The atrous spatial pyramid pooling module includes five branches, which include a 1 × 1 convolution and three 3 × 3 dilated convolutions with dilated rates of 6, 12, and 18, which can obtain rich contextual information. In the decoder, the multi-scale semantic feature maps obtained from the atrous spatial pyramid pooling module are upsampled by a factor of 4, and the features are then fused with the low-level semantic information extracted from the backbone network. After a 3 × 3 convolution, upsampling is performed by a factor of 4 to obtain the semantic segmentation result.

2.2.2. Backbone Network

In this paper, the Dilated Residual Networks (DRN) [31] are used as the backbone feature extraction network. The DRN is an improved model that combines ResNet [32] with dilated convolution [33]. In the model, the classical ResNet downsamples the input image by a factor of 32. If the height and width of the input image is 512 × 512, the final convolution layer output is only 16 × 16. In litchi images, litchi branches, which are small target objects, occupy a small part of the entire image, and it is therefore easy for a model to ignore them when it suffers from excessive dimensionality reduction and spatial sensitivity loss, which are not conducive to image segmentation. The DRN subsamples the input image by a factor of 8 and combines the dilated convolution to increase the receptive field while maintaining the size of the feature map. In addition, the DRN adds dilated convolutions with dilated rates of 2, 4, and 2 in stages 5–7, respectively. Using dilated convolution results in gridding artifacts, but fortunately, DRN-C-26 is a modified model that eliminates gridding artifacts.

DRN-C-26 is an improvement on ResNet18. The modification details are shown in Figure 1. During the network's first two stages, the max pooling operation in ResNet18 leads to high-amplitude high-frequency activations, which further exacerbates gridding artifacts. Therefore, in DRN-C-26, the max pooling layer is changed to the Conv-BN-ReLU combination with residual blocks. During the 7th and 8th stages, since residual connections propagate the gridding artifacts from the 6th stage, DRN-C-26 adds the Conv-BN-ReLU combination without residual blocks to better eliminate these artifacts.

To ensure the model's real-time performance, DRN-C-26 needs to be simplified. DRN-D-22 is a simplified version of DRN-C-26 and is shown in Figure 2. Compared with DRN-C-26, in DRN-D-22, the 1st, 2nd, 7th, and 8th stages remove a Conv-BN-ReLU combination and residual connection, which simplifies the model structure and reduces the computational overhead. In this paper, DRN-D-22 is used as the feature extraction network for DeepLabv3+.

2.2.3. Coordinate Attention

The attention mechanism is widely used in the fields of image classification, object detection, and image segmentation, and it plays an important role in improving the accuracy of network models [34,35]. The channel attention mechanism of squeeze-and-excitation networks [36] only considers the relationship between the feature map's channel information, ignoring the location feature information. Hou et al. [37] proposed the coordinate attention

(CA) mechanism, which not only considers the relationship between channels, but also the feature space's location information to achieve better feature expression without increasing the computational overhead numerously.

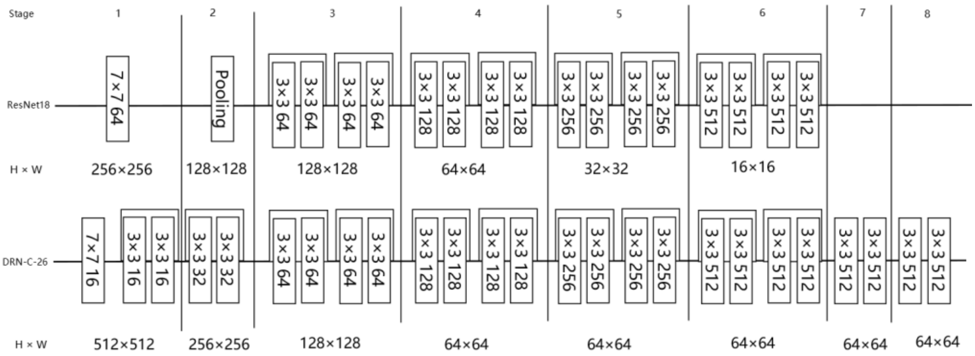

Figure 1. A comparison of ResNet18 and DRN-C-26. Each rectangle in the figure represents a Conv-BN-ReLU combination. The number in the rectangle indicates the size of the convolution kernel and the number of output channels. H × W indicates the height and width of the feature map.

