agronomy

MDPI

Article

A Novel Fusion Perception Algorithm of Tree Branch/Trunk and Apple for Harvesting Robot Based on Improved YOLOv8s

Bin Yan [1,2,3,*], Yang Liu [1,3] and Wenhui Yan [4]

[1] College of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China
[2] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China
[3] Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an 710048, China
[4] College of Mechanical Engineering, Xi'an Shiyou University, Xi'an 710065, China
* Correspondence: yanbin@nwafu.edu.cn

Abstract: Aiming to accurately identify apple targets and achieve segmentation and the extraction of branch and trunk areas of apple trees, providing visual guidance for a picking robot to actively adjust its posture to avoid branch trunks for obstacle avoidance fruit picking, the spindle-shaped fruit trees, which are widely planted in standard modern apple orchards, were focused on, and an algorithm for apple tree fruit detection and branch segmentation for picking robots was proposed based on an improved YOLOv8s model design. Firstly, image data of spindle-shaped fruit trees in modern apple orchards were collected, and annotations of object detection and pixel-level segmentation were conducted on the data. Training set data were then augmented to improve the generalization performance of the apple detection and branch segmentation algorithm. Secondly, the original YOLOv8s network architecture's design was improved by embedding the SE module visual attention mechanism after the C2f module of the YOLOv8s Backbone network architecture. Finally, the dynamic snake convolution module was embedded into the Neck structure of the YOLOv8s network architecture to better extract feature information of different apple targets and tree branches. The experimental results showed that the proposed improved algorithm can effectively recognize apple targets in images and segment tree branches and trunks. For apple recognition, the precision was 99.6%, the recall was 96.8%, and the mAP value was 98.3%. The mAP value for branch and trunk segmentation was 81.6%. The proposed improved YOLOv8s algorithm design was compared with the original YOLOv8s, YOLOv8n, and YOLOv5s algorithms for the recognition of apple targets and segmentation of tree branches and trunks on test set images. The experimental results showed that compared with the other three algorithms, the proposed algorithm increased the mAP for apple recognition by 1.5%, 2.3%, and 6%, respectively. The mAP for tree branch and trunk segmentation was increased by 3.7%, 15.4%, and 24.4%, respectively. The proposed detection and segmentation algorithm for apple tree fruits, branches, and trunks is of great significance for ensuring the success rate of robot harvesting, which can provide technical support for the development of an intelligent apple harvesting robot.

Keywords: agronomy; digital agriculture; picking robot; apple tree; YOLOv8; object detection; semantic segmentation

Citation: Yan, B.; Liu, Y.; Yan, W. A Novel Fusion Perception Algorithm of Tree Branch/Trunk and Apple for Harvesting Robot Based on Improved YOLOv8s. *Agronomy* **2024**, *14*, 1895. https://doi.org/10.3390/agronomy14091895

Academic Editors: Spyros Fountas, Georgios Leontidis and Borja Espejo-García

Received: 27 July 2024
Revised: 15 August 2024
Accepted: 22 August 2024
Published: 24 August 2024

1. Introduction

Apple is a short maturity period fruit with seasonal harvesting characteristics. If ripe apples are not harvested in a timely manner, it will result in fruit decay and affect the apple yield, directly affecting the subsequent stages of fruit storage, transportation, processing, and sales. With the increase in apple yields and a shorter maturity period, the problem of fruit picking gradually becomes more prominent. Apple orchards can mainly be divided into two categories based on the planting mode of the fruit trees: traditional apple

orchards and modern apple orchards (also known as standard apple orchards). Traditional apple orchards have intersecting branches between fruit trees with low and enclosed rows, indicating poor mechanical passability in the orchard, making them unsuitable for robot picking operations [1,2]. In the planting modes of fruit trees of modern apple orchards, the dwarf rootstock dense planting mode is widely utilized. It has multiple advantages, such as a short tree crown, convenient fruit tree management, early fruit bearing, high apple yield, good quality, and convenient intelligent operation for robots, for which it is easy to achieve standardization and scale management, and is widely adopted by advanced apple production countries in the world. Dwarf rootstock dense planting mode is also the cultivation choice of modern apple industry developments. Among them, the spindle-shaped cultivation model has more fruiting branches with more advantages in terms of fruit yield and is widely used in standard modern orchards. Spindle-shaped apple trees in modern apple orchards are shown in Figure 1.

Figure 1. Spindle-shaped apple trees in modern apple orchards.

Nowadays, the harvest of apples still mainly relies on manual picking, which is the most time-consuming and labor-intensive key part of the production process. Robot fruit picking has become a global hot topic in research and application. However, since the relatively complex distribution of branches within the crowns of modern apple trees in orchards, branches will become potential obstacles in the movement path of harvesting robots. On the other hand, there are also situations where apples are obstructed by branches. If a harvesting robot directly pick apples without precise perception of the situations the robot may be faced with in orchards, which are described above, it may cause damage to the fruits and trees, as well as damage to the picking manipulator or arm of the robot, resulting in harvesting failure. Therefore, in-depth research on intelligent perception algorithms for apple tree fruits and branches based on artificial intelligence and deep learning is of great significance for ensuring the success rate of robot harvesting.

Up until now, many researchers have conducted research on the intelligent perception of apple and tree branch/trunk targets in complex orchard environments based on artificial intelligence algorithms. In terms of target perception of apples and fruit trees based on deep learning algorithms, relevant research from recent years is shown in Table 1.

Throughout the current research status of apple and fruit tree target perception based on deep learning algorithms, most existing algorithms only recognize and detect fruit targets on apple trees without integrating segmentation perception for apple tree branches and trunks. Therefore, directly applying these algorithms to the visual perception system of apple harvesting robots may lead to significant security risks. Thus, the spindle-shaped

fruit trees widely planted in standard modern apple orchards were utilized as the research object, and an intelligent perception algorithm design for apple tree fruit detection and branch segmentation for picking robots was improved based on the YOLOv8s detection and segmentation algorithm. The study provides theoretical guidance and technical reference for improving the success rate of apple harvesting robots, which is of great significance for intelligent harvesting in the apple industry.

Table 1. Target perception of apple tree fruits and branches based on deep learning.

Perception Model	Year	Achieve Apple Detection and Branch/Trunk Segmentation Simultaneously	Reference
Improved YOLOv7-RSES	2024	N	[3]
Improved YOLOv5s	2024	N	[4]
Improved YOLOv7	2024	N	[5]
Improved YOLOv3	2020	N	[6]
Improved YOLOv3	2021	N	[7]
CA-YOLOv4	2022	N	[8]
Improved YOLOv5s	2021	N	[2]
DaSNet	2019	Y	[9]
DaSNet-v2	2020	Y	[10]
Faster R-CNN (VGG16)	2020	N	[11]
Faster R-CNN (VGG16)	2020	N	[12]
Improved FCOS	2021	N	[13]
YOLOv4-tiny	2021	N	[14]
Improved FCOS	2022	N	[15]
Improved RetinaNet	2022	N	[16]
Improved YOLOv4	2022	N	[17]
Lad-YXNet	2022	N	[18]
Improved Centernet	2022	N	[19]
Improved YOLOv5m	2022	N	[1]
YOLOv4-SENL	2021	N	[20]
YOLOv7-CEA	2023	N	[21]

Y: yes. N: no.

2. Materials and Methods

2.1. Acquisition and Preprocessing of Image Data

2.1.1. Image Acquisition Method

In the study, fruits on Fuji apple trees of dwarf rootstock dense planting mode in a standard modern orchard were used as the research object, and original images of the apple trees in the standardized orchard at the Agricultural Science and Technology Experimental Demonstration Base of Qian County in Shaanxi Province and the Apple Experimental Station of Northwest A&F University in Baishui County of Shaanxi Province were collected. In the dwarf rootstock dense planting cultivation mode, the row spacing of apple trees is about 4 m. The plant spacing is about 1.2 m, and the tree height is about 3.5 m, which is suitable for an apple picking robot to operate in the orchard. The images of the apple trees were obtained on sunny and cloudy days. The images were captured using a Canon Powershot G16 camera (Canon, Tokyo, Japan), with a variety of angles selected for image acquisition at different shooting distances (0.5–1.5 m), and in total, 436 apple images were obtained, including those with the following conditions: apples occluded by leaves, apples occluded by branches, mixed occlusion, overlap between apples, natural light angle, backlight angle, sidelight angle, etc. (Figure 2). Example images of the branches and trunks of spindle-shaped fruit trees are shown in Figure 3. The resolution of the captured images is 4000 × 3000 pixels, and the format is JPEG.

