agriculture

MDPI

Article

SGW-YOLOv8n: An Improved YOLOv8n-Based Model for Apple Detection and Segmentation in Complex Orchard Environments

Tao Wu [1], Zhonghua Miao [1], Wenlei Huang [1], Wenkai Han [2], Zhengwei Guo [2] and Tao Li [2,*]

[1] School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; 15139759901@163.com (T.W.); zhhmiao@shu.edu.cn (Z.M.); h_wenlei@163.com (W.H.)
[2] Intelligent Equipment Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China; m17165086050@163.com (W.H.); gzwneau@163.com (Z.G.)
* Correspondence: lit@nercita.org.cn

Abstract: This study addresses the problem of detecting occluded apples in complex unstructured environments in orchards and proposes an apple detection and segmentation model based on improved YOLOv8n-SGW-YOLOv8n. The model improves apple detection and segmentation by combining the SPD-Conv convolution module, the GAM global attention mechanism, and the Wise-IoU loss function, which enhances the accuracy and robustness. The SPD-Conv module preserves fine-grained features in the image by converting spatial information into channel information, which is particularly suitable for small target detection. The GAM global attention mechanism enhances the recognition of occluded targets by strengthening the feature representation of channel and spatial dimensions. The Wise-IoU loss function further optimises the regression accuracy of the target frame. Finally, the pre-prepared dataset is used for model training and validation. The results show that the SGW-YOLOv8n model significantly improves relative to the original YOLOv8n in target detection and instance segmentation tasks, especially in occlusion scenes. The model improves the detection mAP to 75.9% and the segmentation mAP to 75.7% and maintains a processing speed of 44.37 FPS, which can meet the real-time requirements, providing effective technical support for the detection and segmentation of fruits in complex unstructured environments for fruit harvesting robots.

Keywords: fruit detection; fruit segmentation; deep learning; occluded targets; attention mechanisms

Citation: Wu, T.; Miao, Z.; Huang, W.; Han, W.; Guo, Z.; Li, T. SGW-YOLOv8n: An Improved YOLOv8n-Based Model for Apple Detection and Segmentation in Complex Orchard Environments. *Agriculture* **2024**, *14*, 1958. https://doi.org/10.3390/agriculture14111958

Academic Editor: Yanbo Huang

Received: 4 October 2024
Revised: 26 October 2024
Accepted: 29 October 2024
Published: 31 October 2024

1. Introduction

As the fourth fruit in global production, the total annual production of apples is about 82.934 million tonnes, while China, as the first big producer of apples, has a total annual average apple production of nearly 45 million tonnes, accounting for about 53.66% of the total global apple production [1]. These data indicate that apples occupy a very important position in the fruit trade in China and the world. Behind the huge production volume is the support of huge productivity; although the existing agricultural automation equipment has greatly liberated the productivity and improved the efficiency of agricultural production [2], as far as apple picking is concerned, the research and development of the related agricultural automation equipment is still facing a lot of difficulties, the first and foremost is how to make the automated picking equipment really like a human being who can see the apples growing in the trees and real-time Detection and classification. Accurately achieving the detection and classification of fruits and providing feasible picking goals for automated picking equipment is to achieve the premise of automated fruit picking tasks.

In recent years, the integration of computer vision and artificial intelligence technology has been widely applied to the process of agricultural production [3], and more and more researchers and scholars are involved in using and optimising the means to promote the research and development and application of intelligent agricultural equipment, which

provides an effective method for solving the detection and classification of fruit in the production task of fruit picking [4]. The development of image processing techniques for fruit detection has gone through three stages: traditional digital image processing, machine learning-based image processing and deep learning-based image processing.

YOLO (You Only Look Once), as a representative of the most commonly used deep learning models, is favoured by scholars and researchers due to its end-to-end real-time processing speed, high prediction accuracy, global feature learning, and simplified training and inference process [5], and YOLOv8, as the newest deep learning model of the YOLO series, with its agricultural Outstanding performance is considered to be the most suitable deep learning model for outdoor complex unstructured environments [6]. Yang et al. [7] proposed a new LS-YOLOv8s model for detecting and grading strawberry ripeness, which is based on the YOLOv8s deep learning algorithm and integrates the LW-Swin Transformer module in the feature fusion stage. The Swin Transformer module was added to the TopDown Layer2 to capture long-range dependencies in the input data and to improve the generalisation ability of the model using a multi-head self-attention mechanism. Finally, a more efficient feature fusion network is achieved by introducing a residual network with learnable parameters and scaling normalisation into the original residual structure of the Swin Transformer, and ultimately LS-YOLOv8s achieves better detection accuracy and speed than YOLOv8m, using only about 51.93% of the parameters to achieve 94.4% of the detection accuracy and 19.23 fps detection speed, an improvement of 0.5% and 6.56 fps, respectively. In order to detect the exact picking location of the main stem of lychee, Qi et al. [8] proposed an open-access workflow using YOLOv5 and PSPNet as the main stem detection and segmentation models, respectively. The flow combines deep learning with traditional image processing algorithms to detect the main stem with YOLOv5, then use PSPNet for semantic segmentation, and finally obtain the pixel coordinates of the main stem picking point. The method has a recall of 76.29% and a precision of 92.50%, which lays the foundation for the subsequent acquisition of 3D coordinates. Zhang et al. [9] chose YOLOv8n as the base model and then replaced the YOLOv8n backbone structure with the Fasternet main module to improve the computational efficiency in the feature extraction process. Then, we redesign the PAN-FPN structure used in the original model to BiFPN structure to make full use of the high-resolution features and extend the perceptual field of the model while balancing the computational amount and model size and finally get the improved target detection algorithm YOLOv8-FCS, and the experimental results show that the grading accuracy of the YOLOv8-FCS model reaches 98.1%, the model size is only 6.4 M, and the FPS is 130.3. Wang et al. [10] proposed an improved PAE-YOLO model for the target detection problem of Yunnan millet spice in a complex background environment. The model integrates the EMA attention mechanism and DCNv3 deformable convolution to improve the feature extraction capability and model inference speed for small targets. The experimental results show that the model achieves a mean average precision (mAP) of 88.8%, an F1 score of 83.2, a model size of 5.7 MB, and GFLOPs of 7.6 G, which is better than the original model. Wang et al. [11] proposed an improved YOLOv8n-vegetable model for the detection of small hot targets in Yunnan millet. The new model enhances feature extraction through C2fGhost convolution and OAM attention mechanism, adds a small target detection layer, and applies the HIoU loss function to optimise regression. Experiments show that the model improves mAP by 6.46% on the vegetable disease detection dataset while reducing parameters and model size. Zhou et al. [12] proposed the DDSC-YOLO model by optimising the YOLOv8n model for the challenge of small target detection in UAV aerial images. The DualC2f structure and DCNv3LKA attention mechanism were introduced to enhance the feature extraction and the ability to adapt to different target sizes. The SDI-FPN and CASFF mechanisms are designed to improve the detection accuracy of small targets and the retention of contextual information. Tests on datasets such as VisDrone2019 show that DDSC-YOLO improves 9.3% over YOLOv8n on mAP0.5, demonstrating its superior generalisation ability. Yang et al. [13] improved YOLOv5s to recognise graspable (unobstructed) and non-graspable