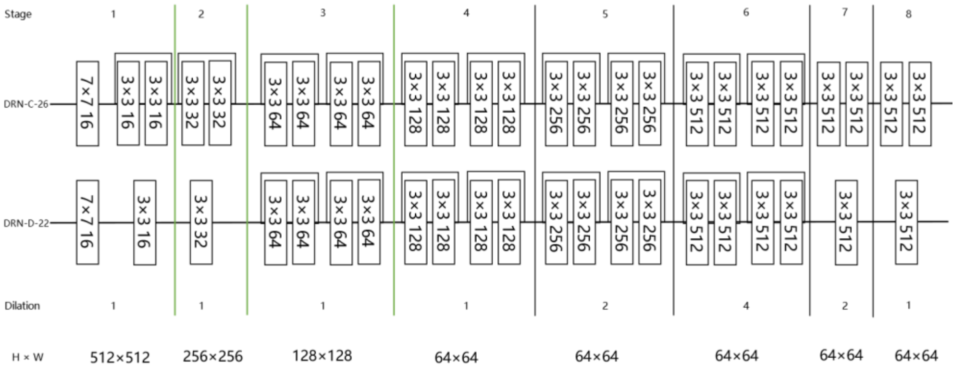

Figure 2. A comparison of DRN-D-22 and DRN-C-26. The DRN is divided into eight stages, and each stage outputs identically-sized feature maps and uses the same expansion coefficient. Each rectangle in the figure represents a Conv-BN-ReLU combination. The number in the rectangle indicates the size of the convolution kernel and the number of output channels. H × W is the height and width of the feature map, and the green lines represent downsampling by a stride of two.

The operation process of the CA module is shown in Figure 3. First, the global average pooling operation is performed on the input feature map in both the height and width directions, followed by feature aggregation and a 1 × 1 convolution operation. Immediately after the batch normalization process and the nonlinear activation function, the feature expressions in both directions are obtained using 1 × 1 convolution and the sigmoid activation function in the height and width directions, respectively. Finally, the feature maps in both directions are multiplied to obtain the final output of the CA module.

The CA module is added behind the ASPP module, and the channel and location information of the multi-scale semantic features acquired by the network are also considered to help the model better segment the litchi branches. The structure of the improved DeepLabv3+ network is shown in Figure 4.

2.2.4. Loss Function Design

Jadon summarized the commonly used loss functions. In the field of semantic segmentation, Cross-Entropy loss is the most commonly used loss function [38]. Cross-Entropy loss accounts for all of the pixels in the image evenly as a whole. However, in the dataset used in this paper, the litchi branches occupy few regions in the entire image, and the area containing the branches and background are severely unbalanced. Milletari et al. [39]

proposed a loss function known as Dice Loss, which is based on the Dice coefficient. This loss function can alleviate the negative impact caused by imbalance between the target area and the background area in the dataset. During the training process, Dice Loss paid more attention to the target area and was more inclined to excavate the target area. However, the Dice Loss function easily falls into local optimal solutions, and the training is unstable. Therefore, it is not conducive to model convergence.

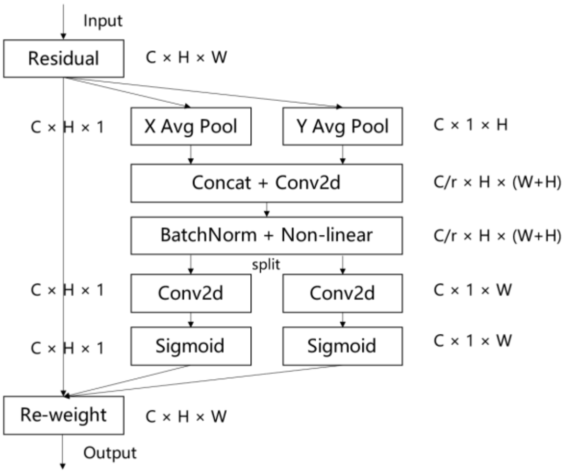

Figure 3. The coordinate attention mechanism.