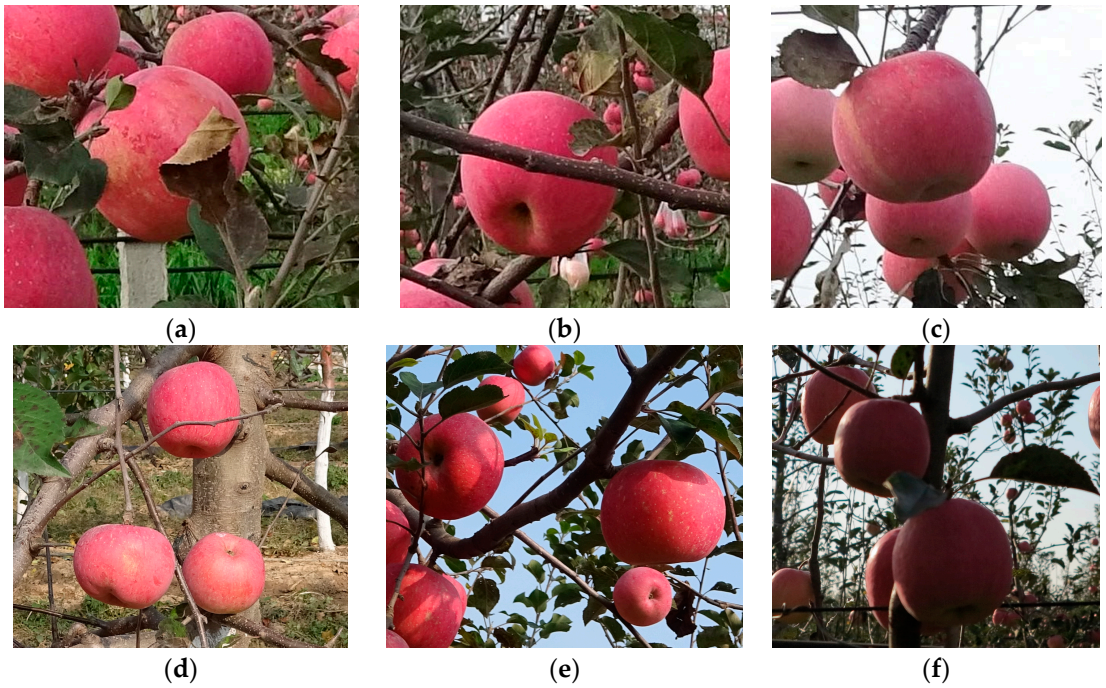


Figure 2. Apple images in different conditions. (**a**) Apples occluded by leaves. (**b**) Apples occluded by branches. (**c**) Overlapped apples. (**d**) Frontlight angle. (**e**) Sidelight angle. (**f**) Backlight angle.

Figure 3. Branches and trunks of spindle-shaped fruit trees.

2.1.2. Preprocessing of Image Data

The main task of preparing a model training dataset is to annotate images. Based on an in-depth analysis of the apple tree fruit and branch perception algorithm for an apple picking robot, as apple fruits are the ultimate target of the picking robot's operation, the apple targets were labeled. On the other hand, since the branches of the fruit tree may block the apples to be picked, and a picking robot needs to avoid branch obstacles to ensure the safety of the picking process, otherwise, it is highly likely to damage the fruit, branch structure, robot's picking hand, etc., leading to robot picking failure, pixel-level segmentation and the labeling of apple tree branches and trunks are necessary.

In the study, Labelme software (version: 5.0.0) was used to annotate images. Labelme is an image annotation tool developed by MIT's (Massachusetts Institute of Technology, MIT) Computer Science and Artificial Intelligence Laboratory (CSAIL). People can use this tool to create customized annotation tasks or perform image annotation. The project source code is already open source (official website: https://github.com/wkentaro/labelme, accessed on 27 July 2024). In the study, Labelme software was used for image annotation. This software can annotate not only fruits but also perform pixel-level image segmentation. For apple targets, the 'Create Rectangle' function in the Labelme annotation interface was utilized. When annotating apples, it is important to note that the four edges of the annotated detection box must be tangent to the edges of apple to ensure that there is not

too much background information in the detection box, thereby ensuring the quality of the dataset and the detection accuracy of the perception model.

The 'Create Linegrip' function in Labelme software was used to label apple tree branches and trunks by attaching label lines along the direction of branches within the area of fruit tree branches and trunks. The annotation process is shown in Figure 4. The branches and trunks are marked with deep red line segments, and apple targets are marked with green rectangular boxes.

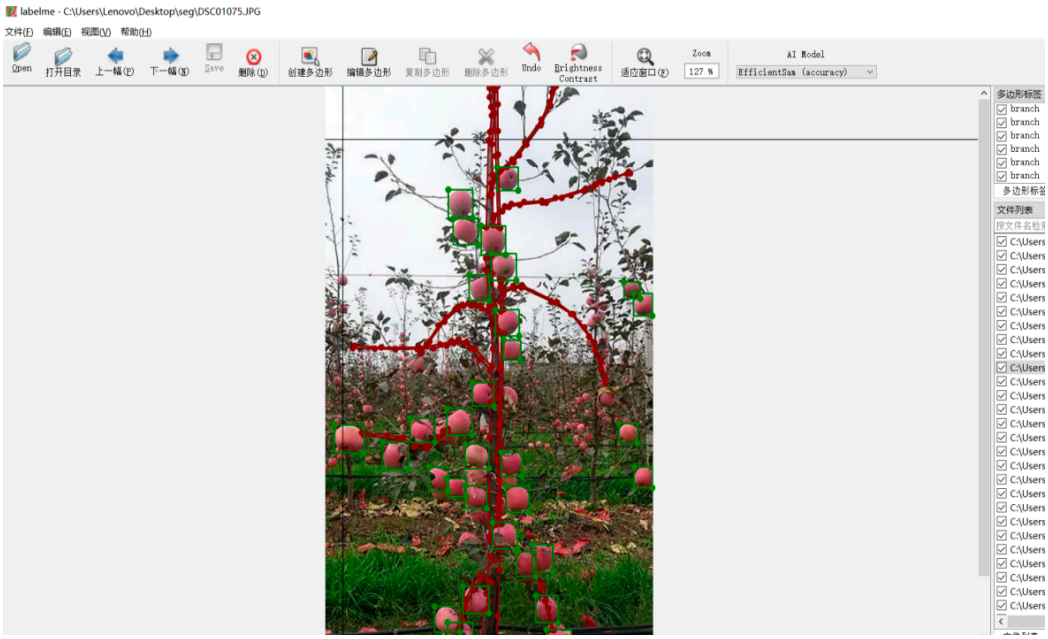


Figure 4. Example of annotation for apples and branches/trunks.

The JSON files typically contain the following information: the name of the image data, the label name (category name) of the target object in the image, the bounding box coordinates of the target object, the semantic segmentation label name, and the annotation style of the target object.

A total of 400 images with stable shooting quality were randomly selected from the collected images for annotation, and corresponding JSON files were generated. Then, the JSON files were converted into TXT files. Due to uncertain factors such as lighting and weather, the visual perception environment of the robots during recognition and picking operations in orchards is very complex. In order to enrich the image data of the training set, improve the generalization ability of the apple recognition and branch/trunk segmentation model, and better extract the characteristics of apples and tree branches/trunks, data augmentation processing was performed on the images of the training set.

Before expanding the data, the dataset was partitioned. Based on the existing 400 samples with stable shooting quality, 340 images were divided into the training set, 30 images were used as the validation set, and the other 30 images were used as the test set. The training data include a total of 8218 apple fruit target labels and 7160 branch/trunk labels.

The training set images were amplified using data augmentation techniques such as image rotation technology, brightness enhancement and reduction technology, contrast enhancement and reduction technology, and adding Gaussian noise in the image technology. A total of 4760 images obtained after data augmentation were utilized as training set data for the subsequent training of the apple recognition and branch/trunk segmentation model. On the other hand, in the process of data augmentation of images, it is also necessary to process the label files containing labeled information corresponding to the image to ensure that the position information of the labeled target object is correct.

2.2. Network Architecture and the Improved Design of YOLOv8s

2.2.1. YOLOv8s Network Architecture

The YOLOv8 network architecture has the advantages of high detection and segmentation accuracy and fast running speed [22–32]. On the other hand, the weight of this network model is relatively small, making the YOLOv8 model suitable for deployment on embedded devices to achieve real-time detection and segmentation of targets. YOLOv8 model was built on the historical version of the YOLO series, introducing new features and improvement points to further enhance the performance and flexibility, making it a priority choice for achieving a series of tasks such as object detection and image segmentation. The YOLOv8 network contains five submodels based on differences in the model size: YOLOv8n, v8s, v8m, v8l, and v8x (detailed parameters are shown in Table 2). Due to the fact that the detection and segmentation accuracy, real-time performance, and lightweighting of the model are directly related to the accuracy and efficiency of robotic recognition of fruits and the segmentation of branches and trunks, factors such as the perception accuracy, model volume, and detection speed were comprehensively considered in the study. An intelligent perception network that integrates fruit detection and branch segmentation for apple picking robots was improved and designed based on the YOLOv8s architecture.

Table 2. Model parameters of the five YOLOv8 architectures.