(obstructed) apples on apple trees. They replaced the original BottleneckCSP module with the improved BottleneckCSP-2 module and inserted the SE module to enhance the feature representation; the feature map fusion method and the initial anchor frame size were optimised. These improvements enabled the model to excel in recall (91.48%), precision (83.83%), mAP (86.75%), and F1 (87.49%), and the recognition time per image was reduced to 0.015 seconds. The improved model has higher mAP and faster recognition speed compared to the original and other models, such as YOLOv3, YOLOv4 and EfficientDet-D0. Existing apple detection algorithms for apple-picking robots often perform poorly due to leaf occlusion, complex lighting and dense small targets. For this reason, Zhang et al. [14] designed an improved model based on the lightweight YOLOv4 with Ghost Net feature extraction, depth-separated convolution, and coordinate attention module to enhance detection accuracy and speed. On the Apple dataset, the improved model achieves a mAP of 95.72%, which is 3.45% better than YOLOv4, and performs well under multiple lighting conditions. Ma et al. [15] proposed a WL-YOLO model based on YOLOv5s to improve lightweight wildlife detection in complex forest environments, which effectively reduces the number of model parameters and enhances the feature representation by introducing depth-separated convolution, compressed excitation module, and CBAM attention mechanism. These improvements enable WL-YOLO to significantly enhance the detection performance in highly covert natural environments, achieving 97.25% mAP, 95.65% F1 score and 95.14% accuracy. Compared to YOLOv5m, WL-YOLO reduces the number of parameters by 44.73 per cent and cuts detection time by 58 per cent, dramatically improving detection efficiency and accuracy. Yuan et al. [16] proposed a grapefruit tree detection method based on attention mechanism and cross-layer feature fusion, introducing a hybrid attention mechanism module based on YOLOx-nano to improve feature extraction in space and on channel, using cross-layer feature fusion (CLFF) to exploit the complementarity of shallow detail information and deeper semantic information, and using the Ghost module instead of traditional convolution for feature extraction, reducing the influence of geometric changes, the number of parameters and computational complexity. Compared with the original model YOLOx-nano, the improved model has a significant increase in AP value from 93.08% to 93.74%, and the model size is only 7.8 MB, which results in faster detection speed, better small target detection capability and anti-obscuration performance. In order to improve the accuracy of fruit tree canopy identification and counting, Zhu et al. [17] proposed an improved YOLOv4 model combining Mobilenetv3, CBAM and ASFF, and optimised pre-selected frame generation using K-means, trained with a cosine annealing strategy. The model achieves fast and accurate detection and counting on UAV images with a mAP of 98.21%, an FPS of 96.25, an F1 score of 93.60%, and an average overall accuracy of 96.73%, which is suitable for digital and intelligent management of orchards. To solve the problem of dense small objects and background noise faced by small object detection in UAV image scenes, Ni et al. [18] proposed an enhanced YOLOv8s model by introducing a parallel multiscale feature extraction module (PMSE) to enhance small object feature extraction through parallel expansion and deformation convolution, and designing a scale-compensated feature pyramid network (SCFPN) to integrate shallow and deep feature information. In addition, the largest object detection layer is removed from the original detection head, the ultra-small object detection layer is added, and the WIOU loss function is used to balance the sample quality. Experiments show that the model improves the accuracy of small object detection on public datasets. The above existing studies were basically conducted in outdoor complex unstructured environments, and they mainly focused on the improvement and optimisation of the three modules of the YOLOv8 network model, namely convolution, attention, and loss function, and ultimately achieved the improvement of the improved model in terms of the average accuracy (mAP), F1 score, and other performance mountains, which provides a good opportunity for subsequent related studies. It provides a good reference for the subsequent related research.

This study focuses on how to solve the problem of detecting and segmenting apples occluded by rigid obstacles (steel wires, steel pipes, thick branches) and flexible obstacles

(leaves, thin branches) in complex unstructured environments by using a network algorithm based on the YOLO deep learning to detect, classify, and segment apples (Obvious, Occluded, and Risky) at different locations in a tree by using a network algorithm based on the YOLO deep learning. This study improves the recognition and segmentation accuracy of occluded targets in complex unstructured environments, improves the picking efficiency of picking robots, and promotes the development of intelligent equipment for fruit picking.

2. Materials

2.1. Collection and Processing of Datasets

The team completed the acquisition of the image dataset on 31 July 2024 in a standardised orchard in Yuncheng City, Shanxi Province, China, as shown in Figure 1.

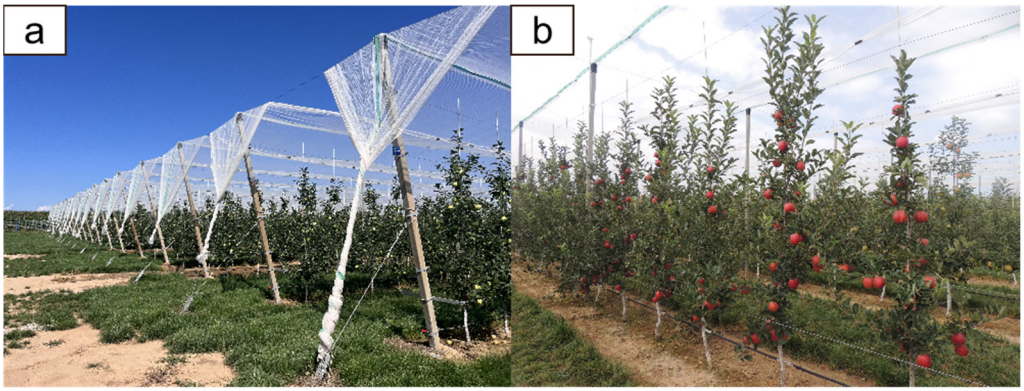

Figure 1. (**a**) An orchard in Yuncheng (**b**) A photograph of the fruit trees.

This dataset is acquired by the tracked automatic acquisition trolley shown in Figure 2. The data acquisition process is as follows: firstly, the two Realsense depth cameras in Figure 2 are used to realise the image acquisition, where camera b is about 2.1 m from the ground, camera c is about 1.1 m from the ground, and the vertical distance between the two cameras is 1 m. This layout not only enables a wider range of images to be acquired but also allows the acquired images to have a lesser overlapping area, and then the two depth cameras are connected to the laptop through the Type-C UCB3.0 data transfer cable connects the two depth cameras and the laptop together, using the program to drive the depth cameras and display the colour images acquired by the cameras in real-time on the desktop of the laptop and the resolution of the colour images acquired by the depth cameras is set to 640 × 480, and finally, the data acquired by the two depth cameras are saved in the form of Rosbag by recording them into the corresponding Rosbag. Finally, the data collected by the two depth cameras are saved to the corresponding Rosbag by recording Rosbag. The tracked automatic acquisition trolley is manually operated by a remote control, and the recorded Rosbag is used to separate the video stream into image datasets by frame-splitting operation, which is then filtered by human beings to obtain the final usable dataset.

