remote sensing

MDPI

Article

Branch Identification and Junction Points Location for Apple Trees Based on Deep Learning

Siyuan Tong, Yang Yue, Wenbin Li, Yaxiong Wang, Feng Kang * and Chao Feng

School of Technology, Beijing Forestry University, Key Lab of State Forestry and Grassland Administration for Forestry Equipment and Automation, Beijing 100083, China
* Correspondence: kangfeng98@bjfu.edu.cn

Abstract: Branch identification is key to the robotic pruning system for apple trees. High identification accuracy and the positioning of junction points between branch and trunk are important prerequisites for pruning with a robotic arm. Recently, with the development of deep learning, Transformer has been gradually applied to the field of computer vision and achieved good results. However, the effect of branch identification based on Transformer has not been verified so far. Taking Swin-T and Resnet50 as a backbone, this study detected and segmented the trunk, primary branch and support of apple trees on the basis of Mask R-CNN and Cascade Mask R-CNN. The results show that, when Intersection over Union (IoU) is 0.5, the bbox mAP and segm mAP of Cascade Mask R-CNN Swin-T are the highest, which are 0.943 and 0.940; as for the each category identification, Cascade Mask R-CNN Swin-T shows no significant difference with the other three algorithms in trunk and primary branch; when the identified object is a support, the bbox AP and segm AP of Cascade Mask R-CNN Swin-T is significantly higher than that of other algorithms, which are 0.879 and 0.893. Next, Cascade Mask R-CNN SW-T is combined with Zhang & Suen to obtain the junction point. Compared with the direct application of Zhang & Suen algorithm, the skeleton obtained by this method is advantaged by trunk diameter information, and its shape and junction points position are closer to the actual apple trees. This model and method can be applied to follow-up research and offer a new solution to the robotic pruning system for apple trees.

Keywords: automated detection; deep learning; branch segmentation; machine vision; skeletonization

Citation: Tong, S.; Yue, Y.; Li, W.; Wang, Y.; Kang, F.; Feng, C. Branch Identification and Junction Points Location for Apple Trees Based on Deep Learning. *Remote Sens.* **2022**, *14*, 4495. https://doi.org/10.3390/rs14184495

Academic Editor: András Jung

Received: 8 July 2022
Accepted: 5 September 2022
Published: 9 September 2022

1. Introduction

As an important part of apple orchard management and maintenance, pruning can change the relationship between fruit trees and the environment, improve ventilation and light transmission conditions, and effectively improve fruit quality and fruit yield [1]. According to the degree of pruning, it can be divided into nonselective pruning and precision pruning [2]. Nonselective pruning is carried out through shaping and pruning by mechanized pruning equipment with a large number of wrong pruning and missing pruning due to its nonselective nature, resulting in reduced fruit production [3]; precision pruning mainly relies on intense and inefficient manual pruning of workers standing on a ladder or orchard lift platform through pruning shears. At present, the labor cost in orchard management and maintenance accounts for 56% of the total cost, among which pruning accounts for about 20% of the total cost [4]. As the population ages and rural labor migrates [5], the development of apple industry has been restricted. Selective pruning by robotics is an effective approach to this problem [6].

The information on fruit tree branches is the premise of robot pruning [7]. Now, a large number of scholars have used images and point clouds to detect trunks and branches. Ji et al. [8] used the contrast limited adaptive histogram equalization to identify branches, and the identification rate reached 94%, which was better than OTSU (maximum between-class variance) and histogram algorithm. Botterill et al. [9] designed a robot system for

the automatic pruning of grape vines to obtain images through trinocular stereo cameras and generate a 3D model of grape vines by triangulating feature matches. When the robot moved at 0.25 m/s, the error was less than 1%. Similarly, in dynamic outdoor environments, Tabb et al. [10] located low-texture regions through superpixels, and then modeled and segmented branches by employing Gaussian mixture model, realizing unsupervised detection. Karkee et al. [11] transformed the preprocessed depth image into 3D point clouds and used the medial axis thinning algorithm to generate the apple tree skeleton and identify of apple tree trunks and branches, through which the accuracy of trunk detection reached 100% and that of branch identification reached 77%. Medeiros et al. [12] used laser sensor to obtain the point cloud data of apple trees, used split-and-merge clustering algorithm to divide the point cloud into trunk candidates, junction point candidates and branches, and, finally, obtained the 3D model of the fruit tree through further fitting and refining, with the detection accuracy of main branches of 98% and the diameter error of 0.6 cm. Mack et al. [13] obtained the tree surface information through the region growing algorithm, and reconstructed the 3D model of grape branches through the Ransac, with a reconstruction accuracy of 98%. In the above studies, traditional computer vision was used to detect the branches of fruit trees. However, because of the complex actual orchard environment, the image-based method would be greatly affected by illumination and noise. Botterill et al. [9] and Tabb et al. [10] carried out artificial lighting and shading, but its huge pruning system limited its application scope. In the identification method based on point clouds, the 3D model needs to be obtained through multi-angle registration [11]. Its real-time performance is poor, the reconstruction accuracy depends on the equipment accuracy, and the price of laser sensors is expensive, which limits the promotion of robotic pruning system.

With the development of AI, the object detection and segmentation technology based on deep learning has been gradually applied to tree branch identification and tested in the orchard environment, and obtained good robustness and precision. Zhang et al. [14] used R-CNN to train Pseudo-Color Image and Depth of apple trees, obtained an average branch detection accuracy of 85.5% and the skeleton fitting correlation coefficient of 0.91, and extracted centroid to apply it to shake-and-catch apple harvesting. Additionally, Zhang et al. [15] also used Alexnet, VGG16 and VGG19 to detect the trunk, branches and apples based on Faster R-CNN model to determine the picking shaking locations, of which VGG19 had the best detection effect, with an mAP of 82.4%. Likewise, Zhang et al. [16] obtained the best results for segmenting trunks, branches, apples and leaves, with an average segmentation accuracy of 97% based on CNN ResNet-18. Majeed et al. [17] segmented the trunk, branches and trellis wires in Foreground-RGB images through Segnet, with the average accuracy of the trunk and branches of 82% and 89%, and introduced boundary-F1 score to the evaluation index, which improved the accuracy of branch boundary detection compared to [14]; on the basis of this method, Majeed et al. [18] also obtained the best segmentation model for grapevine (FCN-VGG16), and fitted the main grapevine by a sixth-degree polynomial. Focusing on the occlusion by leaves, Majeed et al. [19] used ResNet and Faster R-CNN to first detect the visible parts of grapevine canopies, and then performed polynomial fitting on the centroids of detected cordon to obtain the skeleton model. Chen et al. [20] compared the segmentation of U-Net, DeepLabv3 and Pix2Pix on occluded apple branches, and introduced depth difficulty index due to the depth map trained in [14] greatly affected by strong lighting and twigs, but the segmentation effect of the three models was not ideal. Yang et al. [21] used the branch segment merging algorithm to gain the citrus branches based on Mask R-CNN, with the identification of accuracy of 96.27%, and mapped the segmented RGB images to the depth images, with the error of the branch diameter of 1.17 mm. Cuevas-Velasquez et al. [22] used Fully Convolutional Segmentation Network (FCSN) to segment the branches, with the branch segmentation accuracy of 88%, combined the result with the disparity image to conduct 3D reconstruction, and then finished the rose pruning through the help of the robotic arm. Based on U-Net, Liang et al. [23] used the momentum optimization stochastic gradient descent

method as the optimizer to segment the fruit and stem of litchi, effectively improving the segmentation accuracy to 95.54%. In the research of deep learning based on point clouds, Ma et al. [24] used Azure Kinect DK to collect the point cloud data of jujube trees, trained the preprocessed point cloud information through SPGnet, and the segmentation accuracy of the trunk and branches were 93% and 84%. You et al. [25] determined the branch boundary based on CNN, proposed a population-based search method according to the size and structure of the cherry tree, performed 3D reconstruction on it, and, finally, obtained a skeletonized model. In the above-mentioned studies, deep learning algorithms were successfully applied in branch detection, which have effectively solved the limitations of traditional machine vision methods.

CNN, FCN and other algorithms are used in the existing studies to detect and segment branches and achieved good results. However, with the optimization and innovation of algorithms in recent years, the accuracy of object detection and segmentation has been greatly improved. On the basis of Mask R-CNN, Cai et al. [26] put forward Cascade Mask R-CNN, which solved the interference of threshold setting on the results of single-module detection network through cascade structure, and thus greatly improved the accuracy of instance segmentation. In addition, as the most commonly used model for natural language processing (NLP), Transformer performs better in capturing global context information and extracts more powerful features than CNN [27]. With the proposal of Swin Transformer [28], this model has been applied to instance segmentation. With a CNN-like hierarchical architecture, it segments by utilizing shifted windows and carries out computing of self-attention in windows, which has effectively reduced the amount of calculation and greatly improves the accuracy of target detection and semantic segmentation, so the model has already been applied to the detection of strawberries [29] and grapes [30], and achieved good results in the fields of remote sensing [31–36], materials [37,38] and medicine [39,40].