Model	Depth	Width	Layer	Parameters	Size (MB)
YOLOv8n	0.33	0.25	225	3.16×10^6	6.23
YOLOv8s	0.33	0.5	225	1.12×10^7	21.54
YOLOv8m	0.67	0.75	295	2.59×10^7	49.7
YOLOv8l	1	1	365	4.37×10^7	83.73
YOLOv8x	1	1.25	365	6.82×10^7	130.5

The YOLOv8 network architecture includes object detection networks of different resolutions and instance segmentation networks based on YOLACT. The specific YOLOv8 structure is shown in Figure 5:

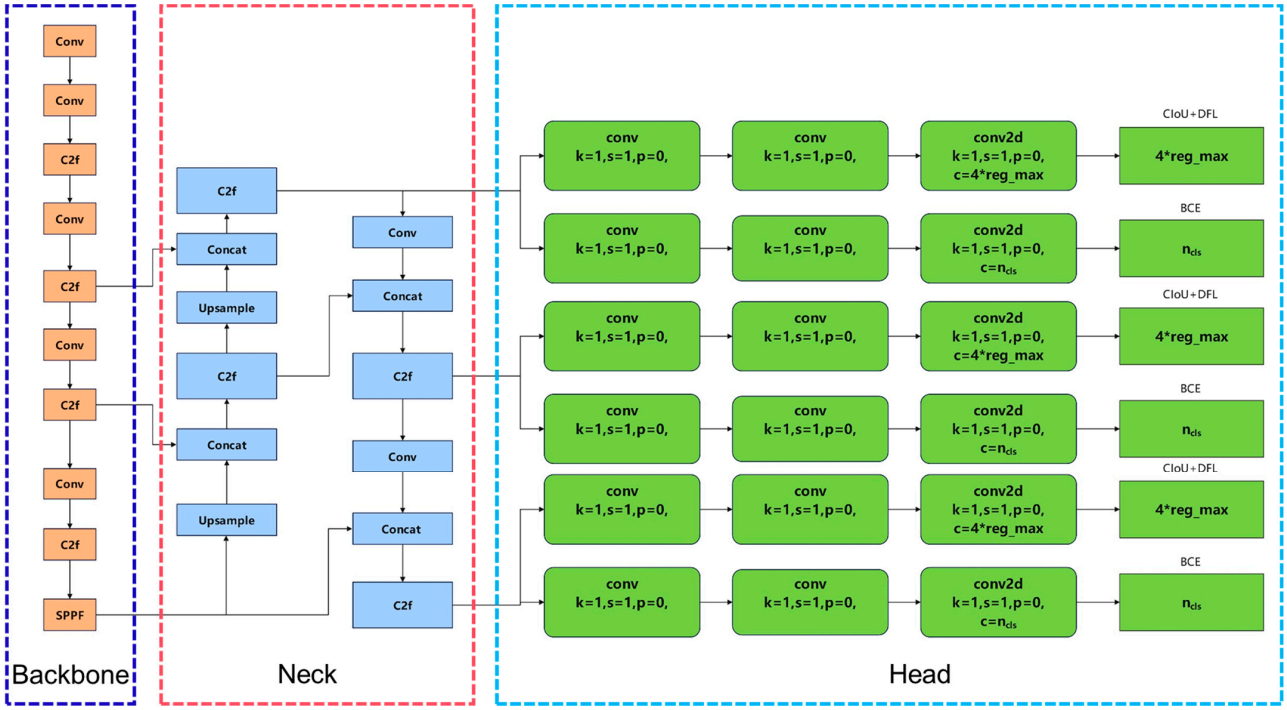


Figure 5. YOLOv8 network architecture.

The YOLOv8 network structure mainly consists of three parts: Backbone, Neck, and Head architectures. The Backbone is the main component of the model and draws inspiration from the CSP module and extracts features separately. The input data are sent to part 1 and part 2, where part 2 performs convolution operations and the C layer is obtained. Then, part 1 and the C layers are connected. In YOLOv8, the C3 module is replaced with the C2f module on the basis of YOLOv5, further achieving a lightweight model architecture, while continuing to use the SPPF module of YOLOv5. The schematic diagrams of the C3 module and C2f module are shown in Figure 6:

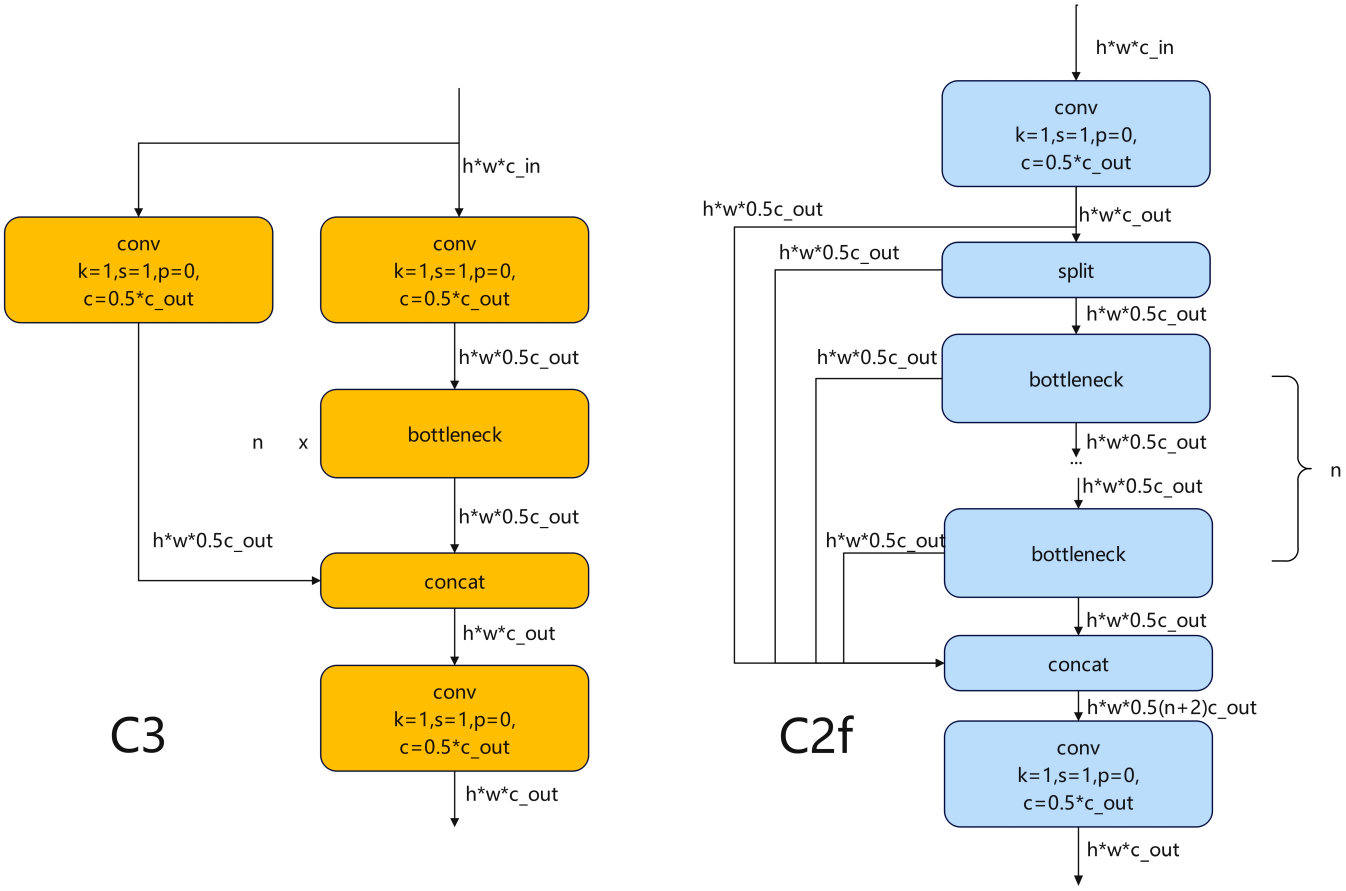


Figure 6. Architectures of C3 module and C2f module.

The resolution of the input Backbone structure image is 640 × 640. After passing through Backbone layers, including Layer 5, Layer 7, and Layer 10, the resolutions of the output feature maps are 80 × 80, 40 × 40, and 20 × 20, respectively. Furthermore, three types of feature maps are input into the Neck structure, which is a bidirectional network that introduces a bottom–up feature extraction method, making it easier for low-level information to be transmitted to the top layer. After the feature maps of Layer 5, Layer 7, and Layer 10 are inputted into the Neck structure, feature maps are upsampled and processed through channel fusion. Finally, the output feature maps of three branches in the Neck structure are sent to the Head layer for apple detection and branch/trunk segmentation.

In the Head structure of YOLOv8, the regression branch and prediction branch are separated, which effectively reduces the number of parameters and computational complexity while enhancing the model's generalization ability and robustness. YOLOv8 adopts an object detection method that does not require anchor-free nodes [33], which mainly represents objects through multiple key points or center points and boundary information, making it more suitable for the detection of small targets. YOLOv8 has achieved significant

performance improvements in the field of object detection while maintaining an efficient balance between speed and accuracy.

In the YOLOv8 network, image segmentation is implemented based on the YOLACT segmentation network. A mask prototype image of the entire image will be outputted by the segmentation branch of the YOLACT-based network. The process of generating a mask prototype image is as follows: Firstly, the feature map is inputted, which has a high resolution and can better preserve spatial detail information while fusing certain semantic information. Then, the feature map is upsampled, and the number of channels in the feature map is adjusted to k through a convolutional layer with a 1×1 kernel, forming a prototype image of k masks. For each target object, the confidence levels of the k masks are multiplied by the prototype graph of the k masks; then, all the results are added up, and a sigmoid nonlinear function is applied to generate the final mask, obtaining the segmentation result of the target object. The mathematical expression for segmentation result M is as follows:

$$M = \sigma(PC^T) \tag{1}$$

Among them, P is k prototype masks with a dimension of $h \times w \times k$, C is k mask coefficients with a dimension of $n \times k$ (n is the predicted target object number in the detection branch), and σ is the sigmoid function.

2.2.2. Inserted Design of SE Module in Backbone Network

Due to the uneven thickness of apple tree branches and trunks, accurate segmentation of apple tree branches requires an improved design of the original YOLOv8 network. Firstly, the SE (sequence and networks, SEnet) attention mechanism module was embedded [10,34] in the Backbone network.