The data collection work was carried out in three time periods, respectively, 7:00 to 9:00 a.m., 12:00 to 2:00 p.m., and 5:00 to 7:00 p.m. in the evening, and a total of 3091 pieces of picture data were collected, as shown in Figure 3, which covered various growth distributions of apples on the fruit trees in the orchard at different times of the day and in different light conditions, and then the data set was annotated by using a software called Anylabeling's automatic (https://pypi.org/project/anylabeling/0.1.2/, accessed on 5 October 2024). An automatic labelling software called Anylabeling is used to annotate the dataset, and the label file generated by the annotation is in json format, and then the json file is converted into a txt file for YOLOv8 training through the program. As shown in Figure 4, based on the experience of many experiments in the previous period and the orchard test, it was finally determined that the growth of apples on the fruit tree was classified into the following three categories: ① Obvious: completely exposed and unobstructed, or obstructed by

leaves and branches not more than 50% of the area of the apple's front view; ② Occluded: obstructed by leaves and branches more than 50% of the area of the apple's front view, or in the edge area of the image; ③ Risky: occluded by wire or steel pipe. Finally, the dataset was randomly generated into a training set, test set and validation set for training in the ratio of 8:1:1, of which 2473 were for the training set, 309 for the test set and 309 for the validation set.

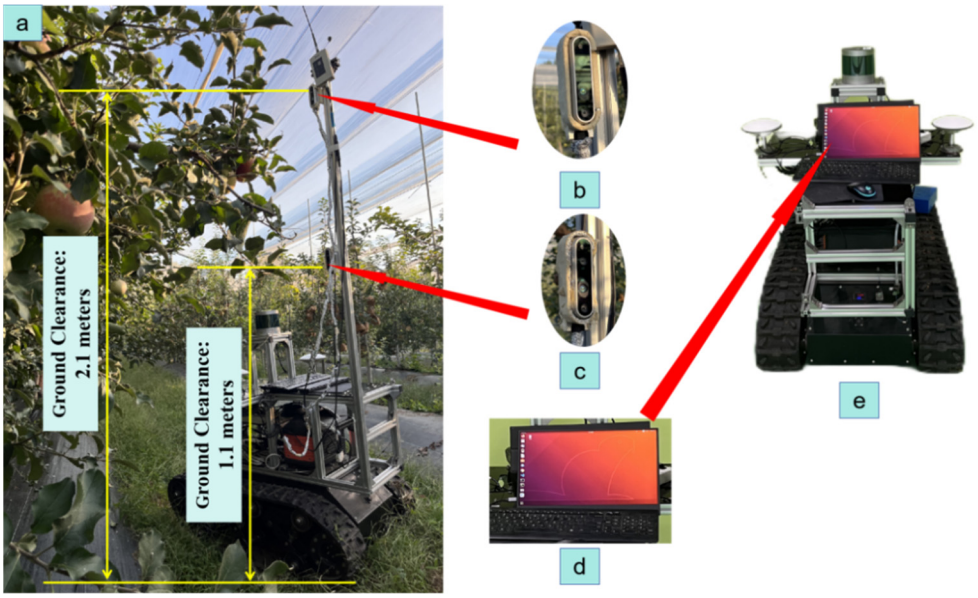

Figure 2. Picture of tracked automatic collection trolley operation. (**a**) Data collection site; (**b**) Reansense 1; (**c**) Reansense 2; (**d**) Data storage laptop; (**e**) Track-mounted automatic collection vehicle.

Figure 3. Sample dataset under different lighting conditions. (**a**) Image data under front lighting conditions; (**b**) Image data under backlighting conditions; (**c**) Image data under strong lighting conditions; (**d**) Image data under low lighting conditions.

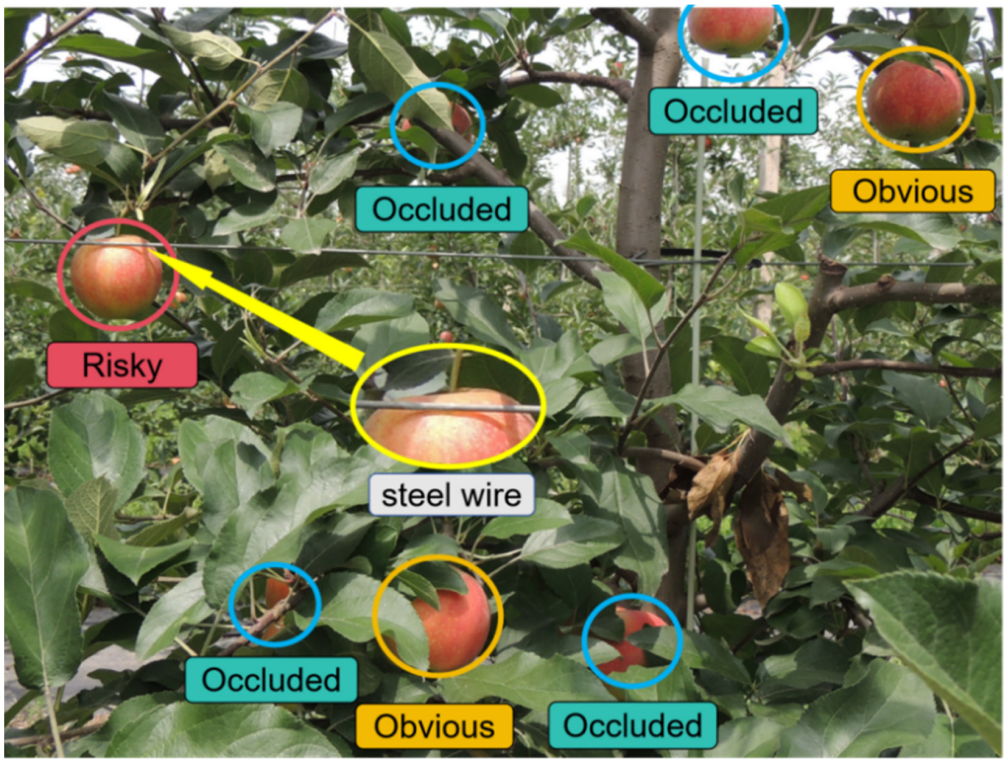

Figure 4. Introduction to the labelling categories of the dataset.

2.2. Hardware and System Environment

The model training platform is equipped with NVIDIA's GeForce RTX 4090 GPU high-performance graphics card, which boasts 40.32 TFLOPS of floating-point computing power and 24 GB of graphics memory, enough to complete the training of complex deep learning models. The central processing unit is an Intel Xeon Platinum 8352 V with 10 cores and a 2.10 GHz clock frequency, paired with 64 GB of system memory to support data processing and multitasking needs. Moreover, 200 GB of storage space is used to ensure sufficient data caching and fast accessibility.