To conclude, in the branch detection research based on deep learning, the main branch skeleton was determined by fitting curves with certain error [14–16]. As the major object of apple tree pruning, primary branches are featured by large number and diameter, which is the main reason for their large labor consumption [41]. Therefore, the study took primary branches as the main identification object. While the identification of primary branch skeleton and junction points is the premise of robotic pruning, the above-discussed research does not deal with the position of junction points. Additionally, in the field of branch detection, there are few relevant studies using Cascade Mask R-CNN and Swin Transformer. So, taking Swin-T and Resnet50 as a backbone, the study identified the trunks, primary branches and supports of apple fruit trees to obtain the primary branch skeleton and junction points based on Mask R-CNN and Cascade Mask R-CNN model. Different from directly extracting the whole skeleton of the fruit tree to obtain the junction points, this study used the obtained optimal algorithm to extract the primary branch skeleton and added it to the trunk image to obtain the junction points. The specific study objectives are:

(1) Based on the above backbone and model, the detection and segmentation effects of the four algorithms (Mask R-CNN Resnet50, Mask R-CNN R50; Mask R-CNN Swin-T, Mask R-CNN SW-T; Cascade Mask R-CNN Resnet50, Cascade Mask R-CNN R50; and Cascade Mask R-CNN Swin-T, Cascade Mask R-CNN SW-T) on apple tree trunks, primary branches and supports in the orchard environment are compared to find the optimal one.
(2) According to the segmentation results obtained by the optimal algorithm, the skeleton structure of the apple tree is constructed with the support of a skeletonization algorithm, so as to locate the junction points between the trunk and the primary branch.

2. Materials and Methods

2.1. Experimental Site

The experimental site was a modern apple orchard (114°33′N, 38°92′E) in Fuping County, Baoding City, Hebei Province, China. The orchard was located in a hilly mountainous area, with Red Fuji as its product. The apple trees were tall spindle apple trees.

The number of primary branches on the main trunk was about 30, presented in a spiral. The average tree age was 3 years, the tree height was 3.5 m, the diameter of the branch junction point was about 10–35 mm, the row spacing was about 4.5 m, and the plant spacing was about 1.5 m. The orchard adopted the ridge cultivation mode and the ground ridge height was 0.3–0.4 m. The apple trees were deciduous trees and the precision pruning was conducted from January to March. The data were collected in December 2021, when the apple tree was in dormant season and the leaves had fallen off. During the data collection, no artificial treatments, such as pruning or bending, were performed on the apple branches. The experimental site is shown in Figure 1.

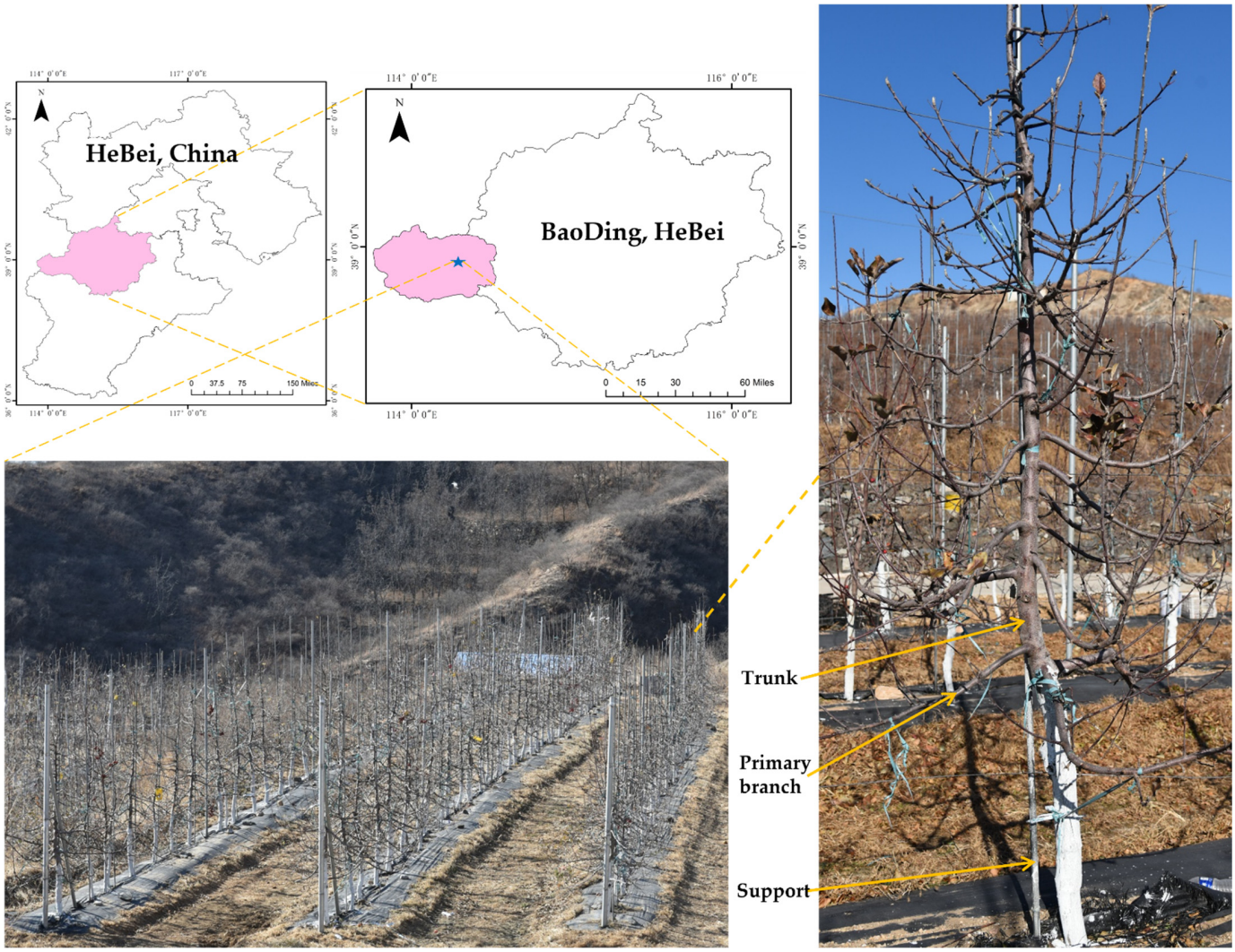


Figure 1. Orchard and apple trees on the experimental site.

2.2. Data Acquisition

The diameter of some primary branches was relatively small, and there were a large number of other fruit branches in the shooting background. Therefore, to ensure the accuracy of the labeling, the research used a high-resolution color camera (Nikon D750) to collect the color information of the apple branches, with 24.32 million effective pixels and a maximum resolution of 6016 × 4016. The camera was fixed horizontally on a tripod. According to the structure principle and working distance of the pruning robot [9], the camera was placed at a height of 1.4 m from the ground and a distance of 1–1.1 m from the trunk center, and randomly selected different time periods (morning, noon, and afternoon) and angle (frontlighting, backlighting, and sidelighting) to collect local images

of the fruit trees. Based on OpenCV, data augmentation was performed on RGB images through brightness adjustment, noise addition, and image flipping, with at least one of them randomly selected for each image. Additionally, the training samples and the testing samples were divided in an 8:2 ratio. Finally, 2000 images were used for training and 500 images for testing. The support of fruit trees is able to interfere with the trunk identification, and thus the robotic arm needs to be controlled to avoid obstacles in the process of robotic pruning. Therefore, the training samples were labeled as tree trunk, primary branch and support through Lableme. The number of labels for the trunks, primary branches and supports was, respectively, 2000, 18,123 and 1865, and training was performed on images of apple tree branches in the format of the coco dataset.

2.3. Branch Segmentation Algorithm

Mask R-CNN [42] and Cascade Mask R-CNN [26] are applied to instance segmentation to complete the three tasks of detection, classification and segmentation of target objects. Different from semantic segmentation, instance segmentation can not only classify and label each pixel in the image, but also identify different instance individuals in the same foreground semantic category. The overall network structure of Mask R-CNN is shown in Figure 2, which mainly includes the backbone module, RPN layer, RoI Align layer, fully connected RCNN module and Mask branch module. First of all, the backbone network is used to extract and fuse the multi-scale information of the input image data to obtain feature maps of different sizes; bounding boxes that may contain target objects are generated by RPN layer, and then the candidate boxes are screened to obtain RoI; RoI Align matches and aligns instance bounding boxes with feature maps; the FC layer module is used to predict classification and regression; the target pixel area is segmented by the Mask branch module; and, finally, the detection and segmentation of tree trunks, primary branches and supports are realized.

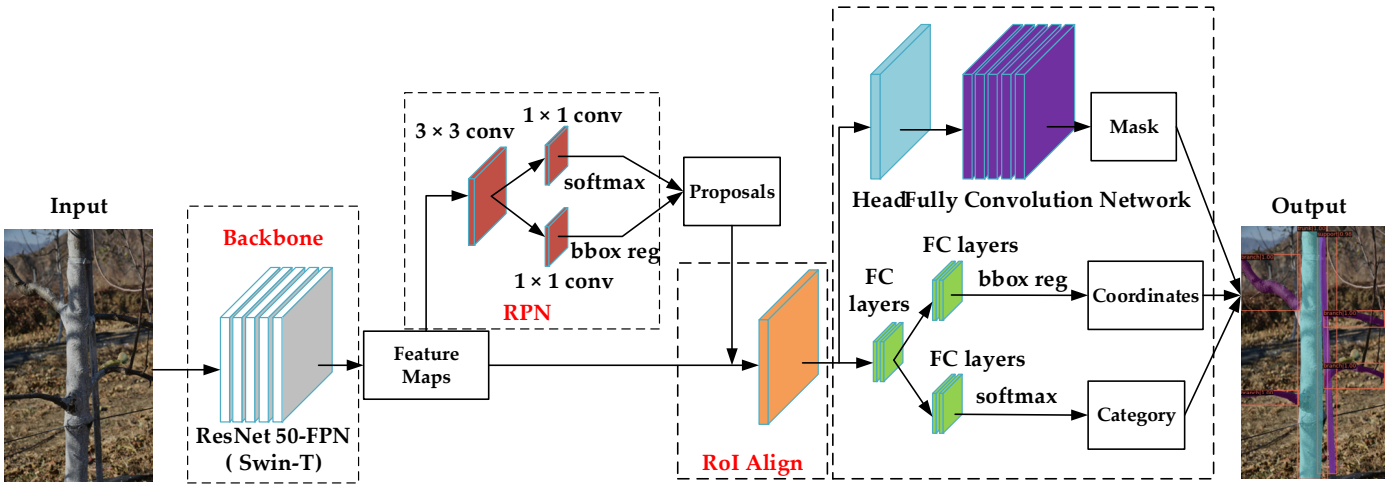


Figure 2. Frame diagram of apple tree segmentation based on Mask R-CNN.