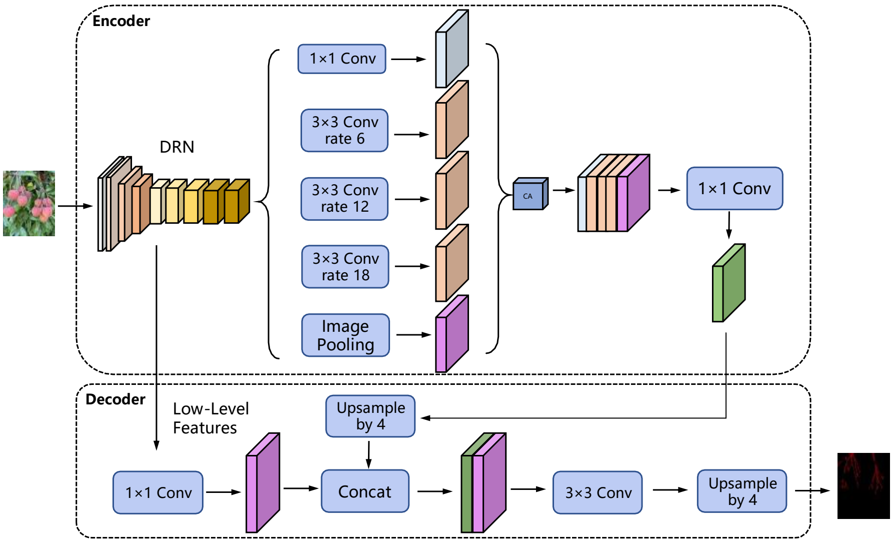

Figure 4. The improved DeepLabv3+ network structure.

Considering the characteristics of the dataset, the loss function used in this paper is *CEDiceLoss*, which is a combination of Cross-Entropy loss and Dice Loss [38]. The calculation formulas are shown in Equations (1)–(3), as follows:

$$CEDiceLoss = CELoss + DiceLoss \tag{1}$$

$$CELoss = -\frac{1}{N}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{M} y_i^M \ln\left(\hat{y}_i^M\right) \tag{2}$$

CELoss represents Cross-Entropy loss, N is the total number of pixels, and M is the total number of categories. In this study, the litchi images were divided into branches and the background, so M was two. y_i^M is the annotation information of pixel i, which was set to one when pixel i is a member of the branch class and zero when pixel i is a member of the background class.

$$DiceLoss = 1 - \frac{2|X \cap Y| + \varepsilon}{|X| + |Y| + \varepsilon} \tag{3}$$

DiceLoss represents the Dice Loss function, X is the ground truth, and Y is the pixel category predicted by the semantic segmentation model. The value of ε was set to prevent a zero denominator, and the value of ε was 1×10^{-6}.

2.3. Transfer Learning

Transfer learning [40] is the transfer of knowledge and skills learned from one or more source tasks to a new target task. Since there is no large publicly available dataset for litchi images and training on a small dataset easily causes the overfitting problem, this paper conducted training based on transfer learning. The backbone feature extraction network, the DRN, was pre-trained on the ImageNet dataset, and a large amount of knowledge learned by the DRN was transferred to the dataset used in this paper to facilitate the extraction of litchi images.

2.4. Model Evaluation Metrics

In semantic segmentation tasks, common evaluation metrics include the mean Intersection over Union (*mIoU*) and mean Pixel Accuracy (*mPA*) [41]. In this paper, *mIoU* and *mPA* were used as evaluation metrics to measure the performance of the model on the test set. The model inference speed is expressed in FPS. Suppose that the dataset has $k + 1$ categories (including the background), and in this paper, $k = 1$. p_{ii} represents the number of pixels that were predicted to belong to category I; that is, the number of pixels that were predicted correctly. p_{ij} represents the number of pixels that belonged to category i that were predicted to belong to category j, and p_{ji} represents the number of pixels that belong to category j but were predicted to belong to category i.

2.4.1. Mean Intersection over Union

The *mIoU* represents the ratio of the intersection and union of the predicted value and the real value, which reflects the coincidence degree between them. The formula is defined in Equation (4) as follows:

$$mIoU = \frac{1}{k+1} \sum\nolimits_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{4}$$

2.4.2. Mean Pixel Accuracy

The *mPA* represents the proportion of the number of correctly predicted pixels per category to the number of all pixels, which is then averaged over all categories. The formula is defined in Equation (5) as follows:

$$mPA = \frac{1}{k+1} \sum\nolimits_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{5}$$

3. Results

3.1. Experimental Environment

The computer used to train the model included the Windows 10 operating system with an Intel (R) Xeon (R) Gold 6240 CPU, a 2TB solid state drive, and a GPU with 48G video memory. The graphics card was a dual-channel Nvidia GeForce GTX3090. The programming language used to implement the model was Python3.8, and the training was based on the PyTorch deep learning framework, version 1.8. The unified computing device architecture was CUDA11.3.