For software, the stable and compatible Ubuntu 20.04 LTS was chosen as the operating system, which is particularly suitable for scientific research and high-performance computing applications. The development environment includes Python 3.9 and Pytorch 2.0, both of which provide me with powerful programming and deep learning library support. It also draws on NVIDIA's CUDA 11.7 and cuDNN 8 plug-ins, which maximise GPU performance for the massively parallel computations required in data science projects.

3. Methods

3.1. Network Model Based on Improved YOLOV8n—SGW-YOLOV8n

With the development of intelligent and automated agricultural machinery, the YOLO series of deep learning network models have been widely used in agricultural fields such as pest and disease detection and classification, crop maturity judgement, poultry counting, fruit segmentation, etc. YOLO was initially proposed by Joseph Redmon [19] in 2016, which is based on the principle of treating the target detection as a single regression problem by a single forward propagation of a convolutional neural network (CNN), which directly predicts multiple bounding boxes and categories in an image, thus greatly improving the detection speed. After continuous development and production needs, YOLOv8 proposed by Ultralytics in the USA in 2023 became the mainstream of the YOLO family of network models, which builds on the real-time detection characteristics of the YOLO family of models by using optimised lightweight backbone networks (MobileNet [20], EfficientNet [21], ShuffleNet [22]), multi-scale feature fusion (FPN [23], PANet [24], DeepLabV3+ [25], Re-

fineNet [26], BiFPN [27]), the Anchor-Free [28] mechanism, and automated hyper-parameter optimisation, which significantly improves the speed and accuracy of the target detection, and also supports multi-task processing such as image segmentation and keypoint detection, which is suitable for a variety of practical application scenarios. In this study, YOLOv8 is chosen as the native network model for improvement, and the improved network model is named SGW-YOLOv8n, whose network structure is shown in Figure 5. In order to solve the problem of detecting and segmenting the different postures of apples on the tree, such as ① Obvious, ② Occluded by leaves and branches, and ③ Risky, and to provide more accurate and feasible picking targets for apple picking robots under the complex and unstructured outdoor environments, YOLOv8n is selected as the reference model among the YOLO series of network models, and the optimised and improved network model is designed.

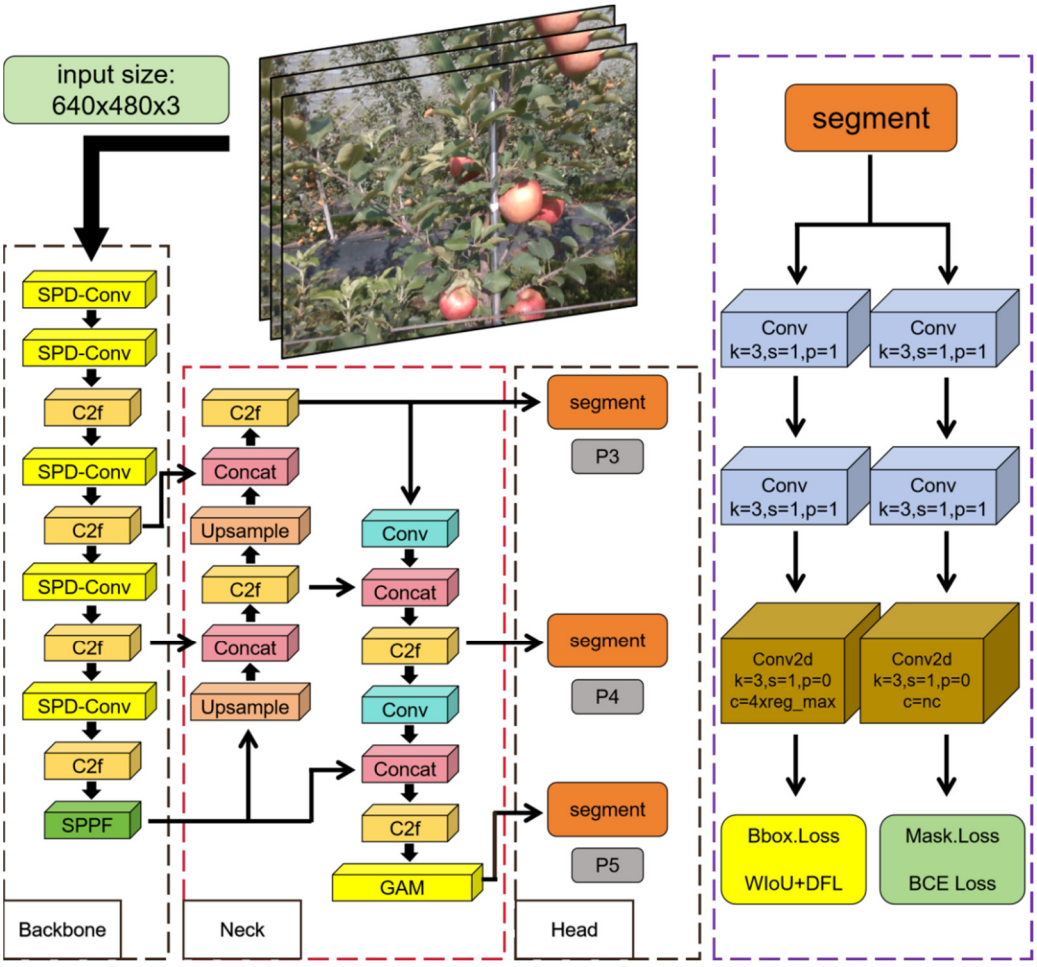

Figure 5. Structure of the SGW-YOLOv8n network.

Firstly, the original convolution module is replaced with the SPD convolution module in the backbone Backbone network structure of the YOLOv8n network model, which effectively solves the performance degradation of low-resolution images and small object detection by replacing the traditional step convolution and pooling layer with the spatial-to-channel conversion and non-stepwise convolution; and then a global attention module is added at the tail end of the Neck neck network module is added to the tail of the Neck neck network, and the channel attention module and spatial attention module are introduced to weight the input features to reduce the information reduction and amplify the global dimensional interaction features; finally, in the segmentation head part, the original CIoU (Complete Intersection over Union Loss) loss function is replaced by the WIoU (Wise Intersection over Union (WIoU) loss function in the segmentation head part,

the adaptive weighting mechanism is introduced to dynamically adjust the weights of the loss function according to the shapes and sizes of different targets as well as other features, which enhances the adaptive ability of the model to the complex scenes and targets, and thus further improves the performance of the model. Since this model is based on the YOLOv8n network architecture, replacing Conv with SPD-Conv in the convolutional layer of the backbone network, adding a global attention module in the neck, and replacing the CIoU loss function with the WIoU loss function, the improved network model is called SGW-YOLOv8n.