In order to improve the detection accuracy during the training process, higher-precision bounding boxes can be obtained by increasing the IoU threshold for determining positive samples, but this will cause a reduction in the number of bounding boxes, over-fitting and decline in detection performance. In response to this problem, Cascade Mask R-CNN adopts a cascade structure of multiple detectors on the basis of Mask R-CNN to input the regression frame output by the previous detector into the next one. By gradually increasing the IoU threshold, the false detection frame is screened out, the loss of positive samples is reduced, and, ultimately, the detection accuracy is improved. The structure of Cascade Mask R-CNN is shown in Figure 3, in which H is the network head, C is the classification result, B is the detection result of the bounding box and M is the mask.

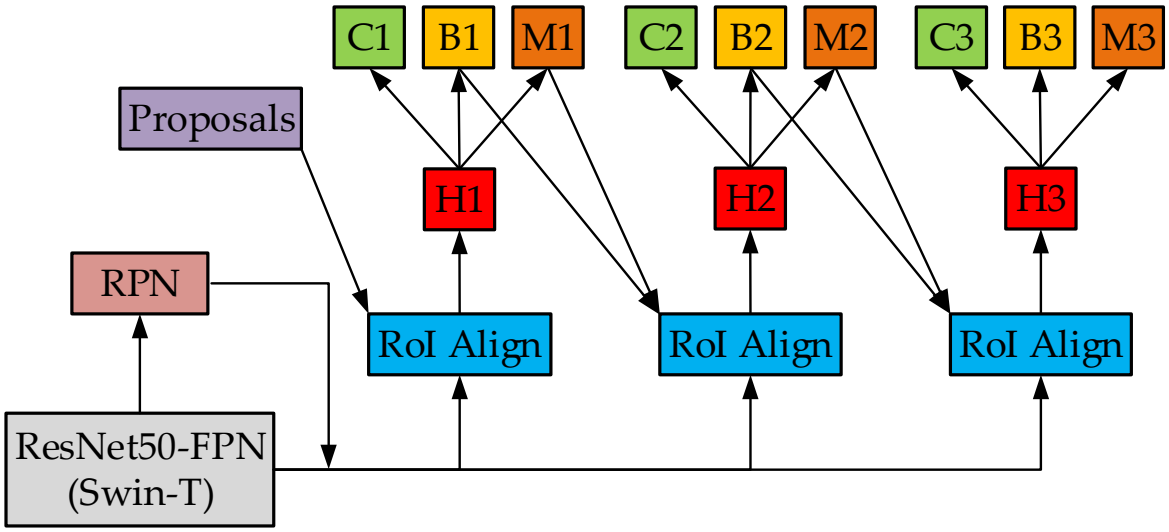


Figure 3. Cascade Mask R-CNN network structure diagram.

Based on Mask R-CNN and Cascade Mask R-CNN model, the study selected ResNet50 and Swin-T as the backbone. As a deep residual network, ResNet50 can solve the problem of gradient disappearance during model training through residual mapping [43], which has been widely used in the field of image detection.

Relying on attention mechanism, Transformer models the global dependencies of input and output, and has achieved excellent achievements in natural language processing (NLP). In recent years, Transformer has been gradually applied in the CV field, and Swin Transformer has performed well in image classification, object detection and semantic segmentation. Swin Transformer provides 4 models, namely, Swin-T, Swin-S, Swin-B and Swin-L. The computational complexity of Swin-T is similar to that of ResNet50 [28]. Therefore, the Swin-T backbone was selected in this research. Its overall framework is shown in Figure 4. First of all, the apple tree RGB image with size $H \times W$ was divided into nonoverlapping patches of size 4×4 through Patch Partition. The flattened feature dimension of each patch was 48, so the whole was a 2D sequence of $\frac{H}{4} \times \frac{W}{4} \times 48$ dimensions. Subsequently, there were four stages: the first stage transformed it into an arbitrary dimension C through Linear Embedding; then, down-sampling was conducted by the Patch Merging layer in the next 3 stages, and each group of 2×2 adjacent pixels were merged into a patch, and the pixels in the same position in each patch were spliced to halve the dimension, and the output size of each stage was $\frac{H}{4} \times \frac{W}{4} \times C$, $\frac{H}{8} \times \frac{W}{8} \times 2C$, $\frac{H}{16} \times \frac{W}{16} \times 4C$ and $\frac{H}{32} \times \frac{W}{32} \times 8C$. Swin Transformer Block was used to calculate the relationship between patches with two structures, namely, W-MSA and SW-MSA. W-MSA divided the feature map into several windows, as shown in Figure 5b, each of which contained the same number of patches and performed self-attention computation individually. As a result, the computation was reduced to less than MSA in VIT [44]. Different from the segmentation method of W-MSA, the segmentation result of SW-MSA was divided into four regions: red, blue, green and gray, which is shown in Figure 5c. Then, the red, blue and green windows were moved to the right and bottom of the image, respectively, as shown in Figure 5d. To conclude, the shifted windows effectively enhance the information interaction between windows.

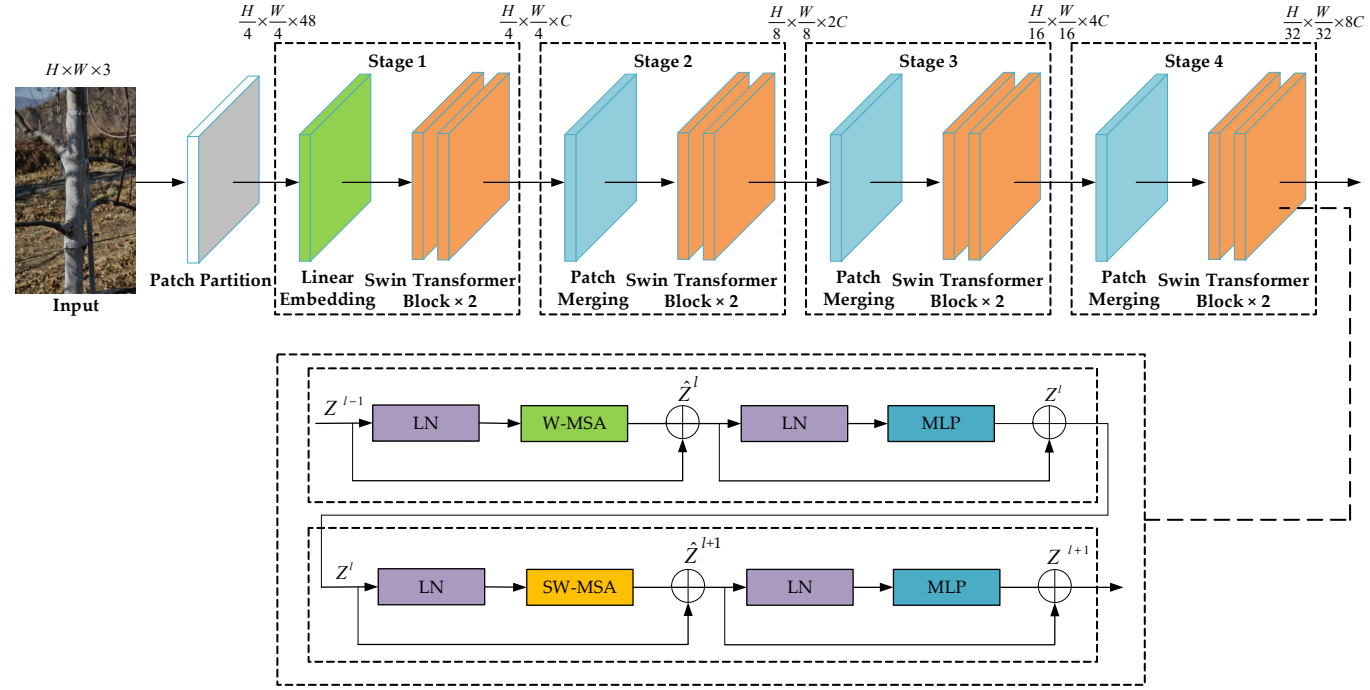
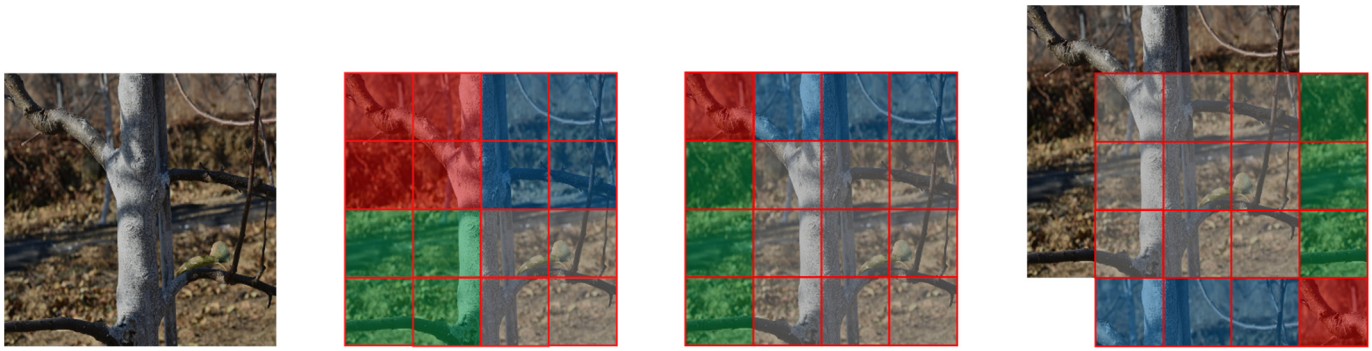


Figure 4. Swin Transformer overall frame diagram.