3.2. Parameter Settings

Using the transfer learning method, the backbone feature extraction network, the DRN, was pre-trained on the ImageNet dataset. Based on the pre-trained weights, the parameters of the DRN network model were adjusted for the litchi image dataset. There are 3416 images in the dataset. 2732 images were randomly selected as the training set, and the remaining 684 images were used as the test set. During the training stage, the input image size was uniformly adjusted to 512 × 512. The batch size was set to eight, the number of training epochs was 100, and the initial learning rate was 0.007. The optimizer adopted the stochastic gradient descent method and the Poly learning rate update strategy.

3.3. Analysis of Experimental Results

3.3.1. Contrast with Transfer Learning

The results of the DeepLabv3+ model segmentation comparison experiments for the DRN-D-22 backbone network with and without transfer learning are shown in Table 1, and the *mIoU* chart is shown in Figure 5. Without transfer learning, the *mIoU* results are unstable and there are large fluctuations. After the 60th epoch, the *mIoU* gradually increased steadily and reached 80.15%, while the *mPA* reached 84.50%. The experiments were conducted under transfer learning conditions, and as shown in Figure 5, the DRN has a relatively good feature extraction capability at the beginning due to the transfer of a large amount of knowledge from the ImageNet dataset to the dataset introduced in this paper. During the first 33 epochs, the *mIoU* was in a phase of rapid growth. During the subsequent epochs, the *mIoU* tended to increase steadily, reaching 86.72%. The *mPA* reached 91.31%. Compared with the case without transfer learning, the *mIoU* and *mPA* improved by 6.57% and 6.81%, respectively.

Table 1. A comparison of the transfer learning results.

Transfer Learning	*mIoU* (%)	*mPA* (%)
No	80.15	84.50
Yes	86.72	91.31

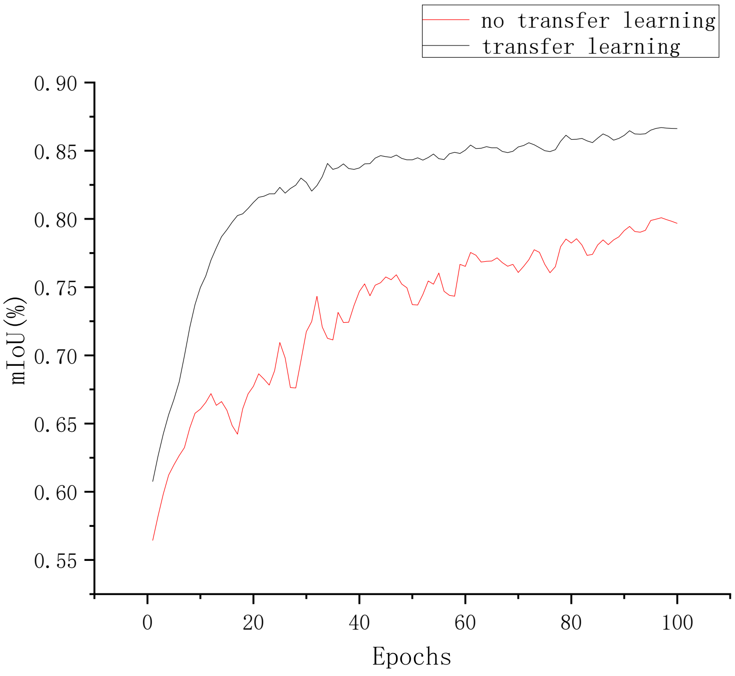

Figure 5. A comparison of the *mIoU* curves for transfer learning.

3.3.2. Ablation Experiment

This section describes the ablation experiment we conducted using the DeepLabv3+ model. Xception-CE indicates the DeepLabv3+ model using Aligned Xception as the backbone network and CE as the loss function, that is, the classical DeepLabv3+ model. DRN-CE and DRN-Dice represent DeepLabv3+ models that used DRN-D-22 as the backbone network, and the loss functions are CE and Dice Loss, respectively. DRN-CE-Dice represents the DeepLabv3+ model that used DRN-D-22 as the backbone network and combined CE and Dice Loss into a loss function. DRN-CE-Dice-CA represents the addition of the CA module to DRN-CE-Dice. The *mIoU* chart is shown in Figure 6, and the model performance evaluation is shown in Table 2. In Table 2, "√" indicates that the marked module is used in the experiment, and no "√" indicates that this module is not added.