The SE module is a kind of visual attention mechanism architecture wherein a novel feature recalibration strategy, illustrating the significance of each feature channel, is automatically obtained through learning, and then useful features are promoted, while unessential features are suppressed accordingly. Since the computation of the SE module is small, and it can effectively improve the expression ability of the model and optimize the content learned, it was inserted in the Backbone of the improved YOLOv8s network design in the study to improve the detection accuracy of the model. The schematic diagram of the SE module structure is shown in Figure 7 to provide a clearer understanding of the mechanism and the process of feature extraction from images using this module.

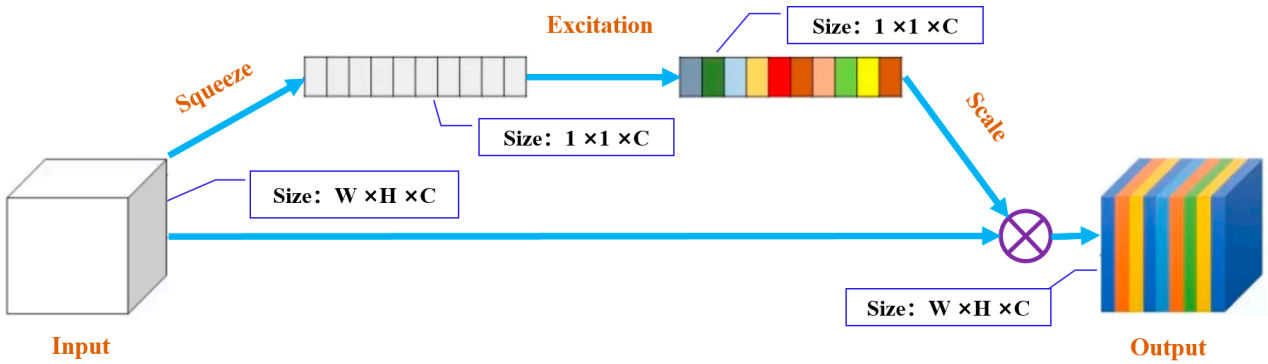


Figure 7. Architecture of SE module.

2.2.3. Inserted Design of Dynamic Snake Convolution Module

The dynamic snake convolution module [32,35,36] can better perceive small local structures and complex global shapes (such as blood vessels). This structure has a good performance in tubular segmentation tasks. The shape of the convolution kernel can be modified based on input features, allowing the network to adapt to the shape of an object rather than being limited to a fixed kernel shape. The dynamic snake convolution module has a good performance in accurate recognition and adaptability for various complex target

shapes, thereby enhancing the detection performance of the network. Therefore, dynamic snake convolution modules were embedded in the improved network to promote the segmentation effect of tree branches and trunks.

Dynamic snake convolution adopts an iterative strategy (as shown in Figure 8) to sequentially select the next position of each target to be processed for observation, ensuring continuity of the focus and avoiding excessive diffusion of the perception range due to large deformation offsets.

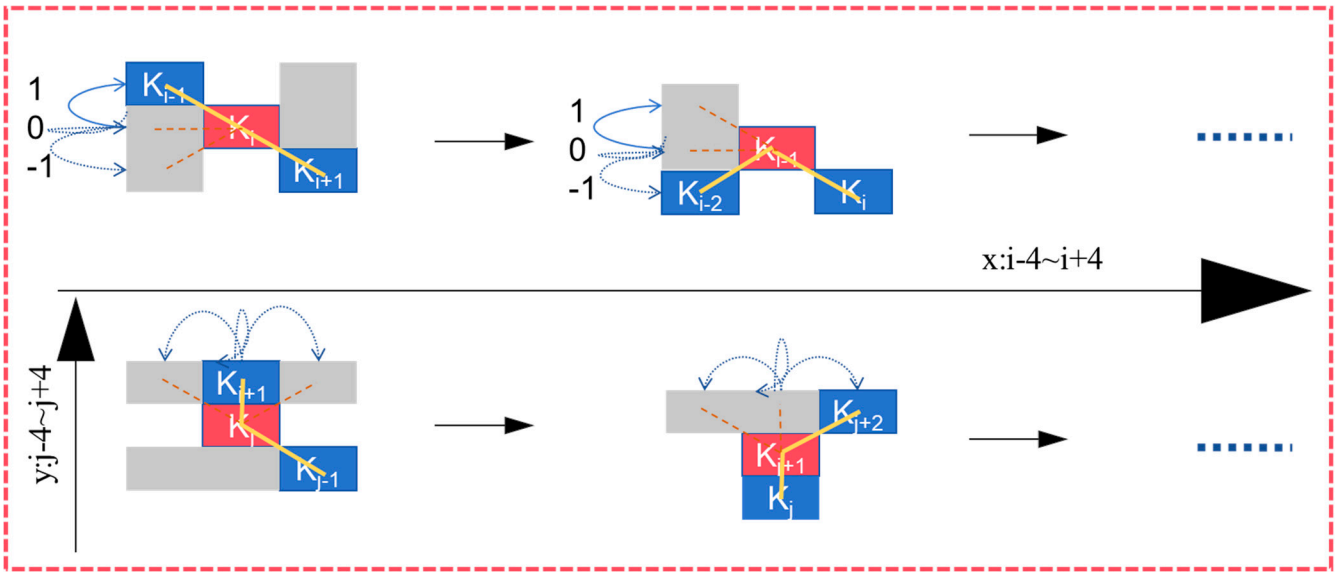


Figure 8. Schematic diagram for iterative method of dynamic snake convolution on *x*-axis and *y*-axis.

According to Figure 8, it can be seen that the design of the snake convolution kernel linearizes the standard convolution kernel on the *x*- and *y*-axes. Taking *x*-axis direction as an example, the specific position of each grid can be represented as follows:

$$K_{\mathrm{i}} \pm c = (X_{\mathrm{i}} \pm c, Y_i \pm c) \tag{2}$$

where c can be set as 0, 1, 2, 3, and 4, respectively, representing the horizontal distance from the center position of the network. The selection of each grid position $K_i \pm c$ in convolutional kernel K is a cumulative process. Starting from central position K_i, the position away from the central grid depends on the position of the previous grid, which is K_{i+1}, and increases the offset $\Delta = \{\delta | \delta \in [-1, 1]\}$ relative to K_i. Therefore, the offset needs to be accumulated to ensure that the convolution kernel conforms to a linear morphological structure.

Due to the variation in the two-dimensional directions (*x*-axis and *y*-axis), the dynamic snake convolution kernel covers a selectable range of 9×9 receptive fields during the deformation process, as shown in Figure 9.

According to Figure 9, it can be seen that each convolution position of dynamic snake convolution is based on its previous position, and the swing direction is freely selected, thereby ensuring the continuity of the model's perception while freely selecting the next position.

In summary, the differences between the improved YOLOv8s network architecture and the original network are as follows: firstly, the SE attention mechanism module was added to the original Backbone network; secondly, dynamic snake convolution was embedded into the C2f module of the original YOLOv8s network Head architecture. The schematic diagram of the improved YOLOv8 network architecture is shown in Figure 10, where the image to be perceived is input from the first Conv module in upper left corner of the figure.

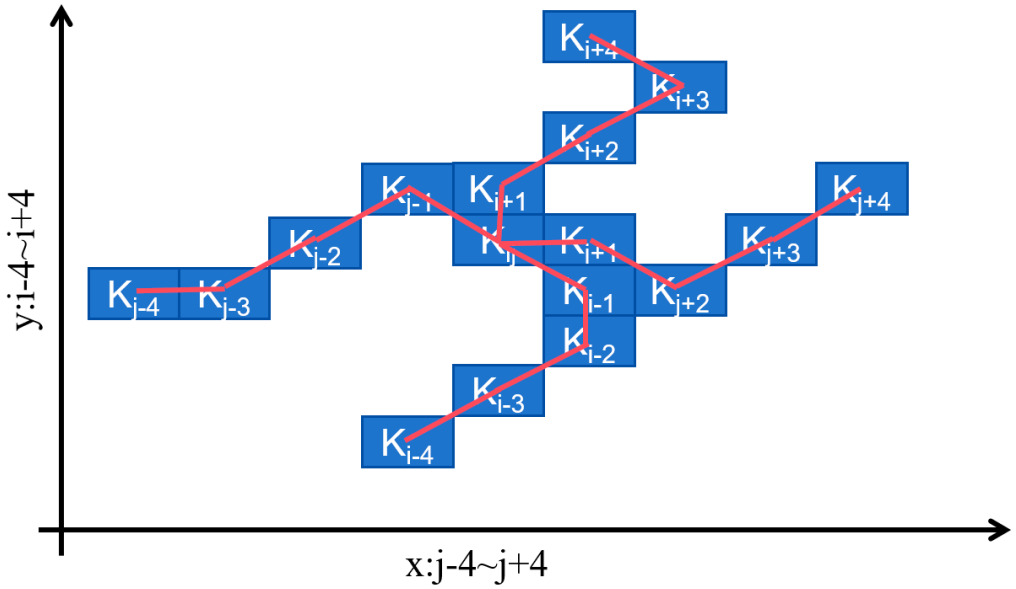


Figure 9. Perception view of dynamic snake convolution.