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

Figure 5. Shifted windows based on self-attention mechanism: (**a**) input image; (**b**) W-MSA segmentation method; (**c**) SW-MSA segmentation method; (**d**) shifted windows.

2.4. Parameter Settings

The computer for training was configured as win10, the GPU was 16 G Nvidia GTX3070, and the software environment was anaconda 3 and pytorch1.9.1. The model was from MMDetection (https://mmdetection.readthedocs.io/en/latest, accessed on 8 December 2021) and trained in Pycharm software. After many times of training and debugging, batch size was set to 2 and epoch to 100 according to the performance of the graphics card. The warm-up optimization scheme was used in the training process, and the learning rate was lowered when the number of epochs was 75 and 90, respectively. Other parameters of each model are shown in Table 1.

Table 1. Other parameters of each model.

Model	Backbone	Optimizer	Initial Learning Rate	Weight Decay
Mask R-CNN	Resnet50	SGD	0.02	0.0001
	Swin-T	AdamW	0.0001	0.05
Cascade Mask R-CNN	Resnet50	SGD	0.02	0.0001
	Swin-T	AdamW	0.0001	0.05

2.5. Evaluation Indicators

In order to evaluate the detection and segmentation effect of the algorithm, Intersection over Union (IoU), average precision (AP) and mean average precision (mAP) were used as the main evaluation indicators. IoU, also known as the Jaccard index (Equation (1)), is used to measure how well the bounding box fits the target; AP is the area enclosed by the P–R curve (the relationship between recall rate and precision rate) and the horizontal axis (Equation (2)), and a larger AP value means higher accuracy; mAP is the mean average precision (Equation (3)), which is one of the most important indicators to measure the performance of target detection and semantic segmentation models. In addition, the calculation equations of recall and precision are shown in Equations (4) and (5).

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{1}$$

$$\text{AP} = \int_0^1 Precision \cdot Recalldr \tag{2}$$

$$\text{mAP} = \frac{\sum_{n=1}^{N} \text{AP}(n)}{N} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

2.6. Location of Branch Junction Points

The determination of the junction point between the primary branch and the trunk is the premise of robotic pruning, which can be realized by skeleton extraction. First, the image needs to be preprocessed (filtering, alignment, smoothing, threshold segmentation, and opening and closing operations) to obtain a binary image of the fruit tree. Then, the skeleton structure of fruit trees can be extracted by skeleton extraction algorithm. Cuevas-Velasquez et al. [22] adopted five algorithms, including Zhang & Suen, parallel thin, and medial axis, to extract branch skeleton. Among them, Zhang & Suen worked best, with F_1 of 91.06%. Therefore, the research used Zhang & Suen [45] to extract the skeleton of the branches.

The algorithm creates a 3×3 pixel window, as shown in Figure 6a. By traversing the central pixel point P1 and its neighboring pixels (P2–P9), it is analyzed whether the neighbors of the pixel meet the requirements of Equations (6)–(8). $N(P1)$ represents the number of neighbor pixels with a value of 0, and $S(P1)$ represents the number of times that the neighbor pixel value changes from 0 to 1 from P2 to P9. The points satisfying Equations (6) and (7) are deleted in the first traversal, as shown in Figure 6b; the points that satisfy Equations (6) and (8) are deleted during the second traversal, as shown in Figure 6c; the skeleton is finally output until no pixels are marked for deletion. Since the output skeleton is connected by a single pixel, the position of the junction point between the branch and the trunk is inconsistent with the actual one. To solve this problem, this study,

according to the prediction results of deep learning, obtained the trunk and the primary branches, respectively, used the Zhang & Suen to extract the central axis skeleton of the primary branches, added the pixels of the central axis image of the primary branch and the trunk image to obtain the apple tree skeleton with the information of trunk diameter and determined its junction points, as well as compared and analyzed it with the method of directly extracting the skeleton through Zhang & Suen.

$$2 \leq N(P1) \leq 6,\ S(P1) = 1 \tag{6}$$

$$\prod_{i\,=\,2,\,4,\,6} P_i = 0,\quad \prod_{i\,=\,4,\,6,\,8} P_i = 0 \tag{7}$$

$$\prod_{i=2,\,4,\,8} P_i = 0,\quad \prod_{i=2,\,6,\,8} P_i = 0 \tag{8}$$

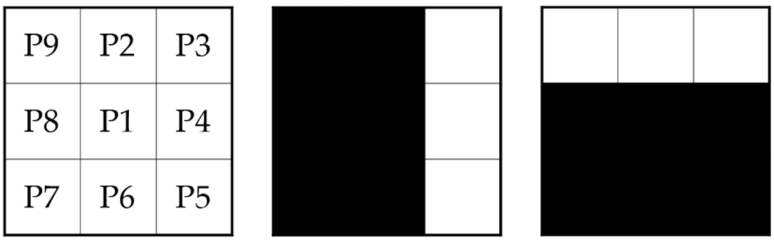


Figure 6. Pixel diagram: (**a**) 3 × 3 pixel window; (**b**) pixel window satisfying the first traversal; (**c**) pixel window satisfying the second traversal.

3. Results

3.1. Segmentation Results

Table 2 shows the bbox mAP and segm mAP of the four algorithms at IoU of 0.5, 0.75 and 0.5:0.95. Models are typically tested at IoU = 0.5 [26]. Therefore, mAP with an IoU of 0.5 is selected for further analysis. When IoU is 0.5, the bbox mAP and segm mAP of Cascade Mask R-CNN are significantly higher than those of Mask R-CNN, those of Cascade Mask R-CNN R50 are improved by 1.8% and 1.8% compared with Mask R-CNN R50, and those of Cascade Mask R-CNN SW-T are improved by 5.2% and 5.3% compared with Mask R-CNN SW-T. Under the same model, the backbone Swin-T is significantly better than Resnet50. Compared with Mask R-CNN R50, the bbox mAP and segm mAP of Mask R-CNN SW-T improves by 7.8% and 8.1%. Compared to Cascade Mask R-CNN R50, those of Cascade Mask R-CNN SW-T improve by 11.5% and 11.8%. Cascade Mask R-CNN SW-T has the highest bbox mAP and segm mAP, which are 0.943 and 0.940, respectively.

Table 2. mAP values of four algorithms under different IOU thresholds.

Model	Backbone	Bbox mAP			Segm mAP		
		IoU0.5	IoU0.75	IoU0.5:0.95	IoU0.5	IoU0.75	IoU0.5:0.95
Mask R-CNN	Resnet50	0.831	0.822	0.787	0.826	0.777	0.646
	Swin-T	0.896	0.791	0.654	0.893	0.776	0.659
Cascade Mask R-CNN	Resnet50	0.846	0.846	0.838	0.841	0.804	0.673
	Swin-T	0.943	0.900	0.781	0.940	0.845	0.709

When the IoU is 0.5, the curves of bbox mAP and segm mAP of the four algorithms' change with epoch are shown in Figure 7. With the increase in epoch, the mAP of each algorithm gradually tends to converge, and the accuracy of Swin-T is always better than that of Resnet50 in the whole process. This is consistent with the above conclusions. The loss mask curve of each algorithm is shown in Figure 8. It can be seen from Figure 8 that the curve tends to converge with the increase in epoch, which proves that the training

results are credible and the convergence effect of Cascade Mask R-CNN is better than that of Mask R-CNN.