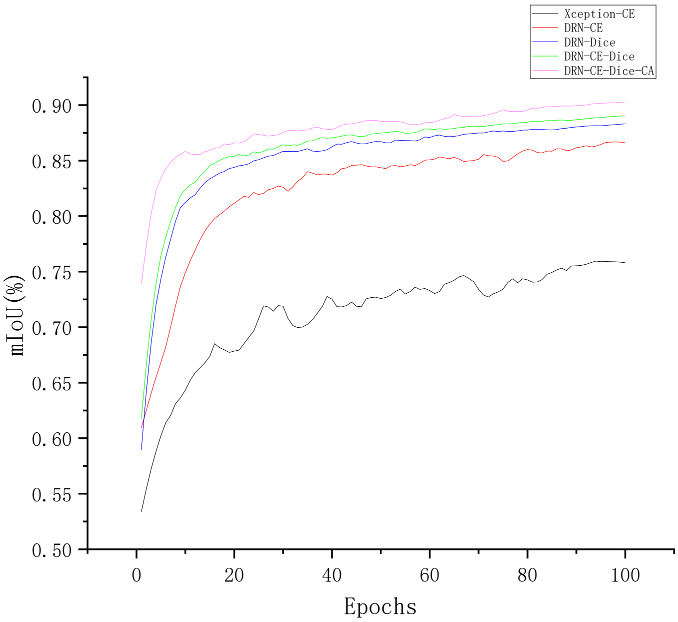

Figure 6. *mIoU* curves of the ablation experiment.

Table 2. A comparison of the ablation results.

Models	Xception	DRN	CE	Dice	CA	*mIoU* (%)	*mPA* (%)	FPS
Xception-CE	√		√			76.71	79.17	18.56
DRN-CE		√	√			86.72	91.31	18.77
DRN-Dice		√		√		88.30	93.75	20.32
DRN-CE-Dice		√	√	√		89.03	94.14	20.31
DRN-CE-Dice-CA		√	√	√	√	90.28	94.95	19.83

As can be seen from Table 2, when the classical DeepLabv3+ model was used to segment the litchi branches, the *mIoU* was only 76.71%, and the *mPA* was 79.17%. However, the *mIoU* and *mPA* of the improved DeepLabv3+ model proposed in this paper were 90.28% and 94.95%, respectively. Compared with the classical DeepLabv3+ network, the model's *mIoU* and *mPA* values were improved by 13.57% and 15.78%, respectively. The improved DeepLabv3+ model proposed in this paper has a frame rate of 19.83 frames per second, which is slightly improved compared with the classical DeepLabv3+ model and improved the semantic segmentation accuracy without affecting the model's inference speed.

Comparing the experimental Xception-CE and DRN-CE, it can be seen that the backbone network of DeepLabv3+ was replaced with DRN-D-22. The input image was downsampled eight times to reduce loss of information during the downsampling process, and the receptive field was increased through combination with dilated convolution. These changes improved the effect of semantic segmentation. Compared with the Xception-CE model, the *mIoU* and *mPA* of the DRN-CE model improved by 10.01% and 12.14%, respectively. Compared with the DRN-CE and DRN-CE-Dice experiments, it can be seen that when CEDiceLoss was used as the loss function during the training process, the model paid more attention to the mining of the foreground region (litchi branches). Compared with the DRN-CE model, the *mIoU* and *mPA* of the DRN-CE-Dice model improved by 2.31% and 2.83%, respectively. Compared with the DRN-CE-Dice and DRN-CE-Dice-CA experiments, it can be seen that adding the CA module while also considering the channel and location information of the multi-scale semantic features acquired by the network helps the model to better segment the litchi branches. Compared with the DRN-CE-Dice model, the *mIoU* of the DRN-CE-Dice-CA model improved by 1.25%.

To show the network improvement effect more intuitively, Xception-CE, DRN-CE, DRN-CE-Dice, and DRN-CE-Dice-CA were selected for testing. The images were labeled by LabelMe, and the results were used as the ground truth. Three images were randomly selected as input images, and the network prediction results are shown in Figure 7.