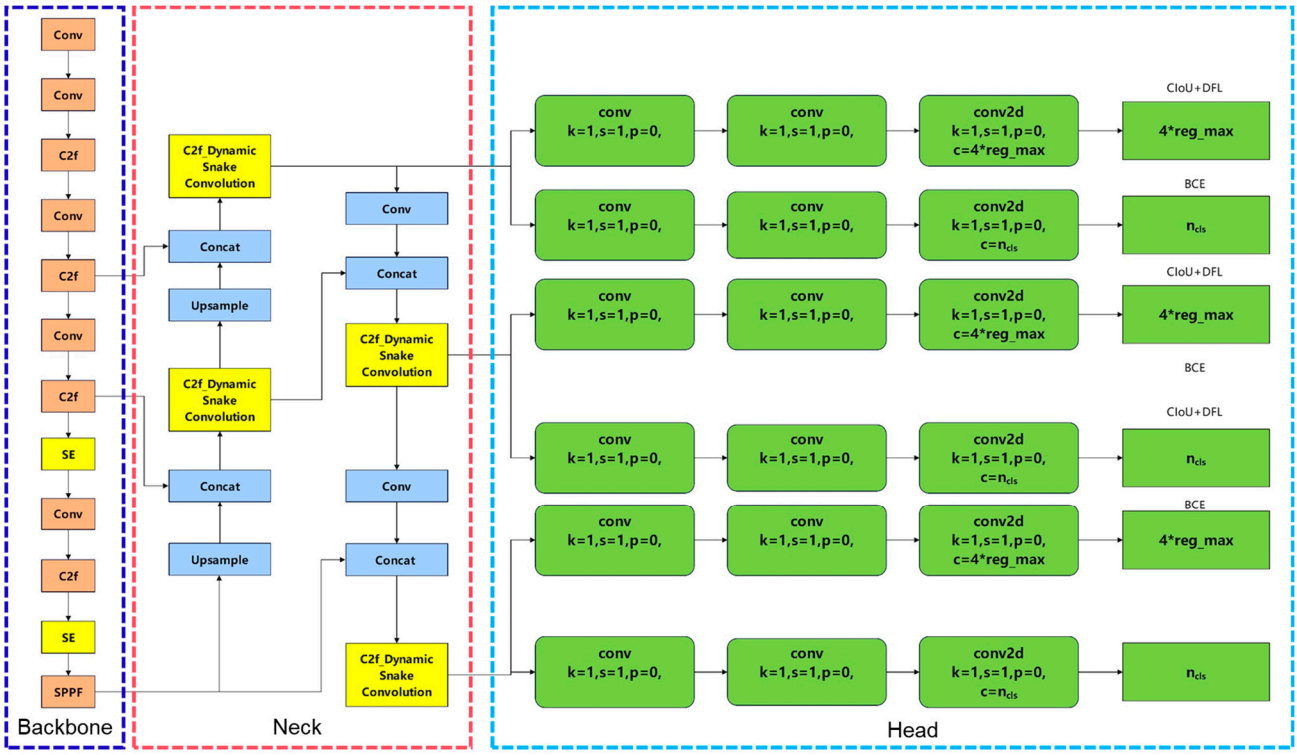


Figure 10. Architecture of improved YOLOv8 network.

3. Training and Evaluation of Model

3.1. Network Training

3.1.1. Training Platform

The training environment for intelligent perception models was established under the Windows 10 control system, with Intel (R) Core (TM) i5-10300H CPU utilized as the processor for model training, and the graphics card configuration Nvidia GeForce RTX 1650 used as the deep learning framework. Python (version 3.8) language was used to write the program code and call Cudnn, CUDA, OpenCV, and other required libraries (pytorch

1.13.0, cuda116, and ultralytics) to realize the training and testing of the fruit and branch perception model.

The logic of running the model mainly includes the following parts: configuring the environment for network operation, installing Anaconda (Anaconda is an open-source package and environment manager that can be used to install different versions of software packages and their dependencies on the same machine and can switch between different environments), installing an integrated development environment, creating a virtual environment, downloading the source code and weight files, installing basic dependencies, loading projects, setting up a virtual environment, and running model testing.

On the other hand, the AdamW algorithm was used as the optimizer, and the cosine learning rate scheduler was utilized to adjust the learning rate for model training. The cosine learning rate scheduler can help the model adjust its learning rate according to the shape of the cosine function during the training process, thereby using a higher learning rate in the early stages of training, which helps with fast convergence, and gradually reducing the learning rate in the later stages of training, which helps to finely adjust the model parameters.

The default hyperparameter settings in the original YOLOv8 project were as follows: The initial learning rate of model was 0.01, the weight attenuation coefficient was 0.0005, and the momentum was 0.937. Due to the features of the apple and branch targets being complex, the perception difficulty is relatively high; therefore, in the early stages of training, in order to ensure the feature extraction effect of the model, avoid oscillation during the gradient descent process of model training, make the convergence of the model more stable, and ultimately achieve higher accuracy, the initial learning rate was set lower than the default value to 0.002. On the other hand, in order to balance the efficiency of model training, the weight attention coefficient and momentum of training were appropriately increased, and the values were set to 0.05 and 0.9406, respectively. The input size of the image was 640 × 640. The number of training epochs was 300. After training, the weight file of the perception model obtained was saved, and the test set was used to evaluate the performance of the model.

3.1.2. Training Results

After training was completed, the loss value for each training iteration was obtained from the log file of model training and plotted as a curve graph. The loss curves of the YOLOv8s model and proposed model are shown in Figure 11. After 300 epochs of training, the loss value of the model reached a relatively low level. The output model after 300 rounds of training was utilized as the detection and segmentation model.

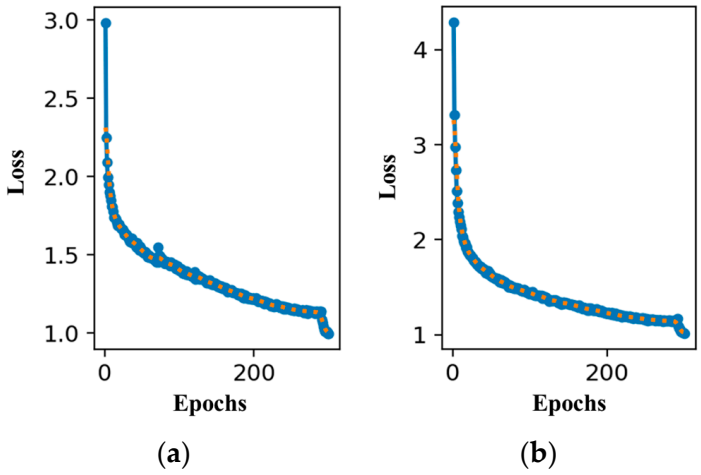


Figure 11. Training loss of YOLOv8s model (**a**) and proposed model (**b**).

From the figure, it can be seen that although the training loss value of the proposed model is relatively high in the initial stage of training, the training loss value shows a decreasing trend with the increase in training rounds, and the decrease rate of the loss value is slightly faster than that of the YOLOv8s model. On the other hand, when the model training reached 300 epochs, both the YOLOv8s model and the proposed model achieved low training loss values.

3.2. Evaluation of Detection and Segmentation Model

3.2.1. Evaluation Indicators for Apple Target Detection

In this study, objective evaluation indicators such as *precision* (3), *recall* (4), m*AP* (mean average precision) (5), and *F*1 score were used to evaluate the performance of the trained apple target identification model. The calculation equations are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

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

$$mAP = \frac{1}{C}\sum_{K=i}^{N} P(k)\Delta R(k) \tag{5}$$

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{6}$$

where *TP* means the number of correctly identified apple targets; *FP* means the number of misidentified background items as apple targets; *FN* represents the number of unidentified apple targets; *C* represents the number of target categories, which was set to 1 in the study; *N* represents the number of IoU thresholds; *K* is the IoU threshold; *P*(*k*) is the precision; and *R*(*k*) is the recall.

3.2.2. Evaluation Indicators for Apple Tree Trunk Segmentation

The Dice coefficient and IoU are evaluation indicators used to evaluate the segmentation effect for tree branches and trunks in images. The Dice coefficient is a set similarity measurement function commonly utilized to calculate the similarity between two samples, and its mathematical expression is as follows:

$$\text{Dice} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

Among them, *TP* (True Positive) represents the number of pixels predicted as positive samples (tree branch pixels) and perceived as positive samples (tree branch pixels), *FP* (False Positive) represents the number of pixels predicted as positive samples (tree branch pixels) but perceived as negative samples (background pixels), and *FN* (False Negative) represents the number of pixels predicted as negative samples (background pixels) but perceived as positive samples (fruit tree branch pixels). In addition, the mathematical expressions for the precision, recall, and mAP indicators in the evaluation of tree branch/trunk segmentation are shown in Equations (3), (4), and (5). The range of Dice coefficient values is between 0 and 1. The closer its value is to 1, the higher the overlap between the predicted results and true labels. The higher the similarity, the better the segmentation performance of the model.

IoU is a widely used metric in object detection and semantic segmentation and represents the ratio of the intersection and union of the predicted results and real labels. Its calculation formula is as follows:

$$\text{IoU} = \frac{TP}{TP + FP + FN} = \frac{Dice}{2 - Dice} \tag{8}$$

The calculation methods of the Dice coefficient and IoU are slightly different, with the main difference being that the Dice coefficient contributes equally to the intersection and union of the predicted results and real labels, while IoU focuses more on the intersection of the predicted results and real labels. Therefore, the Dice coefficient is more sensitive to smaller targets, while IoU focuses more on the segmentation of larger targets. Due to the varying sizes of apple tree branches/trunks that need to be segmented in the study, these two image segmentation evaluation indicators are introduced to comprehensively evaluate the segmentation performance of the model. Similar to the Dice coefficient, the value range of IoU is also between 0 and 1. The closer its value is to 1, the higher the overlap between the predicted results and real labels. The higher the similarity, the better the segmentation effect of the model.