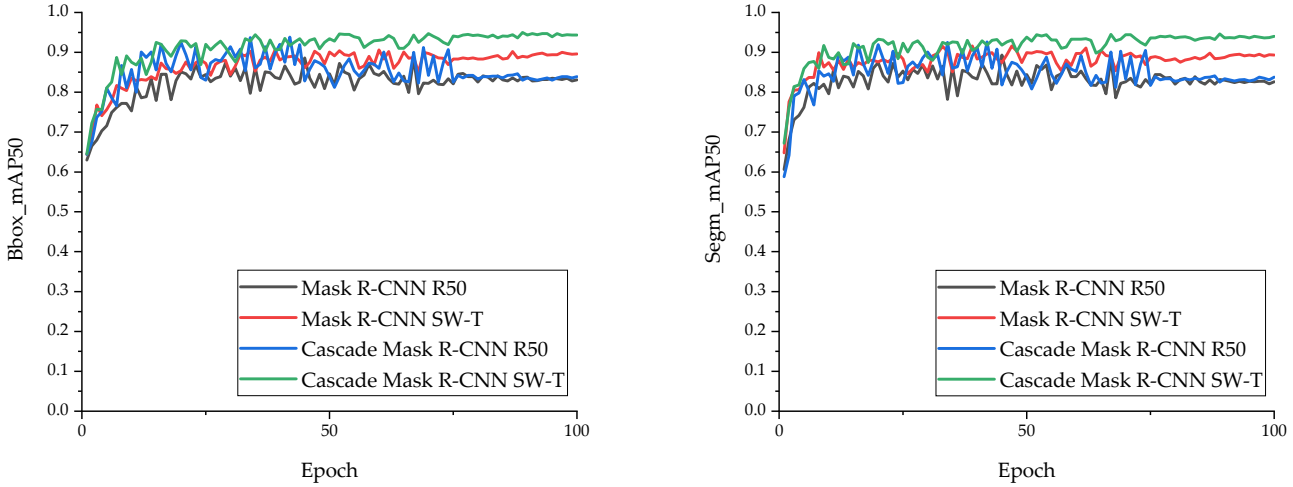


Figure 7. Variation curve of mAP50 of four algorithms with epoch: (**a**) changes in bbox mAP50 with epoch; (**b**) changes in segm mAP50 with epoch.

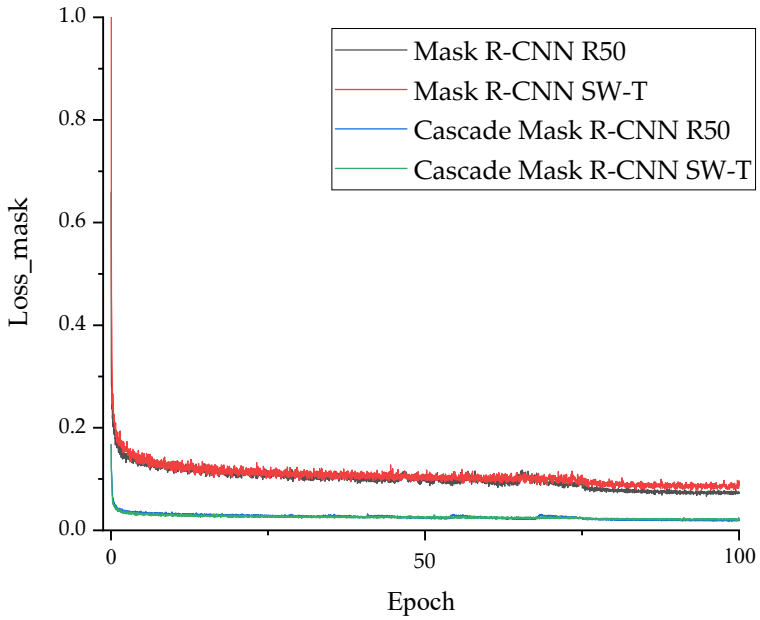
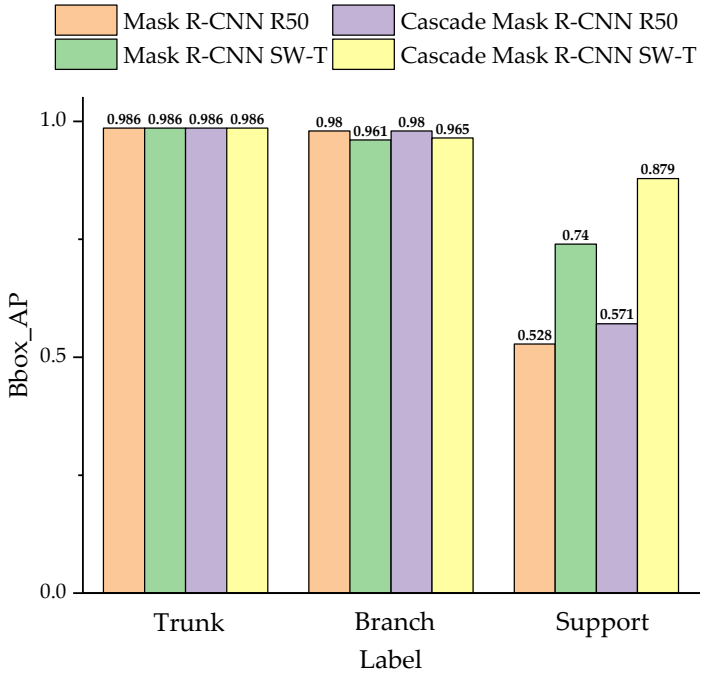


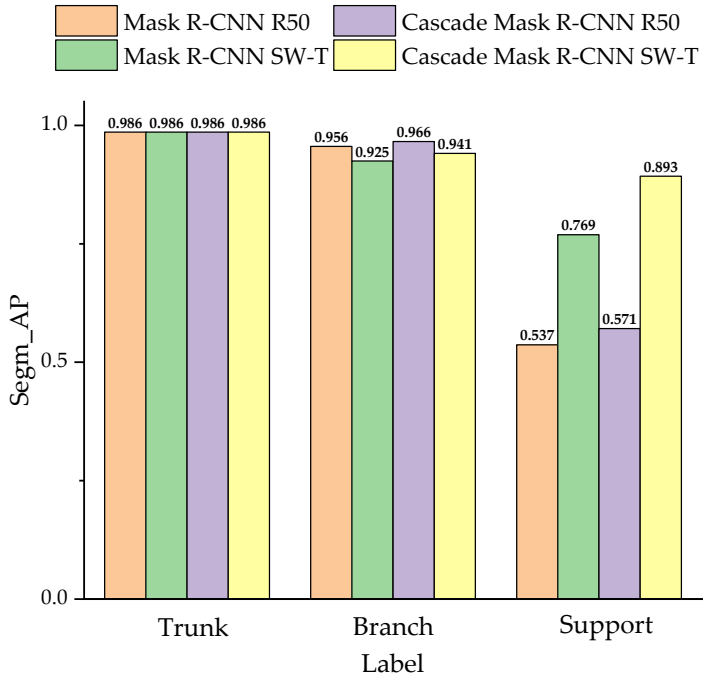
Figure 8. Loss mask graph of four algorithms.

When the IoU is 0.5, the AP values of the trunks, primary branches and supports under the four algorithms are shown in Figure 9. As can be seen from the figure, for the trunks and primary branches, the four algorithms have higher AP values and show better detection and segmentation effects. Among them, the AP difference between the trunks and the primary branches is small. In the identification of supports, Cascade Mask R-CNN performs better than Mask R-CNN. Compared with Mask R-CNN R50, the bbox AP and segm AP of Cascade Mask R-CNN R50, respectively, increase by 8.1% and 6.3%. Compared with Mask R-CNN SW-T, those of Cascade Mask R-CNN SW-T increase by 18.8% and 16.1%. The AP values of Swin-T are noticeably higher than Resnet50. Compared with Mask R-CNN R50, the bbox AP and segm AP of Mask R-CNN SW-T increase by 40.2% and 43.2%.

Compared with Cascade Mask R-CNN R50, those of Cascade Mask R-CNN SW-T increase by 53.9% and 56.4%. Cascade Mask R-CNN SW-T not only has the highest detection and segmentation accuracy for the supports, 0.879 and 0.893, respectively, but also achieves relatively good results for the identification of trunks and primary branches. The bbox AP and segm AP of the trunks are 0.986 and 0.986, and those of primary branches are 0.965 and 0.941.



(**a**)



(**b**)

Figure 9. AP values of each category when the IoU threshold is 0.5: (**a**) bbox AP of each category; (**b**) segm AP of each category.

3.2. Skeleton Extraction and Junction Point Location

Figure 10 shows the process of skeletonization and junction point location. The first method used Zhang & Suen to directly extract the skeleton of the binary image of the apple tree, and the result is shown in Figure 10f. Since the central axis is a single pixel, when $N(P1) \geq 3$, this point is the branch junction point. The junction point is represented by a blue point in the figure.