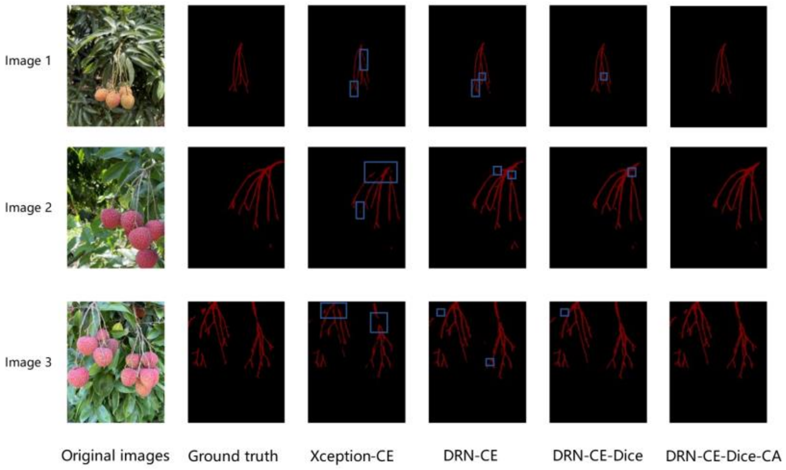

Figure 7. A comparison of the network prediction effects.

As can be seen from Figure 7, Xception-CE found multiple branches that are not segmented completely, as shown in the blue box, and the segmentation is poor. DRN-CE and DRN-CE-Dice produced some incomplete segmentation, as shown in the blue box. On the whole, the segmentation effect of DRN-CE-DICE is better than that of DRN-CE. The areas not segmented by DRN-CE-Dice were segmented completely by DRN-CE-Dice-CA, and overall, DRN-CE-Dice-CA more completely segmented whole litchi branches.

3.3.3. Comparing Other Networks

To compare the segmentation performance of the proposed method with other networks, FCN-8S [9], U-Net [10], SegNet [11], PSPNet [12], HRNetV2 [42], DeepLabv3+ [14], and the improved DeepLabv3+ model proposed in this paper were applied to the dataset introduced in this paper. The *mIoU* of each model is shown in Table 3.

Table 3. A comparison of *mIoU* with different networks.

Models	*mIoU* (%)
U-Net	65.45
SegNet	69.90
FCN-8S	71.95
HRNetV2	73.17
PSPNet	76.75
DeepLabv3+	76.71
Proposed	90.28

Table 3 shows that the *mIoU* of the model proposed in this paper is better than that of the other networks. Due to the complexity of litchi pictures, it is difficult to segment litchi branches accurately. The *mIoU* values produced by U-Net, SegNet, FCN-8S, and HRNetV2 were 65.45%, 69.90%, 71.95%, and 73.17%, respectively. PSPNet and the classical DeepLabv3+ performed comparably on the dataset used in this paper with *mIoU* values of 76.75% and 76.71%, respectively. Compared with the above classical semantic segmentation models, the improved DeepLabv3+ model proposed in this paper has a better segmentation effect with an *mIoU* of 90.28%, which indicates that it can accurately segment litchi branches.

4. Discussion

With the development of automated litchi harvesting technology, many scholars have used deep learning-based image segmentation methods for litchi branch segmentation to improve their accuracy. In this paper, we propose a litchi branch segmentation method. We built a dataset containing 488 litchi images under various conditions and manually labeled litchi branches as label files. To improve the model's generalization ability, offline data augmentation methods were used to expand the number of samples. Since pictures of litchi contain complex orchard backgrounds and litchi branches are small target objects, they are difficult to segment. According to the characteristics of the dataset introduced in this paper, we propose an improved litchi branch segmentation method based on DeepLabv3+; the proposed method enhances the network's feature extraction ability by replacing the classic DeepLabv3+ backbone network. The attention mechanism module was also added to cause the network to pay attention to both the channel and location information to improve the semantic segmentation accuracy. During the training process, a combined Cross-Entropy and Dice coefficient loss function was used to alleviate the negative impact caused by the imbalance between the target area and the background area in the dataset. As a result, the model is more inclined to excavate the litchi branch area. From the experiments, it can be seen that the model proposed in this paper achieved a good segmentation effect and provided a new idea for litchi branch segmentation.