4. Results and Discussion

4.1. Results and Analysis of Apple Detection and Branch/Trunk Segmentation Based on Improved YOLOv8s

In order to verify the performance of the designed apple target detection and branch/trunk segmentation model, further analysis was conducted on the recognition and segmentation results of the model on test set images. The experimental results of apple target detection and branch/trunk segmentation using the model are shown in Table 3. It can be seen that for apple targets, the precision, recall, and mAP values of the proposed model in the study are 99.6%, 96.8%, and 98.3%, respectively.

Table 3. Apple recognition and branch/trunk segmentation results using improved YOLOv8s network.

Improved YOLOv8s	Apple detection	Precision (%)	Recall (%)	mAP (%)
		99.6	96.8	98.3
	Branch/trunk segmentation	mAP (%)	Dice (%)	IoU (%)
		81.6	77.8	63.7

For the segmentation of apple tree branches and trunks, the mAP, Dice, and IoU values of the proposed model are 81.6%, 77.8%, and 63.7%, respectively.

Examples of the recognition and segmentation results of the proposed model for fruits and tree branches in different weather and lighting conditions are shown in Figure 12. Red boxes were used for the labeling of apple fruits. Blue masks were used for the labeling of branches. As can be seen in Figure 12, the proposed recognition and segmentation model is not only suitable to perceive the images captured under uniform illumination on cloudy days but also applicable to perceive the images captured under sunny conditions. Moreover, fruit targets could also be well recognized under frontlight, backlight, and sidelight conditions on a sunny day utilizing the proposed model.

Due to the fact that although the data used for verification and testing consist of 60 images, there are many target objects (fruits and tree branches/trunks) contained in a single image. On the other hand, the data analysis of the model performance testing results also demonstrates the objectivity of the model performance evaluation. Therefore, the amount of data used for validation and testing in this study meets the requirements for the evaluation of model performance.

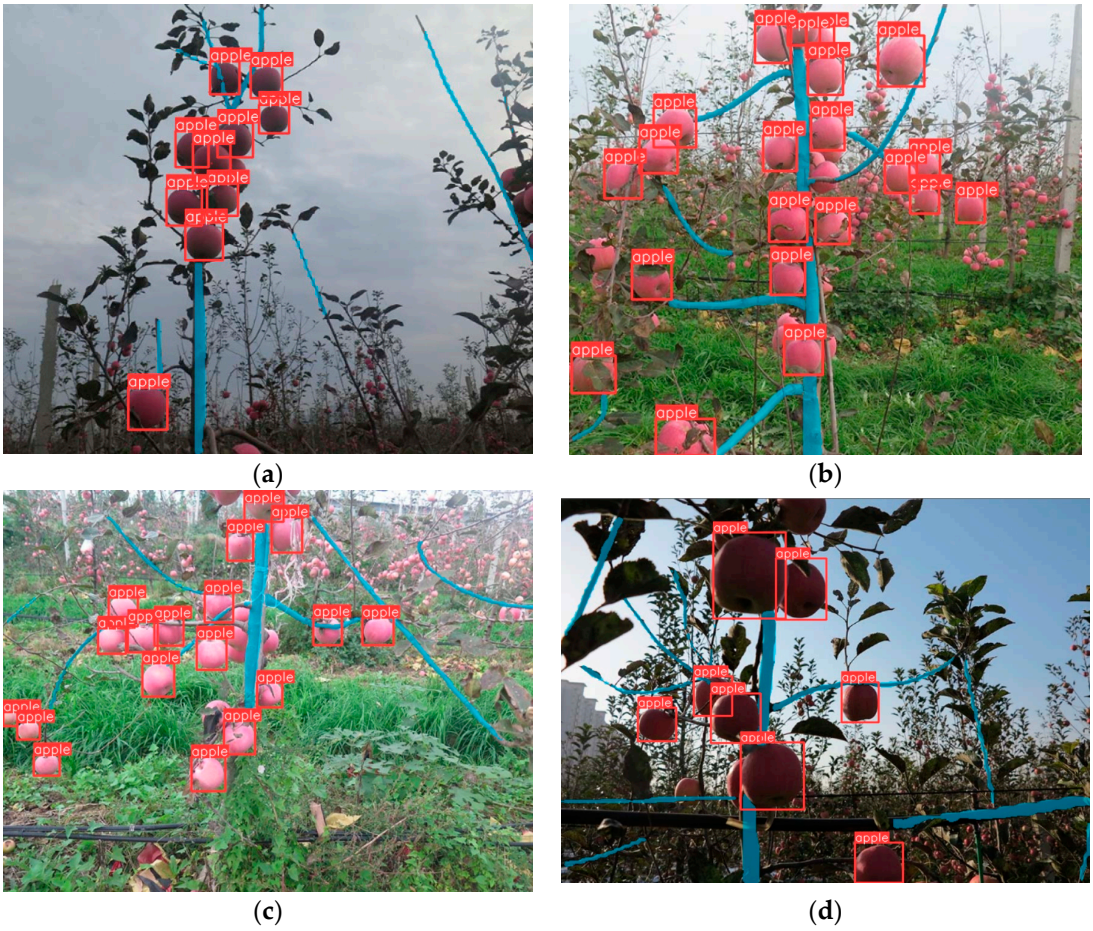


Figure 12. Apple recognition and branch/trunk segmentation results using improved YOLOv8s network: cloudy (**a**), sunny (**b**), frontlight of sunny (**c**), and backlight of sunny (**d**).

4.2. Comparison of Apple Detection and Branch/Trunk Segmentation Performance of Different Models

In order to further analyze the performance of the proposed apple detection and branch segmentation algorithm, the recognition and segmentation results of the improved YOLOv8s network were compared with the original YOLOv8s, YOLOv5s, and YOLOv8n networks on test set images. The experimental results of apple target detection using four models are shown in Table 4. It can be seen that for apple targets, the precision, recall, and mAP values determined using the proposed model were 99.6%, 96.8%, and 98.3%, respectively. Compared with the original YOLOv8s model, it has improved by 0.1%, 2.9%, and 1.5%, respectively. Compared with the detection performance of YOLOv8n and YOLOv5s model architectures, the mAP values increased by 2.3% and 6%, respectively, indicating the better apple detection performance of the proposed improved model. On the other hand, calculations were also made for the mAP 0.5–0.95 indicator, as shown in Table 4.

From Table 4, it can be seen that the perception times (ms/pic) of the four models for fruits and branches are 73.1 ms (YOLOv5s), 9.8 ms (YOLOv8n), 16.1 ms (YOLOv8s), and 17.7 ms (improved YOLOv8s), respectively. Although the improved YOLOv8s model increases the perception time for each image by 1.6 ms compared to the original model, it already satisfies the real-time recognition requirements of the picking robot vision system for targets.

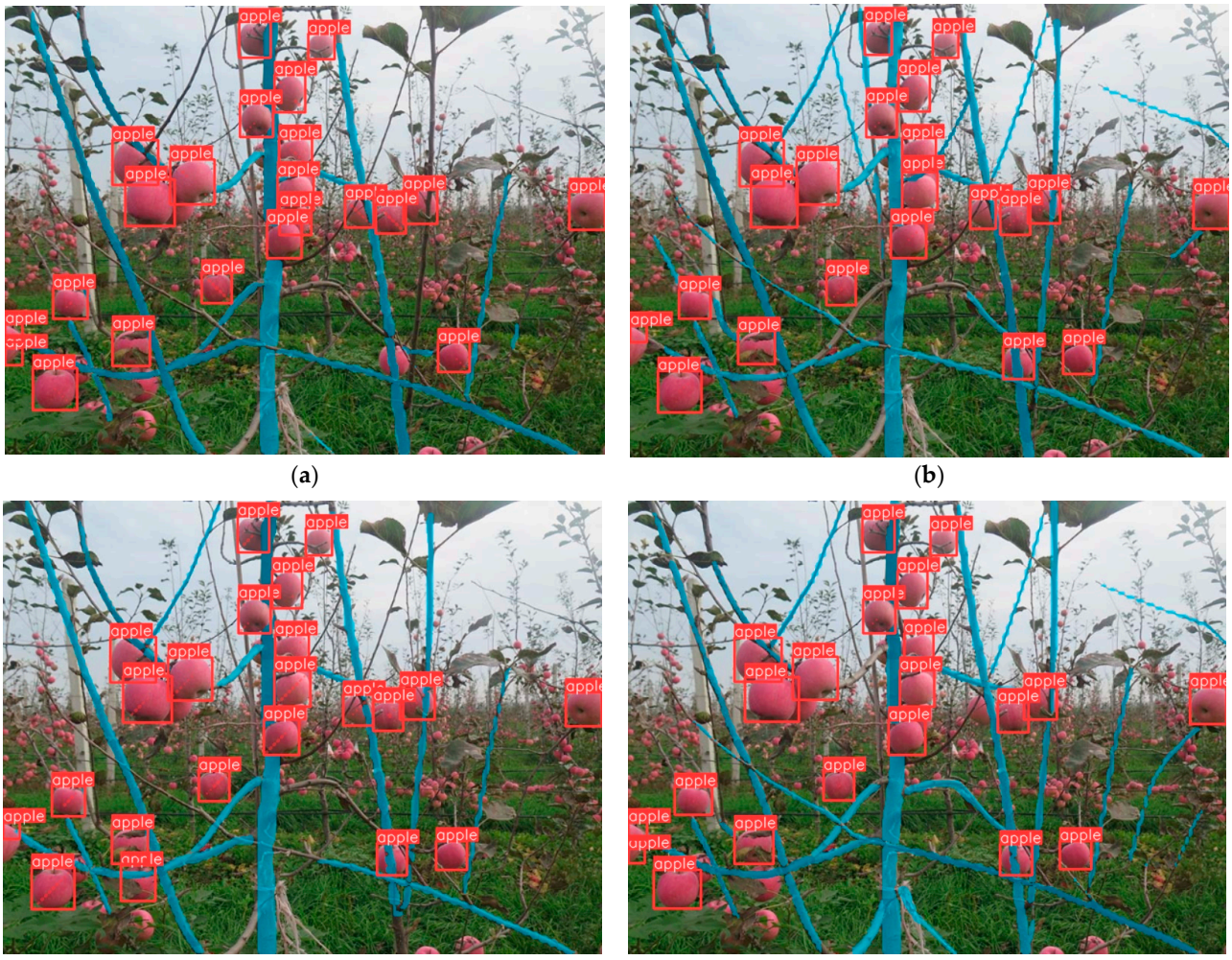
The experimental results of the four models for apple tree branch/trunk segmentation are shown in Table 4. It can be seen that the mAP value of the proposed model for pixel-level segmentation of the branch/trunk was 81.6%, which was 3.7%, 15.4%, and 24.4% higher