3.2. SPD-Conv Module and Its Composition

3.2.1. Overview of the SPD-Conv Module

The original convolution module in YOLOv8 is shown in Figure 6, where the default values of the convolution kernel size k and step size s are all 1, the default value of the padding p is None, the activation function is SiLU, and the default in the state is True. From the figure, it can be seen that the convolutional layer has a simple structure that is easy to understand and implement, and it has a good ability to generalise, which can be achieved by sharing the weights across the entire input image to reduce the model's parameter number, thus achieving the effect of reducing the risk of overfitting and increasing the speed of model training, but since Conv is a standard convolutional layer, it is unable to capture more complex or abstract features and a single size convolution is not enough to capture all the feature information, all these will lead to unsatisfactory training results in the end.2022 Raja Sunkara et al. [29] proposed an improved Convolutional Neural Network (CNN) innovative module SPDConv for processing images, replacing the step-size convolutional and pooling layers in traditional CNN architectures. SPD-Conv consists of a space-to-depth (SPD) layer and a non-step-size convolutional layer (Conv), and new CNN architectures are created by applying SPD-Conv to YOLOv5 and ResNet, and experimentally showing that the method significantly outperforms state-of-the-art deep learning models, especially on more difficult tasks with low-resolution images and small objects. The principle of operation is shown in Figure 7. The network components, the channel attention mechanism, and the spatial attention mechanism are described in detail below.

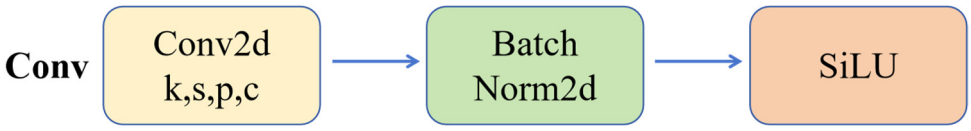

Figure 6. Schematic diagram of the operation of Conv Convolution.

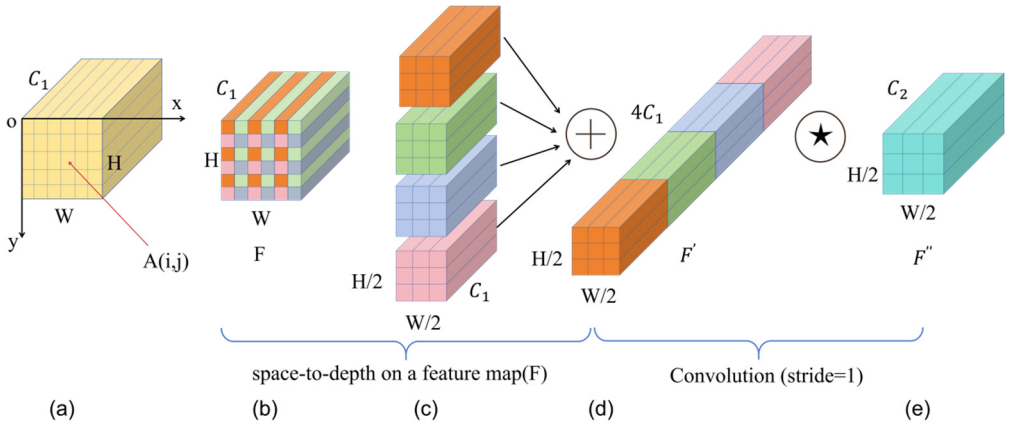

Figure 7. Operating schematic of SPD-Conv. (**a**) Any pixel point on the feature map; (**b**) Feature map F; (**c**) The four feature maps obtained after decomposition; (**d**) The new feature map F' after rearrangement; (**e**) The feature map F'' obtained through non-strided convolution.

3.2.2. Space-to-Depth Layer (SPD)

SPD acts as a transformation layer that converts the spatial dimension of the input image into a depth dimension, thereby increasing the depth of the feature map without losing information. The reason for using the SPD layer is that when dealing with low-resolution images and small objects, it is necessary to retain as much spatial information as possible. The SPD layer avoids the loss of information in the traditional stepwise convolution and pooling operations by converting the information from spatial dimension to depth dimension; its principle is as follows.

Suppose there is a feature map F, with dimensionions $H \times W \times C$, where H is the height, W is the width, and C is the number of channels. In performing the SPD transformation, we choose a scale factor s (here s = 2). The transformation rearranges each s *x* s block in F to the channel dimension, a new feature map with dimensions is yielded so that the new-to-feature map dimensions in the figure are. Assuming the position of any pixel point in the feature map F, its new position in the feature map is calculated as:

$$i' = \left\lfloor \frac{i}{s} \right\rfloor \tag{1}$$

$$j' = \left\lfloor \frac{j}{s} \right\rfloor \tag{2}$$

$$c' = c + (i \bmod s) \cdot s + (j \bmod s) \cdot s \cdot C \tag{3}$$

In the above equations, i', j' are the spatial coordinates in the new feature map, c' is the new channel index, c is the channel index representing the original feature map F, and C represents the total number of channels in the original feature map F.

3.2.3. Non-Step Convolutional Layers

The convolutional layer applied after the SPD transformation does not use a step size to preserve fine-grained information. The reason for applying the non-step-size convolutional layer after the SPD layer is that the non-step-size convolution is able to perform feature extraction without reducing the size of the feature map, further preserving the fine-grained information of the image, which is crucial for improving the recognition performance of low-resolution images and small objects. Assuming that the number of filters for non-stepwise convolution is set to be C_2 (the number of output channels) and the filter size is $k \times k$, the size of the output feature map F'' is $\frac{H}{s} \times \frac{H}{s} \times C_2$. Assuming that the position of any pixel points in the feature map F' is $B(i, j)$, its new position in the feature map is calculated as follows:

$$F''(\mathrm{m}, n, d) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \sum_{c=0}^{s^2C-1} K(i, j, c, d) \cdot F'(m + i, n + j, c) + b_d \tag{4}$$

Equation (4) shows a standard convolution operation in convolutional neural networks with a default step size of 1, which is mainly used to compute an element of the output feature map. Specifically, for the position (m, n) and channel d in the output feature map F''; its value is obtained from the input feature map F' with the convolution kernel K through convolution, and finally a bias b_d is added.

In summary, the SPD-Conv convolution module preserves a large amount of detailed information through a spatial-to-depth transformation while reducing the spatial dimension. This transformation effectively reduces the loss of information by compressing the spatial information into the channel dimension, which is especially effective for capturing small objects or detailed features. The use of non-spanning convolution further enhances the feature extraction capability of the network, allowing it to capture finer features while preserving spatial location, making it ideal for applications requiring highly accurate feature recognition, such as fine-grained image classification. SPD-Conv is designed not only

to be easy to integrate into existing CNN architectures but also to be flexible enough to adapt to a wide range of complex vision tasks by adjusting its parameters, which makes it show excellent performance in small object recognition and processing low-resolution images. Moreover, its high compatibility and scalability make it a powerful and practical tool for a wide range of vision-processing tasks.

3.3. Global Attention Mechanism Module

3.3.1. Overview of Global Attention Mechanisms

The global attention mechanism was proposed by Liu et al. [30] in 2021, and its network structure is shown in Figure 8, assuming that the input is a three-dimensional degree feature map F, where H is the height of the feature map, W is the width of the feature map, and C is the number of channels of the feature map (also known as the number of depths or convolutional kernels), and the input feature maps pass through the channel attention module, in turn, The input feature maps are sequentially passed through the channel attention module, which assigns weights to each channel in the channel dimension to enhance the important channels; and then, through the spatial attention module, which assigns weights to each pixel location in the spatial dimension to highlight the important region where the target is located. Ultimately, the combination of channel attention and spatial attention generates a more discriminative feature map, which is used to enhance the performance of target detection or classification. Because of its ability to capture global contextual information and dynamically adjust the feature maps, GAM performs well in complex outdoor scenes and operates efficiently on edge computing devices. The network components, channel attention mechanism and spatial attention mechanism, are described in detail below.