In future work, we will collect images of litchi from different locations and of different varieties to build a richer dataset and obtain a more general expression of litchi branch features. In addition, we will attempt to apply this model to a litchi image dataset captured by an RGB-D camera to obtain more detailed litchi branch location information. At the same time, to reduce the number of model parameters and improve the real-time image segmentation performance, it is necessary to perform pruning operations on the model or select other more efficient models to improve the model's inference speed to better adapt to litchi-picking robots.

5. Conclusions

In this paper, an improved DeepLabv3+ model was proposed for litchi branch segmentation in which the backbone network of the classical DeepLabV3+ model was replaced by DRN-D-22. DRN-D-22 downsamples the input image eight times to reduce information loss during the downsampling process and is combined with dilated convolution to increase the receptive field, which improves the model's feature extraction ability. During the process of model training, a combined Cross-Entropy and Dice coefficient loss function was used, and as a result, the model paid more attention to the litchi branch area. In addition, the CA module was added to cause the model to consider both the channel and location information of the multi-scale semantic features acquired by the network to help the model segment the litchi branches better. The experimental results show that the *mIoU* of the improved DeepLabv3+ model proposed in this paper is 90.28%, the *mPA* is 94.95%, and the FPS is 19.83. According to the experiments, the improved DeepLabv3+ model proposed in this paper can accurately segment litchi branches, which provides powerful technical support that helps litchi-picking robots find branches.

Author Contributions: Conceptualization, J.X., T.J., S.L. and J.L.; methodology, T.J., H.Y., D.S., W.W., S.L. and J.L.; software, T.J., B.C. and J.P.; validation, H.Y., D.S. and W.W.; formal analysis, T.J., B.C. and J.P.; investigation, J.X., T.J., B.C., J.P., X.Z., P.H. and A.X.; resources, H.Y., D.S., W.W., S.L. and J.L.; writing—original draft preparation, T.J.; writing—review and editing, J.X.; visualization, X.Z., P.H. and A.X.; supervision, J.X.; project administration, S.L. and J.L.; funding acquisition, J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Co-constructing Cooperative Project on Agricultural Sci-tech of New Rural Development Research Institute of South China Agricultural University (No. 2021XNYNYKJHZGJ032). It was also partly supported by Laboratory of Lingnan Modern Agriculture Project (No. NT2021009); the Guangdong Province Science and Technology Special Fund ("Major Project + Task List") project, China (No. 2020020103); Guangdong Province Rural Revitalization Strategy Projects (No. TS-1-4); Basic and Applied Basic Research Project of Guangzhou Basic Research Plan in 2022 (No. 202201010077); the China Agriculture Research System of MOF and MARA, China (No. CARS-32-14); National Natural Science Foundation of China (No. 32271997); and National College Students' innovation and entrepreneurship training program(No. 202110564044).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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 policy of the organization.

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

References

1. Xie, J.; Chen, Y.; Gao, P.; Sun, D.; Xue, X.; Yin, D.; Han, Y.; Wang, W. Smart Fuzzy Irrigation System for Litchi Orchards. *Comput. Electron. Agric.* **2022**, *201*, 107287. [CrossRef]
2. Zhu, Q.; Lu, R.; Lu, J.; Li, F. Research status and development trend of litchi picking machinery. *For. Mach. Woodwork. Equip.* **2021**, *49*, 11–19. [CrossRef]
3. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [CrossRef] [PubMed]
4. Zheng, T.; Jiang, M.; Feng, M. Vision based target recognition and location for picking robot: A review. *Chin. J. Sci. Instrum.* **2021**, *42*, 28–51. [CrossRef]
5. Xiong, J.; Lin, R.; Liu, Z.; He, Z.; Tang, L.; Yang, Z.; Zou, X. The Recognition of Litchi Clusters and the Calculation of Picking Point in a Nocturnal Natural Environment. *Biosyst. Eng.* **2018**, *166*, 44–57. [CrossRef]
6. Zhuang, J.; Hou, C.; Tang, Y.; He, Y.; Guo, Q.; Zhong, Z.; Luo, S. Computer Vision-Based Localisation of Picking Points for Automatic Litchi Harvesting Applications towards Natural Scenarios. *Biosyst. Eng.* **2019**, *187*, 1–20. [CrossRef]
7. Xiong, J.; He, Z.; Lin, R.; Liu, Z.; Bu, R.; Yang, Z.; Peng, H.; Zou, X. Visual Positioning Technology of Picking Robots for Dynamic Litchi Clusters with Disturbance. *Comput. Electron. Agric.* **2018**, *151*, 226–237. [CrossRef]
8. Luo, L.; Zou, X.; Xiong, J.; Zhang, Y.; Peng, H.; Lin, G. Automatic positioning for picking point of grape picking robot in natural environment. *Trans. Chin. Soc. Agric. Eng.* **2015**, *31*, 14–21.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–15 June 2015; pp. 3431–3440.
10. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
11. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
12. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
13. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef]
14. 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.
15. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9157–9166.