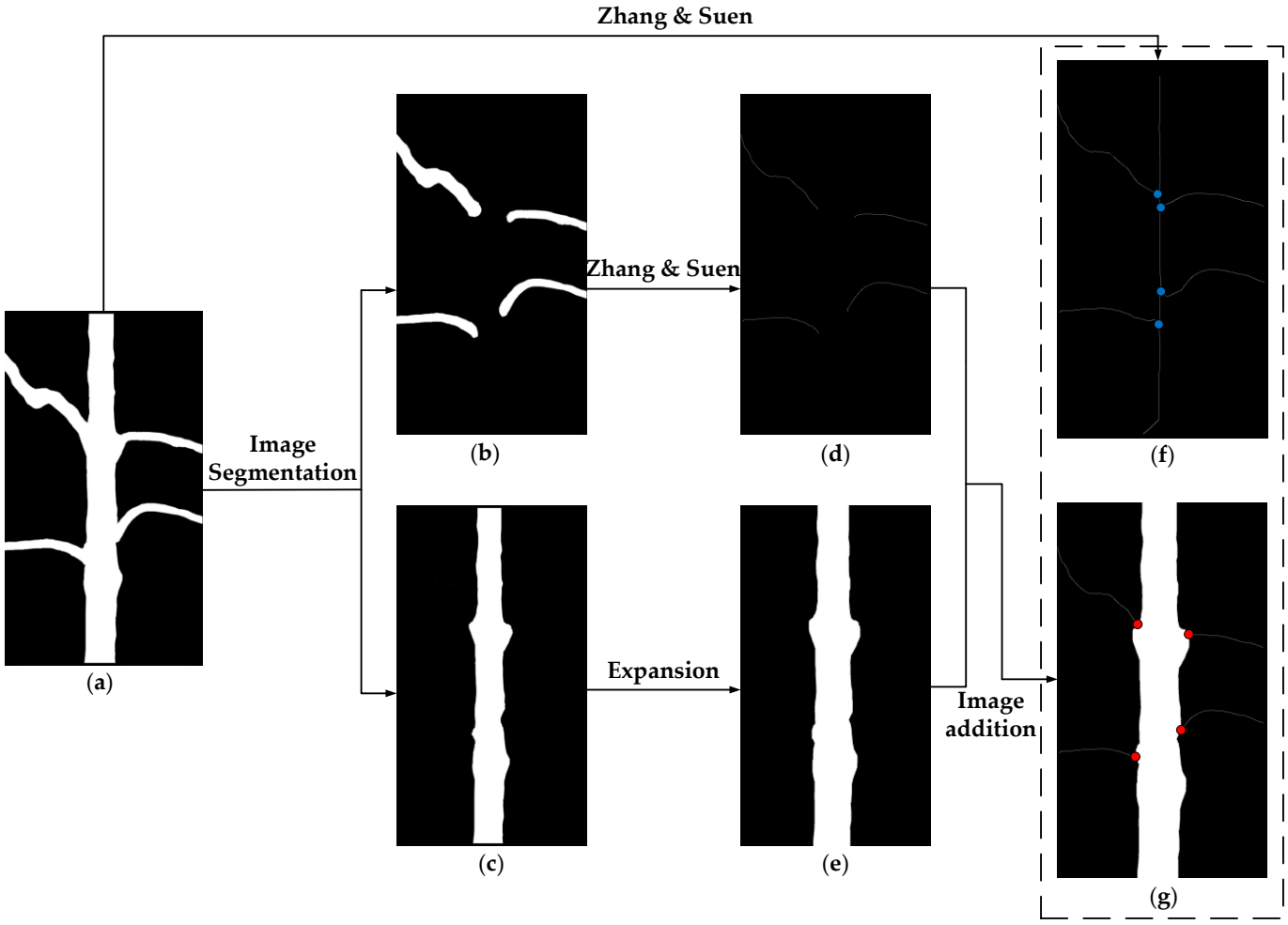


Figure 10. Process of skeletonization and junction point location: (**a**) binary image of the apple tree; (**b**) mask image of primary branch; (**c**) mask image of the trunk; (**d**) primary branch skeleton processed by Zhang & Suen; (**e**) expanded trunk mask image; (**f**) skeleton and junction points of apple tree directly processed by Zhang & Suen; (**g**) skeleton and junction points obtained by Cascade Mask R-CNN SW-T combined with Zhang & Suen.

The second method divided and extracted the trunk and primary branches according to the optimal algorithm Cascade Mask R-CNN SW-T discussed above. Then, they were thresholded through image preprocessing to obtain the mask images of primary branches (Figure 10b) and the trunk (Figure 10c). The mask image of the primary branch was processed by Zhang & Suen to extract the central axis of primary branch (Figure 10d). Since Zhang & Suen has a certain loss in the edge when thinning the primary branch, the mask of the trunk needs to be expanded (Figure 10e) to compensate for the loss value and prevent the failure of the central axis of the primary branch to connect with the trunk mask. Finally, the skeleton image was obtained through pixel addition performed between the central axis image of the primary branch and the expanded trunk mask image (Figure 10g). According to the characteristics of single pixel connection in the axis of the primary branch, the junction point was determined by combining with the generation process of the skeleton

image. The first step was to obtain the endpoint of the central axis of the primary branch from the central axis image of the primary branch (Figure 10d), that is, $N(P1) = 1$. Since there were only primary branches in the image, these endpoints include the junction points between the primary branches and the trunk. After the branch was connected to the trunk mask, its connection point was $N(P1) \geq 3$. However, the trunk was not a single-pixel axis, and there were also pixels with $N(P1) \geq 3$. Thus, it was infeasible to directly determine the point where $N(P1) \geq 3$ as the junction point. In the second step, it was necessary to determine the endpoint in the final skeleton image (Figure 10g), namely $N(P1) = 1$, excluding the same pixels detected twice, and the remaining pixels were the junction points between the trunk and the primary branches. The red dots in Figure 10g represent the junction points.

4. Discussion

4.1. Segmentation Results Analysis

When the IoU is 0.5, the bbox mAP and segm mAP of the four algorithms are presented in Table 2. As indicated in the above findings, the instance segmentation effect of Cascade Mask R-CNN is better than that of Mask R-CNN, and the segmentation and detection accuracy of Swin-T in the backbone network is significantly higher than that of Resnet50 according to Figure 7. The bbox mAP and segm mAP of the four algorithms from high to low are Cascade Mask R-CNN SW-T, Mask R-CNN SW-T, Cascade Mask R-CNN R50, and Mask R-CNN R50. Among them, the bbox mAP and segm mAP of Cascade Mask R-CNN SW-T both reach 0.940, which is more than 5% higher than that of Mask R-CNN SW-T. Compared with Cascade Mask R-CNN R50, it is more than 11% higher, which means that Swin-T can be used for apple branch detection with better results.

As shown in Figure 9, in the detection and segmentation of trunks, primary branches and supports, the difference between the four algorithms for trunks and primary branches is small. In the detection and segmentation of supports, the order of AP value of the four algorithms from high to low is the same as that of mAP, and the bbox AP and segm AP of Cascade Mask R-CNN SW-T for supports are higher than 0.870, which is more than 16% higher than other algorithms. In robotic pruning, supports can cause great interference to the pruning operation of the mechanical arm, so the identification of supports is of great significance. Therefore, this algorithm is more suitable for the detection of apple branches in robotic pruning. The above conclusions can be verified in the identification prediction diagram. The detection and segmentation prediction diagram of the four algorithms in different scenarios is shown in Figure 11. Green areas represent trunks, purple areas represent primary branches and pink areas represent supports. The targets (trunk, primary branch and support) are successfully segmented in the complex orchard environment. There is no significant difference in the detection effects of the four algorithms under backlighting and left-side lighting. Under the right-side lighting, Mask R-CNN R50 and Cascade Mask R-CNN R50 do not detect any support, Mask R-CNN SW-T detects part of the support, and Cascade Mask R-CNN SW-T detects the support more completely; under the frontlighting, Cascade Mask R-CNN SW-T similarly has the best detection effect on the support. There is much interference of branches in the right-side lighting image, and some of the supports are blocked by trunks in the frontlighting image. The complex environment of the orchard has a great impact on the identification accuracy of the supports. In addition, there are a small number of supports in the training sample, which leads to lower identification accuracy. This also suggests that light has no obvious effect on segmentation effect. It is verified again that, among the above four algorithms, Cascade Mask R-CNN SW-T has the best generalization ability and is more suitable for the detection and segmentation of apple branches.