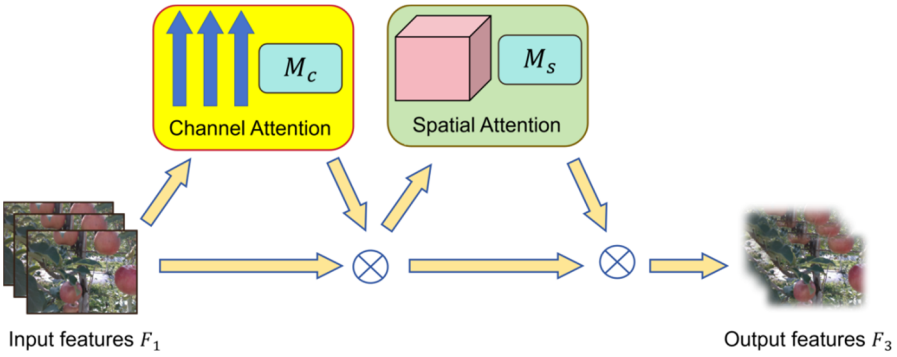

Figure 8. Schematic diagram of GAM operation.

3.3.2. Channel Attention Mechanism

The working principle of the channel attention mechanism is shown in Figure 9, assuming that the input is a three-dimensional feature map $F \in R^{H \times W \times C}$, where H is the height of the feature map, W is the width of the feature map, and C, which is the number of channels of the feature map. The channel attention mechanism aims to enhance the network's attention to the important channels in the feature map and automatically assign weights to different feature channels to improve the target's feature extraction ability. In the deep convolutional neural network, each channel represents a different feature, such as colour, texture, edge, etc., while the channel attention mechanism improves the performance of the model by emphasising the important channels and weakening the irrelevant or redundant channels. The basic process of the channel attention mechanism usually includes the following steps:

1. Global information extraction

$$F_{avg}(C) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_c(i, j) \tag{5}$$

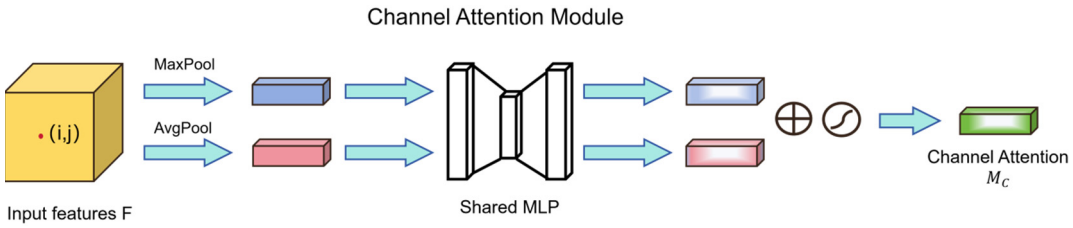

Figure 9. Schematic diagram of channel attention mechanism.

Equation (5) is a global average pooling of each channel of the feature map, calculating the average of all pixel points on each channel. This will compress each channel into a scalar and extract the overall feature information on the channel.

$$F_{max}(C) = \max_{i,j} F_c(i,j) \tag{6}$$

Equation (6) is to perform maximum pooling on each channel of the pair feature map and extract the most significant feature value in each channel. Maximum pooling preserves the feature with the strongest response in each channel.

2. Channel weight generation

$$W_C = \sigma\left(W_2\left(\mathrm{Re}LU\left(W_1\left(F_{avg}/F_{max}\right)\right)\right)\right) \tag{7}$$

Equation (7) is a calculation formula of the weight generation process, where W_1 and W_2 are weight matrices of the first and second layers, respectively, $ReLU$ is an activation function, and σ is a Sigmoid function for limiting the output weight value between 0 and 1.

3. Weight adjustment

$$F_{out}(i,j,c) = W_c \cdot F(i,j,c) \tag{8}$$

Equation (8) is to apply the generated channel weight $\boldsymbol{W_c}$ to the corresponding feature map $\boldsymbol{F}$, so as to adjust the feature intensity of each channel.

3.3.3. Spatial Attention Mechanism

The role of the spatial attention mechanism is to highlight the key areas in the image and ignore the unimportant parts by analysing the importance of the feature map in the spatial dimension (i.e., position in height and width). Different from the channel attention mechanism, channel attention focuses on the weight of different features, while spatial attention focuses on the pixel position of the feature map and strengthens the spatial region where the target object is located, thus helping the model to locate the target more accurately. The workflow of the spatial attention mechanism usually consists of the following steps:

1. Feature map aggregation

$$F_{avg}(i,j) = \frac{1}{C}\sum_{k=1}^{C} F(i,j,k) \tag{9}$$

Equation (9) is to calculate the average of the feature map at spatial location $(\boldsymbol{i},\boldsymbol{j})$ over the channel dimension. Where $\boldsymbol{C}$ is the number of channels and is averaged over all channels at position $(\boldsymbol{i},\boldsymbol{j})$. Therefore, this formulation is used to extract global features at spatial locations but for the aggregated results of multiple channels.

$$F_{\max}(i,j) = \max_{k} F(i,j,k) \tag{10}$$

Equation (10) is to calculate the feature value on the $\boldsymbol{k}(\boldsymbol{k} \in [1,\boldsymbol{C}])$ channel of the feature map at the spatial position $(\boldsymbol{i},\boldsymbol{j})$ and output the maximum feature value and highlight

the significant feature of each position in the image by selecting the maximum channel response of each position.

2. Generate a spatial attention map

$$F_{concat}(i,j) = Concat\left(F_{avg}(i,j), F_{max}(i,j)\right) \tag{11}$$

Equation (11) is a splicing operation of the results of the global average pooling $(\boldsymbol{F_{avg}})$ and global maximum pooling $(\boldsymbol{F_{max}})$ in the channel dimension to generate a new feature map ***F_concat***, where ***F_concat*** $\in \boldsymbol{R}^{W\times H\times 2}$ is a feature map containing two channels.

$$M_S(i,j) = \sigma(Conv_{3\times 3}(F_{concat}(i,j))) \tag{12}$$

Equation (12) is a representation of the process of generating the spatial attention map $\boldsymbol{M_s}$ by the spatial attention mechanism. By performing convolution operation on the spliced feature map ***F_concat***, and then using the Sigmoid activation function to generate the weight map $\boldsymbol{M_s}$ for each spatial location; its function is to assign different weights to each spatial location so as to enhance the features of the important regions and ignore the irrelevant background parts.