than the original YOLOv8s, YOLOv8n, and YOLOv5s models, respectively, indicating the better branch/trunk segmentation performance of the improved model.

Table 4. Performance comparison of four perception networks for apple recognition and branch/trunk segmentation.

Perception Model	Apple Recognition					Branch/Trunk Segmentation	Average Perception Speed (ms/pic)
	mAP (%) (mAP 0.5)	mAP 0.5–0.95 (%)	F1(%)	Precision (%)	Recall (%)	mAP (%)	
YOLOv5s	92.3	83.3	91.6	98.1	86.4	57.2	73.1
YOLOv8n	96	86.3	95.6	97.3	93.9	66.2	9.8
YOLOv8s	96.8	90.8	96.6	99.5	93.9	77.9	16.1
Improved YOLOv8s	98.3	94.8	98.2	99.6	96.8	81.6	17.7

The results of the four detection and segmentation models for identifying and segmenting spindle-shaped apple tree fruits and branches/trunks in modern apple orchards are shown in Figure 13. It can be seen that the perception results for target objects were more accurate utilizing the improved YOLOv8s network proposed in the study.



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

Figure 13. Examples of apple recognition and branch/trunk segmentation based on four perception network models: YOLOv5s (**a**), YOLOv8n (**b**), YOLOv8s (**c**), and improved YOLOv8s (**d**).

From the segmentation performances of several models on tree branches in the same image, it can be seen that the improved YOLOv8 model embedded with the dynamic snake convolution module can achieve segmentation of finer tree branches, which is more delicate than the original v8 model. This also reflects the characteristic of the dynamic snake convolution module in finely extracting small local structures and complex global shape (such as blood vessels) features from images.

In order to effectively monitor apple fruits throughout the entire growth process of the smart orchard, a lightweight model called YOLOv8-ShuffleNetv2-Ghost-SE has been proposed [37]. This method achieves smaller models and a faster detection speed and can serve as a reference for the development of smart devices in apple orchards. Another study [38] proposed an improved detection method based on the YOLOv8 deep learning model that can effectively detect flowers, fruits, and nodes in tomato plants. The proposed model integrates the Squeeze and Excitation (SE) block attention module into its Head architecture, enhancing the model's recognition ability by focusing more on the classes being studied, thereby improving the overall detection performance. Compared with our study, other studies [37,38] also applied SE modules in their improved model, but they did not extend the functionality of the YOLOv8 network and, therefore, did not achieve segmentation of the target object; that is, pixel-level segmentation of the target object could not be achieved. On the other hand, another study [37] focused on the detection of fruit targets. Due to the inability of the designed network to perform semantic segmentation, it cannot perceive apple tree branches that may become obstacles during the picking process. Therefore, this model cannot be applied to apple picking robots to perform fruit obstacle avoidance during picking operations. The discussion above also indicates the potential contributions of the proposed algorithm in our study to the precision and smart agriculture field.

5. Conclusions

(1) Considering the fact that most existing algorithms only recognize and detect fruit targets on apple trees without integrating segmentation perception for apple tree branches and trunks, the spindle-shaped fruit trees, which are widely planted in standard modern apple orchards, were focused on, and an intelligent perception algorithm for apple tree fruit detection and branch segmentation for picking robots was proposed based on an improved YOLOv8s model design, providing technical support for intelligent obstacle avoidance picking of apples using harvesting robots.

(2) The Backbone and Neck architectures of the original YOLOv8s network were improved by embedding the SE module visual attention mechanism behind the C2f module of the Backbone structure, and then, the dynamic snake convolution module was embedded into the Neck structure, achieving the enhancement of the ability to extract deep detail features from images and better extracting the features of different apple targets and detailed information of tree branches and trunks, with the perceptual performance of the deep learning models being optimized.

(3) The proposed improved network model can effectively recognize apple targets in images and segment tree branches and trunks. The experimental results of the test set showed that the recall for apple recognition was 96.8%, the precision was 99.6%, and the mAP value was 98.3%. The Dice value for branch and trunk segmentation was 77.8%, the IoU was 63.7%, and the mAP value was 81.6%.

(4) The proposed improved YOLOv8s algorithm was compared with the original YOLOv8s, YOLOv8n, and YOLOv5s algorithms in the recognition of apple targets and segmentation of tree branches and trunks on the test set. The results showed that compared with the other three algorithms, the improved YOLOv8s algorithm increased the mAP values for apple recognition by 1.5%, 2.3%, and 6%, respectively, and increased the mAP values for branch and trunk segmentation by 3.7%, 15.4%, and 24.4%, respectively.

6. Future Work

Although the fusion perception algorithm proposed in the study has important value for apple picking robots to achieve intelligent perception of fruit and branch information in apple trees, there are still some limitations in the algorithm. The proposed perception algorithm may not have a good performance in the perception of fruits and branches at night. In addition, there are many green apple trees in modern apple orchards besides red apple trees, but the proposed algorithm cannot be directly applied to perceive green apple fruits.

In order to improve the application scope of our algorithm and apple picking robot, a large amount of green apple image data will be captured. In addition, images of red and green apples and branches/trunks will also be captured at night using artificial lighting. All the above image samples will be added to the training set used to train the perception model in order to achieve automatic recognition of red or green apple targets and segment branches/trunks in daytime and night-time images.

YOLOv10 (official website: https://github.com/THU-MIG/yolov10, accessed on 27 July 2024) is the latest network architecture of the YOLO series and is designed for object detection tasks. The highlight of this network design lies in the use of dual-label allocation and consistent metric matching for free NMS, proposing a consistent dual allocation strategy to solve the redundant prediction problem in post-processing. It allows the model to enjoy rich and harmonious supervision during the training process while eliminating the need for NMS in the inference process, thereby achieving efficient, competitive performance. Secondly, by conducting a comprehensive inspection of each component in YOLO, a model design strategy driven by the overall efficiency accuracy of the model architecture is proposed. In order to improve the efficiency, a lightweight classification Head, spatial channel decoupled downsampling, and sorting guidance block design are proposed to reduce significant computational redundancy and achieve a more efficient architecture. In order to improve the accuracy, they explored large kernel convolution, proposed effective partial self-attention modules to enhance the model capability, and explored the potential for performance improvement at a low cost. According to the official model comparison test, YOLOv10 outperforms the YOLOv8 network architecture used in this paper in both accuracy and speed for object detection tasks. Therefore, the apple detection and branch/trunk segmentation based on the YOLOv10 network will have better performance. However, due to the fact that the officially released YOLOv10 project is only for object detection tasks, further programming operations are needed to achieve the fusion perception effect of fruits and branches in this article in order to expand the semantic segmentation function of the YOLOv10 network. On the other hand, the improved YOLOv8 algorithm proposed in this article can already meet the real-time operational requirements of apple picking robots in terms of its perception accuracy and speed in detecting fruits and branches. Therefore, in the future, we will conduct research on the fruit and branch perception algorithm of apple trees based on YOLOv10 while overcoming the limitations of the algorithm proposed in this paper in order to achieve all-weather fruit recognition and branch segmentation with multiple color schemes of apples based on YOLOv10.

On the other hand, as the proposed algorithm can achieve real-time perception of apple targets, branches and trunks, it can be combined with the motion control strategy of the grasping end effector in the future to achieve the picking of apples that are obstructed by branches or fruits by adjusting the picking angle and end effector position. Future work also includes identifying other apple tree varieties or horticultural fruits based on the proposed detection and segmentation algorithm.