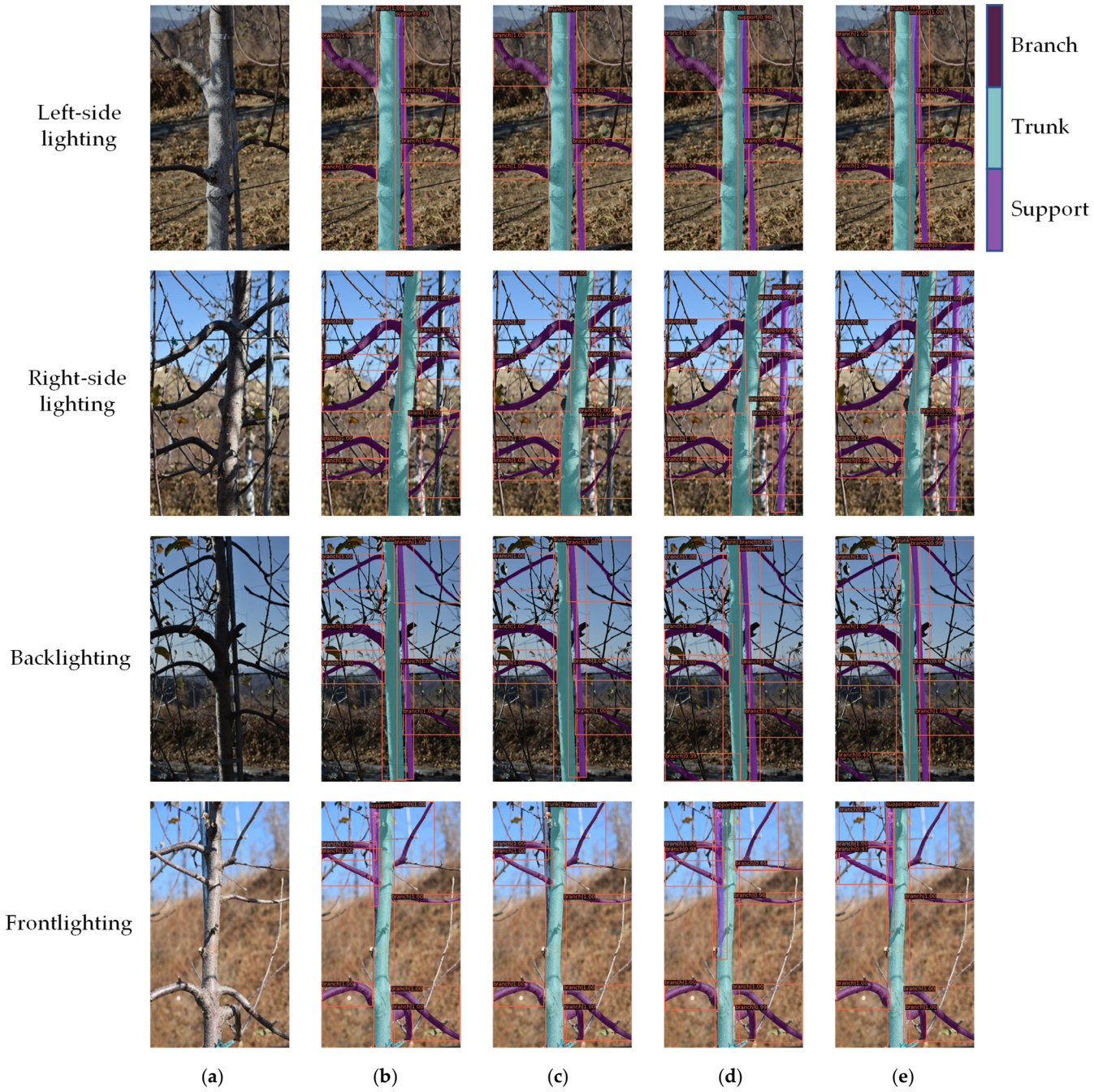
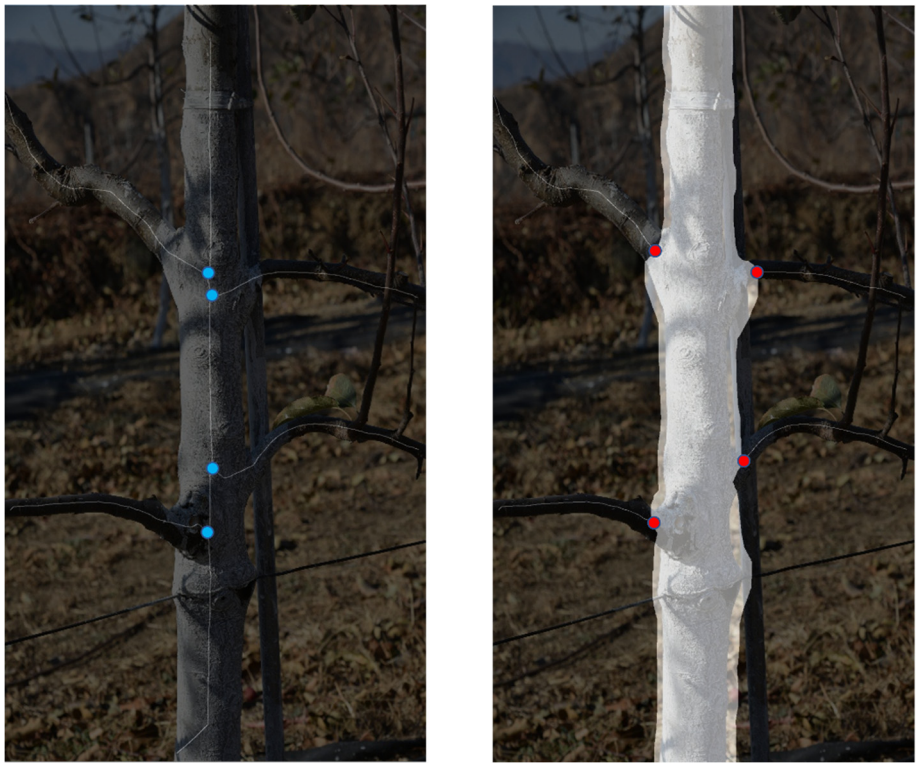


Figure 11. Prediction graphs of four algorithms for detection and segmentation in different scenarios: (**a**) input image; (**b**) identification result of Mask R-CNN R50; (**c**) identification result of Cascade Mask R-CNN R50; (**d**) identification result of Mask R-CNN SW-T; (**e**) identification result of Cascade Mask R-CNN SW-T.

4.2. Skeleton Extraction and Junction Point Location Results Analysis

The comparison between the results of the two methods and the actual apple tree are shown in Figure 12. Due to the influence of the trunk, the skeleton model obtained by directly extracting the skeleton is quite different from the actual apple tree shape, while the apple tree skeleton obtained by segmentation is more in line with the actual shape. In addition, the branch junction points all locate on the single-pixel trunk skeleton in Figure 12a, which is inconsistent with the real junction point positions. On the contrary, Figure 12b contains diameter information of the trunk, and the position of the junction

points is closer to the actual one. Therefore, in the following research, the apple tree skeleton and junction point information can be obtained by combining deep learning with skeleton extraction algorithm.



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

Figure 12. Comparison between two skeleton extraction methods and actual apple tree: (**a**) comparison of the skeleton and junction points of the apple tree directly processed by Zhang & Suen with those of actual apple tree; (**b**) comparison between skeleton and junction points obtained by Cascade Mask R-CNN SW-T combined with Zhang & Suen and those of the actual apple tree.

4.3. Comparison with Previous Studies

The premise of robotic pruning of fruit trees is to identify primary branches and locate junction points. In previous studies, researchers have conducted numerous related studies using their respective datasets and methods. In the research on branch identification, in order to improve the accuracy, researchers trained the Foreground-RGB images or Pseudo-Color Image and Depth of fruit trees [14,16,17], segmented the trunk, branches, etc., and determined the branch skeleton through curve fitting [14–16]. In this study, the Swin Transformer with shifted window is applied to the branch detection, and the RGB images of apple trees are directly trained, which greatly improved the identification accuracy, showed excellent generalization ability in complex orchard environments and reduced the preliminary work of image processing and the error of curve fitting. In the research on branch junction points, the researchers obtained the skeleton model of the crop through the skeleton extraction algorithm, and determined the skeleton intersection point between branches and trunk as the junction point [22]. In this study, the skeleton of the branches segmented by Cascade Mask R-CNN SW-T is extracted and the junction point with the trunk diameter information is located by adding it to the trunk image. On this basis, the branch diameter and the pruning point position obtained in the subsequent research can be more accurate. Furthermore, the next research direction is to obtain the diameter as well as spacing of the branches and determine the pruning points according to the pruning rules.

5. Conclusions

This study took the dormant high-spindle apple tree as the research object and adopted the method of deep learning to detect and segment the trunk, primary branch and support of apple trees, during which the detection results of the four algorithms were compared. Based on the prediction results of deep learning, the apple tree skeleton was extracted and the junction points were located, and, finally, the junction points between the primary branches and the trunk were obtained. To summarize, the following conclusions are drawn from this study:

(1) When the IoU was 0.5, the identification effect of Cascade Mask R-CNN was better than that of Mask R-CNN, and Swin-T was better than Resnet50. At the same time, the bbox mAP and segm mAP of Cascade Mask R-CNN SW-T were the highest, which were 0.943 and 0.940, respectively.

(2) In the detection and segmentation of each category, the four algorithms had a small difference in accuracy for the trunk and primary branch. In the detection of support, the accuracy of Cascade Mask R-CNN was higher than that of Mask R-CNN, and Swin-T was higher than Resnet50. Likewise, the same conclusion was obtained in the prediction results of the testing samples. Cascade Mask R-CNN SW-T was determined as the optimal algorithm. Its trunk bbox AP and segm AP were 0.986 and 0.986, primary branch were 0.965 and 0.941, and support were 0.879 and 0.893. The algorithm was more suitable for the detection of apple branches in robotic pruning. In addition, it was verified that lighting had no obvious effect on the detection effect of deep learning.

(3) Compared with the direct application of Zhang & Suen, combining Cascade Mask R-CNN SW-T with Zhang & Suen to extract the apple tree skeleton had the advantage that the obtained skeleton had the trunk diameter information and its shape and junction point position were closer to actual apple trees.

In this study, a high-accuracy apple branch segmentation model (Cascade Mask R-CNN SW-T) was obtained through deep learning. Additionally, the model was combined with the skeleton extraction algorithm (Zhang & Suen) to obtain more accurate branch junction points. In the future work, the branch diameter can be obtained by the distance between the branch edge and junction point, the spatial position relationship of the branches can be determined according to the distance between the branch junction points, and the intelligent pruning rules of apple trees can be formulated. According to the pruning rules, Cascade Mask R-CNN SW-T can be used to identify the branches to be pruned, locate the pruning point, use the depth camera for spatial positioning, and combine the robotic arm and the end-effector to form an intelligent pruning robot. However, the limitations of two-dimensional images can lead to problems such as mutual occlusion of apple branches being unable to be solved. Therefore, the subsequent research can also combine the advantages of 3D point clouds and images to develop a real-time and holistic algorithm for apple branch detection and pruning point identification. What is more, in the future, the research methods can be extended to fruit trees, such as grape vines and upright fruiting offshoot (UFO) cherry trees, to realize intelligent pruning of various standardized orchards.