3. Weighting adjustment

$$F_{out}(i,j,k) = M_S(i,j) \cdot F(i,j,k) \tag{13}$$

Equation (13) represents the process of generating the output feature map in the spatial attention mechanism, by multiplying the spatial attention map $\boldsymbol{M_s}$ with the original feature map $\boldsymbol{F}$ pixel by pixel, the feature value of each spatial location is adjusted, so that the model can better focus on the important regions and ignore the unimportant background information.

The Global Attention Mechanism (GAM) has shown significant benefits in target detection tasks in outdoor environments, mainly because of its ability to efficiently capture global contextual information and emphasise important regions, its principle is shown in Figure 10. The GAM utilises both channel and spatial attention mechanisms to enhance the feature representation of the target, improve the robustness of the model under complex lighting and background interference, and also optimise the recognition of occluded targets, especially in orchards where the fruit detection tasks. In addition, GAM is adaptable to the detection of multi-scale targets and can enhance the model's adaptability under various lighting variations and extreme weather conditions while maintaining low computational complexity, making it ideal for applications in embedded and real-time processing platforms. These features make GAM ideal for high-precision and high-efficiency target detection in outdoor environments.

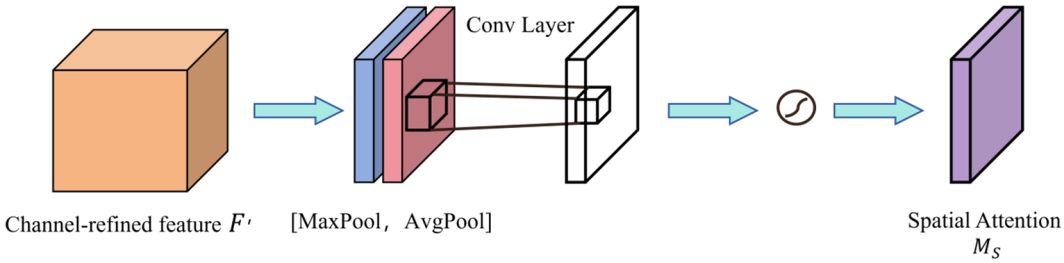

Figure 10. Schematic diagram of spatial attention mechanism.

3.4. Wise Intersection over Union (Wise-IoU) Loss Function Module

3.4.1. The Proposal of Wise-IoU and Its Core Mechanism

Wise-IoU was proposed by Zan et al. [31] in 2023 to improve the performance of traditional IoU loss in Bounding Box Regression (BBR) mainly through a dynamic focusing

mechanism. It combines distance and geometric factors by adjusting the gradient to help the model optimise the bounding box more efficiently. The core mechanism of Wise-IoU is as follows: firstly, it introduces a dynamic focusing mechanism, which dynamically adjusts the loss value according to the quality of the overlap between the anchor box and the target box. Lower-quality anchor frames will significantly amplify the loss and help the model learn better from non-overlapping or less overlapping frames. And for higher quality anchor frames (i.e., those that are highly overlapped with the target frame), it will attenuate the effect of the loss value so as to reduce the unnecessary gradient interference; the distance-attention mechanism is introduced, which pays particular attention to the distance between the centroids of the bounding boxes. By calculating the distance between the anchor frame and the centre point of the target frame, Wise-IoU is able to effectively reduce the position error between the anchor frame and the target frame. This approach helps the model to adjust the position of the anchor frame more accurately, especially in the case of poor overlap between the two; the introduction of the weight dynamic adjustment mechanism enables the model to converge faster by dynamically adjusting the gradient of different anchor frames. Its dynamic adjustment not only relies on the distance between the anchor frame and the target frame but also combines geometric information such as the size ratio of the anchor frames, thus avoiding the problem of vanishing gradient in some traditional IoU losses.

3.4.2. Schematic Diagram of Wise-IoU and Its Main Workflow

Figure 11 shows the schematic diagram of Wise-IoU, in which the blue bounding box is the anchor box, the green bounding box is the target box, and the large black border is the minimum enclosing box, which is the smallest external rectangle that can completely enclose the blue anchor box and the green bounding box at the same time, and the role of the minimum enclosing box is to be used for calculating the geometric difference between the anchor box and the bounding box, so as to better optimise the positioning and size of the anchor box. The main workflow is as follows:

1. Matching of anchor and target frames and calculation of loss function

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{14}$$

Equation (14) calculates the ratio of the area of the overlap region of the green bounding box and the blue anchor box to the area of the union region to obtain the standard IoU values for the anchor box and the target box.

$$L_{IoU} = 1 - IOU \tag{15}$$

Equation (15) is the standard IoU loss function, which measures the difference between two boxes by calculating 1 minus the IoU value. The smaller the loss value, the smaller the difference between the anchor box and the target box, and the more accurately the model predicts. Therefore, the objective of the loss function is to minimise during the training process.

2. Calculate the distance of the centre point.

$$R_{DIoU} = \frac{\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2}{W_g^2 + H_g^2} \tag{16}$$

Equation (16) is used to calculate the distance between the centre points of the anchor box and the target box and normalise them with respect to their minimum bounding rectangles so as to optimise the bounding box regression in combination with other IoU information. Where x and y are the abscissa and ordinate values of the centre point of the

anchor box, x_{gt} and y_{gt} are the abscissa and ordinate values of the centre point of the target box, and W_g and H_g are the width and height of the minimum bounding box.

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{17}$$

Equation (17) is used to generate a weight $\boldsymbol{R_{WIoU}}$ that will be combined with the IoU loss to adjust the difference in distance between the anchor and target boxes. In this way, when the distance between the anchor box and the centre point of the target box is large, the loss value will increase, which helps the model to better optimise the positioning of the bounding box. The exponential function is used to smooth the influence of the distance measure, and the value of the exponential function decays rapidly as the distance increases, thus imposing a larger penalty on the anchor box farther away. And $\left(\boldsymbol{W_g^2 + H_g^2}\right)^*$ is used to normalise the distance between the centre points, ensuring that the distance is not affected by the size of the box. However, this value should not affect the update of the gradient during the optimisation of the bounding box, so it is marked with an asterisk to indicate that it will not participate in backpropagation.

3. Dynamic non-monotonic focusing mechanism

$$\beta = \frac{L_{IoU}^*}{L_{IoU}} \in [0, +\infty) \tag{18}$$

Equation (18) is to calculate the outlier measure $\boldsymbol{\beta}$, which represents the ratio between the expected IoU loss and the current IoU loss (i.e., the quality of the anchor box). $\boldsymbol{\beta}$ is used in the IoU improvement loss function to dynamically adjust the loss weight in order to better handle the regression problem between the anchor box and the target box.

$$r = \frac{\beta}{\delta + \alpha\beta^{-\delta}} \tag{19}$$

Equation (19) is to adjust the gradient gain value $\boldsymbol{r}$ by the outlier metric $\boldsymbol{\beta}$, where $\boldsymbol{\delta}$ is the hyperparameter controlling the gradient gain, and when $\boldsymbol{\beta} = \boldsymbol{\delta}$, the anchor box will get the maximum gradient gain.