Author Contributions: All authors contributed to this manuscript. B.Y. and Y.L. conceived and designed the study. B.Y. also contributed to obtaining the data, conceptualization, writing, and funding acquisition. Y.L. performed the experiments and analyzed the results. W.Y. contributed to the original draft preparation and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (Grant No. 62406244), the National Natural Science Foundation of China (Grant No. 62473311), and the Doctoral Research Project of Xi'an University of Technology (Grant No. 256082311).

Data Availability Statement: The original contributions presented in the study are included in the article material; further inquiries can be directed to the corresponding author.

Acknowledgments: We sincerely thank the editors and reviewers for their detailed comments and efforts toward improving our study.

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

References

1. Yan, B.; Fan, P.; Wang, M.; Shi, S.; Lei, X.; Yang, F. Real-time apple picking pattern recognition for picking robot based on improved YOLOv5m. *Trans. CSAM* **2022**, *53*, 28–38+59. [CrossRef]
2. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
3. Ma, H.; Li, Y.; Zhang, X.; Li, Y.; Li, Z.; Zhang, R.; Zhao, Q.; Hao, R. Target Detection for Coloring and Ripening Potted Dwarf Apple Fruits Based on Improved YOLOv7-RSES. *Appl. Sci.* **2024**, *14*, 4523. [CrossRef]
4. Liu, J.; Zhao, G.; Liu, S.; Liu, Y.; Yang, H.; Sun, J.; Yan, Y.; Fan, G.; Wang, J.; Zhang, H. New Progress in Intelligent Picking: Online Detection of Apple Maturity and Fruit Diameter Based on Machine Vision. *Agronomy* **2024**, *14*, 721. [CrossRef]
5. Sekharamantry, P.K.; Melgani, F.; Malacarne, J.; Ricci, R.; de Almeida Silva, R.; Marcato Junior, J. A Seamless Deep Learning Approach for Apple Detection, Depth Estimation, and Tracking Using YOLO Models Enhanced by Multi-Head Attention Mechanism. *Computers* **2024**, *13*, 83. [CrossRef]
6. Wu, X.; Qi, Z.; Wang, L.; Yang, J.; Xia, X. Apple Detection Method Based on Light-YOLOv3 Convolutional Neural Network. *Trans. CSAM* **2020**, *51*, 17–25. [CrossRef]
7. Zhao, H.; Qiao, Y.; Wang, H.; Yue, Y. Apple fruit recognition in complex orchard environment based on improved YOLOv3. *Trans. CSAE* **2021**, *37*, 127–135. [CrossRef]
8. Lu, S.; Chen, W.; Zhang, X.; Karkee, M. Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. *Comput. Electron. Agric.* **2022**, *193*, 106696. [CrossRef]
9. Kang, H.; Zhou, H.; Chen, C. Visual Perception and Modeling for Autonomous Apple Harvesting. *Ieee Access* **2020**, *8*, 62151–62163. [CrossRef]
10. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* **2020**, *171*, 105302. [CrossRef]
11. Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* **2020**, *197*, 245–256. [CrossRef]
12. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Comput. Electron. Agric.* **2020**, *176*, 105634. [CrossRef]
13. Long, Y.; Li, N.; Gao, Y.; He, M.; Song, H. Apple fruit detection under natural condition using improved FCOS network. *Trans. CSAE* **2021**, *37*, 307–313. [CrossRef]
14. Gao, F.; Wu, Z.; Suo, R.; Zhou, Z.; Li, R.; Fu, L.; Zhang, Z. Apple detection and counting using real-time video based on deep learning and object tracking. *Trans. CSAE* **2021**, *37*, 217–224. [CrossRef]
15. Zhang, Z.; Jia, W.; Shao, W.; Hou, S.; Ze, J.; Zheng, Y. Green Apple Detection Based on Optimized FCOS in Orchards. *Spectrosc. Spectr. Anal.* **2022**, *42*, 647–653. [CrossRef]
16. Sun, J.; Qian, L.; Zhu, W.; Zhou, X.; Dai, C.; Wu, X. Apple detection in complex orchard environment based on improved RetinaNet. *Trans. CSAE* **2022**, *38*, 314–322. [CrossRef]
17. Wang, Z.; Wang, J.; Wang, X.; Shi, J.; Bai, X.; Zhao, Y. Lightweight Real-time Apple Detection Method Based on Improved YOLO v4. *Trans. CSAM* **2022**, *53*, 294–302.
18. Hu, G.; Zhou, J.; Chen, C.; Li, C.; Sun, L.; Chen, Y.; Zhang, S.; Chen, J. Fusion of the lightweight network and visual attention mechanism to detect apples in orchard environment. *Trans. CSAE* **2022**, *38*, 131–142. [CrossRef]
19. Yang, F.; Lei, X.; Liu, Z.; Fan, P.; Yan, B. Fast Recognition Method for Multiple Apple Targets in Dense Scenes Based on CenterNet. *Trans. CSAM* **2022**, *53*, 265–273. [CrossRef]
20. Song, H.; Jiang, M.; Wang, Y.; Song, L. Efficient detection method for young apples based on the fusion of convolutional neural network and visual attention mechanism. *Trans. CSAE* **2021**, *37*, 297–303. [CrossRef]
21. Song, H.; Ma, B.; Shang, Y.; Wen, Y.; Zhang, S. Detection of Young Apple Fruits Based on YOLO v7-ECA Model. *Trans. CSAM* **2023**, *54*, 233–242. [CrossRef]
22. Zhong, H.; Zhang, Z.; Liu, H.; Wu, J.; Lin, W. Individual Tree Species Identification for Complex Coniferous and Broad-Leaved Mixed Forests Based on Deep Learning Combined with UAV LiDAR Data and RGB Images. *Forests* **2024**, *15*, 293. [CrossRef]
23. Zhao, X.; Zhang, W.; Zhang, H.; Zheng, C.; Ma, J.; Zhang, Z. ITD-YOLOv8: An Infrared Target Detection Model Based on YOLOv8 for Unmanned Aerial Vehicles. *Drones* **2024**, *8*, 161. [CrossRef]

24. Ye, R.; Gao, Q.; Qian, Y.; Sun, J.; Li, T. Improved YOLOv8 and SAHI Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea. *Agronomy* **2024**, *14*, 1034. [CrossRef]
25. Yang, S.; Yao, J.; Teng, G. Corn Leaf Spot Disease Recognition Based on Improved YOLOv8. *Agriculture* **2024**, *14*, 666. [CrossRef]
26. Wang, C.; Wang, H.; Han, Q.; Zhang, Z.; Kong, D.; Zou, X. Strawberry Detection and Ripeness Classification Using YOLOv8+ Model and Image Processing Method. *Agriculture* **2024**, *14*, 751. [CrossRef]
27. Sun, D.; Zhang, K.; Zhong, H.; Xie, J.; Xue, X.; Yan, M.; Wu, W.; Li, J. Efficient Tobacco Pest Detection in Complex Environments Using an Enhanced YOLOv8 Model. *Agriculture* **2024**, *14*, 353. [CrossRef]
28. Niu, S.; Nie, Z.; Li, G.; Zhu, W. Early Drought Detection in Maize Using UAV Images and YOLOv8+. *Drones* **2024**, *8*, 170. [CrossRef]
29. Ma, N.; Su, Y.; Yang, L.; Li, Z.; Yan, H. Wheat Seed Detection and Counting Method Based on Improved YOLOv8 Model. *Sensors* **2024**, *24*, 1654. [CrossRef]
30. Liu, M.; Cui, M.; Wei, W.; Xu, X.; Sun, C.; Li, F.; Song, Z.; Lu, Y.; Zhang, J.; Tian, F.; et al. Sorting of Mountage Cocoons Based on MobileSAM and Target Detection. *Agriculture* **2024**, *14*, 599. [CrossRef]
31. Lian, X.; Li, Y.; Wang, X.; Shi, L.; Xue, C. Research on Identification and Location of Mining Landslide in Mining Area Based on Improved YOLO Algorithm. *Drones* **2024**, *8*, 150. [CrossRef]
32. He, C.; Wan, F.; Ma, G.; Mou, X.; Zhang, K.; Wu, X.; Huang, X. Analysis of the Impact of Different Improvement Methods Based on YOLOV8 for Weed Detection. *Agriculture* **2024**, *14*, 674. [CrossRef]
33. Yu, S.; Xue, G.; He, H.; Zhao, G.; Wen, H. Lightweight Detection of Ceramic Tile Surface Defects on improved YOLOv8. *Comput. Eng. Appl.* **2024**, 1–19. [CrossRef]
34. Yao, J.; Qi, J.M.; Zhang, J.; Shao, H.M.; Yang, J.; Li, X. A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5. *Electronics* **2021**, *10*, 1711. [CrossRef]
35. Li, G.; Shi, G.; Zhu, C. Dynamic Serpentine Convolution with Attention Mechanism Enhancement for Beef Cattle Behavior Recognition. *Animals* **2024**, *14*, 466. [CrossRef]
36. Bai, Z.; Pei, X.; Qiao, Z.; Wu, G.; Bai, Y. Improved YOLOv7 Target Detection Algorithm Based on UAV Aerial Photography. *Drones* **2024**, *8*, 104. [CrossRef]
37. Ma, B.; Hua, Z.; Wen, Y.; Deng, H.; Zhao, Y.; Pu, L.; Song, H. Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. *Artif. Intell. Agric.* **2024**, *11*, 70–82. [CrossRef]
38. Firozeh, S.; Angelo, C.; Giovanni, D.; Angelo, P.; Stephan, S.; Francesco, C.; Vito, R. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [CrossRef]