Author Contributions: Conceptualization, F.K., S.T. and Y.Y.; methodology, F.K., S.T. and Y.Y.; software, S.T. and Y.Y.; formal analysis, S.T. and Y.Y.; investigation, F.K., S.T., Y.W. and Y.Y.; resources, F.K., W.L. and Y.W.; data curation, S.T. and Y.Y.; writing—original draft preparation, F.K. and S.T.; writing—review and editing, F.K. and S.T.; supervision, F.K., W.L., Y.W. and C.F.; project administration, F.K., W.L. and Y.W.; funding acquisition, F.K., W.L. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the NingXia key research and development program (Grant No. 2019BBF02009).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We thank the orchard in Fuping for providing the experimental site.

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

References

1. Kolmanič, S.; Strnad, D.; Kohek, Š.; Benes, B.; Hirst, P.; Žalik, B. An algorithm for automatic dormant tree pruning. *Appl. Soft Comput.* **2021**, *99*, 106931. [CrossRef]
2. Poni, S.; Tombesi, S.; Palliotti, A.; Ughini, V.; Gatti, M. Mechanical winter pruning of grapevine: Physiological bases and applications. *Sci. Hortic.* **2016**, *204*, 88–98. [CrossRef]
3. Martin-Gorriz, B.; Porras Castillo, I.; Torregrosa, A. Effect of mechanical pruning on the yield and quality of 'fortune' mandarins. *Span. J. Agric. Res.* **2014**, *12*, 952–959. [CrossRef]
4. Zahid, A.; Mahmud, M.S.; He, L.; Heinemann, P.; Choi, D.; Schupp, J. Technological advancements towards developing a robotic pruner for apple trees: A review. *Comput. Electron. Agric.* **2021**, *189*, 106383. [CrossRef]
5. Zheng, Y.; Jiang, S.; Chen, B.; Lü, H.; Wan, C.; Kang, F. Review on technology and equipment of mechanization in hilly orchard. *Trans. Chin. Soc. Agric.* **2020**, *51*, 1–20.
6. Lehnert, R. Robotic Pruning. Good Fruit Grower. 1 November 2012. Available online: https://www.goodfruit.com/robotic-pruning (accessed on 15 March 2022).
7. He, L.; Schupp, J. Sensing and automation in pruning of apple trees: A review. *Agronomy* **2018**, *8*, 211. [CrossRef]
8. Ji, W.; Qian, Z.; Xu, B.; Tao, Y.; Zhao, D.; Ding, S. Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot. *Optik* **2016**, *127*, 11173–11182. [CrossRef]
9. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines. *J. Field Rob.* **2017**, *34*, 1100–1122. [CrossRef]
10. Tabb, A.; Medeiros, H. Automatic segmentation of trees in dynamic outdoor environments. *Comput. Ind.* **2018**, *98*, 90–99. [CrossRef]
11. Karkee, M.; Adhikari, B.; Amatya, S.; Zhang, Q. Identification of pruning branches in tall spindle apple trees for automated pruning. *Comput. Electron. Agric.* **2014**, *103*, 127–135. [CrossRef]
12. Medeiros, H.; Kim, D.; Sun, J.; Seshadri, H.; Akbar, S.A.; Elfiky, N.M.; Park, J. Modeling dormant fruit trees for agricultural automation. *J. Field Rob.* **2017**, *34*, 1203–1224. [CrossRef]
13. Mack, J.; Lenz, C.; Teutrine, J.; Steinhage, V. High-precision 3D detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Comput. Electron. Agric.* **2017**, *135*, 300–311. [CrossRef]
14. Zhang, J.; He, L.; Karkee, M.; Zhang, Q.; Zhang, X.; Gao, Z. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). *Comput. Electron. Agric.* **2018**, *155*, 386–393. [CrossRef]
15. Zhang, J.; Karkee, M.; Zhang, Q.; Zhang, X.; Yaqoob, M.; Fu, L.; Wang, S. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Comput. Electron. Agric.* **2020**, *173*, 105384. [CrossRef]
16. Zhang, X.; Karkee, M.; Zhang, Q.; Whiting, M.D. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. *J. Field Rob.* **2020**, *38*, 476–493. [CrossRef]
17. 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]
18. Majeed, Y.; Karkee, M.; Zhang, Q.; Fu, L.; Whiting, M.D. Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks. *Comput. Electron. Agric.* **2020**, *171*, 105308. [CrossRef]
19. Majeed, Y.; Karkee, M.; Zhang, Q. Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards. *Comput. Electron. Agric.* **2020**, *176*, 105671. [CrossRef]
20. Chen, Z.; Ting, D.; Newbury, R.; Chen, C. Semantic segmentation for partially occluded apple trees based on deep learning. *Comput. Electron. Agric.* **2021**, *181*, 105952. [CrossRef]
21. Yang, C.; Xiong, L.; Wang, Z.; Wang, Y.; Shi, G.; Kuremot, T.; Zhao, W.; Yang, Y. Integrated detection of citrus fruits and branches using a convolutional neural network. *Comput. Electron. Agric.* **2020**, *174*, 105469. [CrossRef]
22. Cuevas-Velasquez, H.; Gallego, A.; Fisher, R. Segmentation and 3d reconstruction of rose plants from stereoscopic images. *Comput. Electron. Agric.* **2020**, *171*, 105296. [CrossRef]
23. 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]
24. Ma, B.; Du, J.; Wang, L.; Jiang, H.; Zhou, M. Automatic branch detection of jujube trees based on 3D reconstruction for dormant pruning using the deep learning-based method. *Comput. Electron. Agric.* **2021**, *190*, 106484. [CrossRef]
25. You, A.; Grimm, C.; Silwal, A.; Davidson, J.R. Semantics-guided skeletonization of upright fruiting offshoot trees for robotic pruning. *Comput. Electron. Agric.* **2022**, *192*, 106622. [CrossRef]
26. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1483–1498. [CrossRef]

27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2021**, arXiv:2010.11929v2.
28. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
29. Zheng, H.; Wang, G.; Li, X. Swin-MLP: A strawberry appearance quality identification method by Swin Transformer and multi-layer perceptron. *J. Food Meas. Charact.* **2022**, *16*, 2789–2800. [CrossRef]
30. Wang, J.; Zhang, Z.; Luo, L.; Zhu, W.; Chen, J.; Wang, W. SwinGD: A Robust Grape Bunch Detection Model Based on Swin Transformer in Complex Vineyard Environment. *Horticulturae* **2021**, *7*, 492. [CrossRef]
31. Yuan, W.; Xu, W. MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer. *Remote Sens.* **2021**, *13*, 4743. [CrossRef]
32. Xu, X.; Feng, Z.; Cao, C.; Li, M.; Wu, J.; Wu, Z.; Shang, Y.; Ye, S. An Improved Swin Transformer-Based Model for Remote Sensing Object Detection and Instance Segmentation. *Remote Sens.* **2021**, *13*, 4779. [CrossRef]
33. Wang, Z.; Zhao, J.; Zhang, R.; Li, Z.; Lin, Q.; Wang, X. UATNet: U-Shape Attention-Based Transformer Net for Meteorological Satellite Cloud Recognition. *Remote Sens.* **2022**, *14*, 104. [CrossRef]
34. Xiao, X.; Guo, W.; Chen, R.; Hui, Y.; Wang, J.; Zhao, H. A Swin Transformer-Based Encoding Booster Integrated in U-Shaped Network for Building Extraction. *Remote Sens.* **2022**, *14*, 2611. [CrossRef]
35. Xu, Z.; Zhang, W.; Zhang, T.; Yang, Z.; Li, J. Efficient Transformer for Remote Sensing Image Segmentation. *Remote Sens.* **2021**, *13*, 3585. [CrossRef]
36. Xia, R.; Chen, J.; Huang, Z.; Wan, H.; Wu, B.; Sun, L.; Yao, B.; Xiang, H.; Xing, M. CRTransSar: A Visual Transformer Based on Contextual Joint Representation Learning for SAR Ship Detection. *Remote Sens.* **2022**, *14*, 1488. [CrossRef]
37. Liu, P.; Song, Y.; Chai, M.; Han, Z.; Zhang, Y. Swin–UNet++: A Nested Swin Transformer Architecture for Location Identification and Morphology Segmentation of Dimples on 2.25Cr1Mo0.25V Fractured Surface. *Materials* **2021**, *14*, 7504. [CrossRef]
38. Gao, L.; Zhang, J.; Yang, C.; Zhou, Y. Cas-VSwin transformer: A variant swin transformer for surface-defect detection. *Comput. Ind.* **2022**, *140*, 103689. [CrossRef]
39. Liao, Z.; Fan, N.; Xu, K. Swin Transformer Assisted PriorAttention Network for Medical Image Segmentation. *Appl. Sci.* **2022**, *12*, 4735. [CrossRef]
40. Jiang, Y.; Zhang, Y.; Lin, X.; Dong, J.; Cheng, T.; Liang, J. SwinBTS: A Method for 3D Multimodal Brain Tumor Segmentation Using Swin Transformer. *Brain Sci.* **2022**, *12*, 797. [CrossRef]
41. Schupp, J.; Winzeler, H.; Kon, T.; Marini, R.; Baugher, T.; Kime, L.; Schupp, M. A method for quantifying whole-tree pruning severity in mature tall spindle apple plantings. *HortScience Horts.* **2017**, *52*, 1233–1240. [CrossRef]
42. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 386–397. [CrossRef] [PubMed]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision (ECCV ′16), Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645. [CrossRef]
44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 12 June 2017; pp. 5998–6008.
45. Zhang, T.; Suen, C. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* **1984**, *27*, 236–239. [CrossRef]