17. Li, Q.; Jia, W.; Sun, M.; Hou, S.; Zheng, Y. A Novel Green Apple Segmentation Algorithm Based on Ensemble U-Net under Complex Orchard Environment. *Comput. Electron. Agric.* **2021**, *180*, 105900. [CrossRef]
18. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
19. Cai, C.; Tan, J.; Zhang, P.; Ye, Y.; Zhang, J. Determining Strawberries' Varying Maturity Levels by Utilizing Image Segmentation Methods of Improved DeepLabV3+. *Agronomy* **2022**, *12*, 1875. [CrossRef]
20. Ning, Z.; Luo, L.; Liao, J.; Wen, H.; Wei, H.; Lu, Q. Recognition and the optimal picking point location of grape stems based on deep learning. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 222–229.
21. Xue, J.; Wang, Y.; Qu, A.; Zhang, J.; Xing, Z.; Wei, H.; Sun, H. Image segmentation method for Lingwu long jujubes based on improved FCN-8s. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 191–197.
22. Yang, W.; Duan, L.; Yang, W. Deep learning-based extraction of rice phenotypic characteristics and prediction of rice panicle weight. *J. Huazhong Agric. Univ.* **2021**, *40*, 227–235. [CrossRef]
23. Li, J.; Tang, Y.; Zou, X.; Lin, G.; Wang, H. Detection of Fruit-Bearing Branches and Localization of Litchi Clusters for Vision-Based Harvesting Robots. *IEEE Access* **2020**, *8*, 117746–117758. [CrossRef]
24. Peng, H.; Xue, C.; Shao, Y.; Chen, K.; Xiong, J.; Xie, Z.; Zhang, L. Semantic Segmentation of Litchi Branches Using DeepLabV3+ Model. *IEEE Access* **2020**, *8*, 164546–164555. [CrossRef]
25. Peng, H.; Zhong, J.; Liu, H.; Li, J.; Yao, M.; Zhang, X. Resdense-Focal-Deeplabv3+ Enabled Litchi Branch Semantic Segmentation for Robotic Harvesting 2022. Available online: https://ssrn.com/abstract=4162665 (accessed on 12 August 2022).
26. Zhong, Z.; Xiong, J.; Zheng, Z.; Liu, B.; Liao, S.; Huo, Z.; Yang, Z. A Method for Litchi Picking Points Calculation in Natural Environment Based on Main Fruit Bearing Branch Detection. *Comput. Electron. Agric.* **2021**, *189*, 106398. [CrossRef]
27. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A Visual Detection Method for Nighttime Litchi Fruits and Fruiting Stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [CrossRef]
28. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [CrossRef]
29. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv* **2016**, arXiv:1412.7062.
30. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
31. Yu, F.; Koltun, V.; Funkhouser, T. Dilated Residual Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–16 July 2017; pp. 636–644.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122. [CrossRef]
34. Su, F.; Zhao, Y.; Wang, G.; Liu, P.; Yan, Y.; Zu, L. Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment. *Agronomy* **2022**, *12*, 1638. [CrossRef]
35. Chen, Z.; Wu, R.; Lin, Y.; Li, C.; Chen, S.; Yuan, Z.; Chen, S.; Zou, X. Plant Disease Recognition Model Based on Improved YOLOv5. *Agronomy* **2022**, *12*, 365. [CrossRef]
36. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
37. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
38. Jadon, S. A Survey of Loss Functions for Semantic Segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Viña del Mar, Chile, 27–29 October 2020; pp. 1–7.
39. Milletari, F.; Navab, N.; Ahmadi, S.-A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
40. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
41. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857. [CrossRef]
42. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.