4. Loss function formula

$$L_{WIoU=}\mathrm{r} \cdot \ L_{IoU} \cdot R_{WIoU} = \frac{\beta}{\delta + \alpha\beta^{-\delta}} \cdot (1 - IOU) \cdot \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{20}$$

The loss function formula of Wise-IoU is finally determined as Equation (20) under the influence of geometric factors such as the distance between the centre points of the anchor frame and the target frame and the overlapping area. By dynamically adjusting the gradient gain of the anchor frame, the harmful gradient caused by the low-quality anchor frame is avoided, and the competitiveness of the high-quality anchor frame is also reduced, which improves the model's attention to the ordinary quality anchor frame.

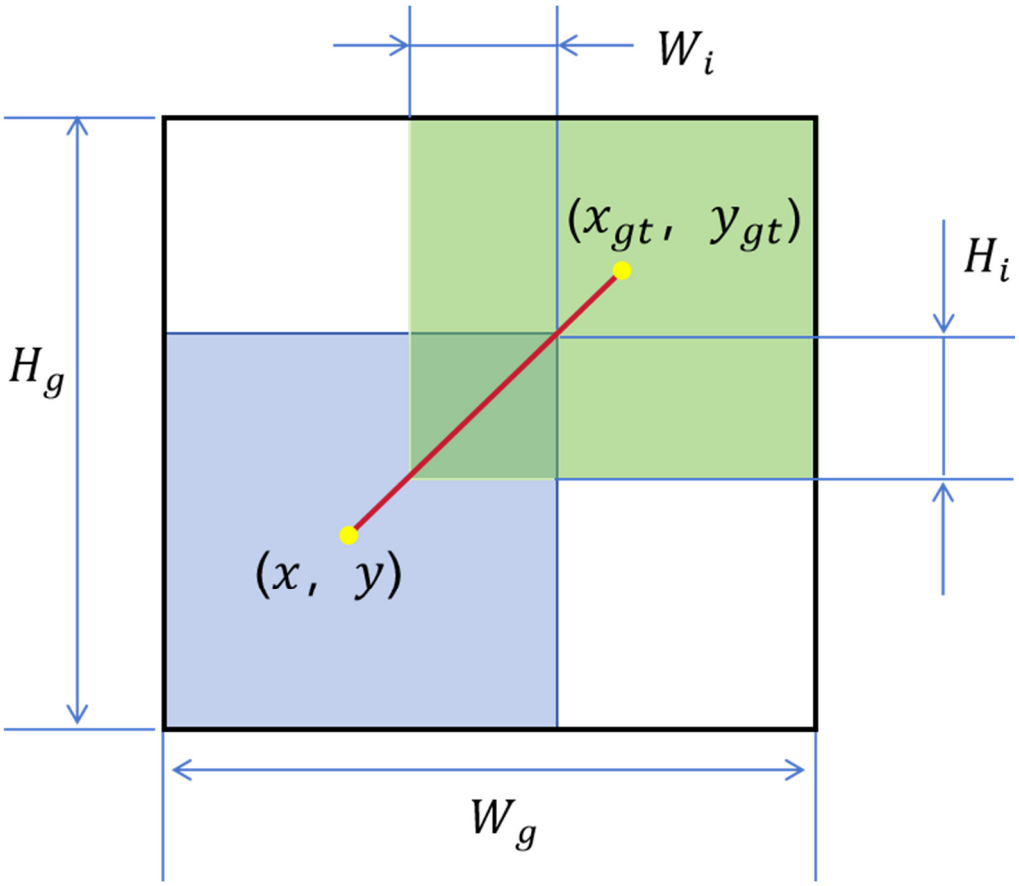

Figure 11. Schematic diagram of Wise-IoU.

3.5. Relevant Evaluation Indicators of YOLO Deep Learning Network Structure

Based on several important evaluation indexes of the YOLOv8 network model and the actual application, we evaluate the performance of the model from the following aspects. The definitions of the corresponding performance indexes and related formulas are as follows:

$$FPS = \frac{Total\ Frames\ Processed}{Total\ Time\ in\ Seconds} \tag{21}$$

FPS refers to the number of frames or images processed by the model per second, which directly reflects the response speed and processing ability of the model.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{22}$$

Recall is an indicator to evaluate the ability of the model to identify positive samples (TP), which indicates the proportion of samples that are correctly predicted as positive by the model among all samples that are actually positive.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{23}$$

Precision rate is an indicator to evaluate the prediction results of the classification model, which represents the proportion of samples that are actually positive among all the samples predicted to be positive by the model.

$$AP = \int_0^1 Percision(Recall)d(Recall) \tag{24}$$

$$mAP = \frac{\sum_{1}^{N} \int_{0}^{1} Percision(Recall) d(Recall)}{N} \quad (25)$$

Box mAp @ 50 (%) measures the average precision mean of the model's detection boxes across all categories when the IoU threshold is set to 0.50.

Mask mAp @ 50 (%) measures the average accuracy of the model's segmentation mask across all classes when the IoU threshold is set to 0.50

$$F1 - Score = 2 \times \frac{Percision \times Recall}{Percision + Recall} \quad (26)$$

BOX F1 is an evaluation index used in the target detection task, which combines the precision and recall of the detection box to measure the accuracy and completeness of the detection box.

Mask F1 is an evaluation index used in the instance segmentation task, which combines the precision and recall of the segmentation mask to measure the accuracy and completeness of the segmentation mask.

4. Results

4.1. Ablation Experiment and Result Analysis

In order to verify the effectiveness of the improved network based on YOLOv8n, SGW-YOLOv8n, in practical applications, we designed the following ablation experiments to verify it. Firstly, the improved modules are applied to the YOLOv8n network and named; that is, the network name obtained by applying the SPD-Conv module to the YOLOv8n network is S-YOLOv8n. The network name obtained by applying the SPD-Conv module and the GAM module to the YOLOv8n network is SG-YOLOv8n; the network name obtained by applying the SPD-Conv module, the GAM module, and the Wise-IoU loss function to the YOLOv8n network is SGW-YOLOv8n. Then, the model training equipment mentioned above is used to train the four different YOLO networks on the data set prepared in advance, in which the epoch, batch size and other parameters are the same, and the training results are shown in Table 1.

Table 1. Ablation Experiment.

Model	SPD-CONV	GAM	Wise-IoU	Box F1 (%)	Mask F1 (%)	FPS	Weight Size (MB)	Box mAP@50(%)				Mask mAP@50(%)			
								①	②	③	All	①	②	③	All
Yolov8n	x	x	x	70	70	48.96	6.44	92.9	63.3	60.5	72.2	92.7	63.7	61	72.5
S-Yolov8n	√	x	x	72	72	45.72	50.5	93.6	63.7	63.9	73.7	93.3	64.3	64.8	74.1
SG-Yolov8n	√	√	x	72	72	42.18	52.6	93.5	66.6	63.2	74.4	93.5	67.9	636	75
SGW-Yolov8n	√	√	√	74	74	44.37	52.3	93.2	67.7	66.8	75.9	93.1	68.1	66	75.7

As can be seen from Table 1, The training performance of the S-Yolov8n network, which only applied the SPD-CONV convolution module, showed a 2% improvement in both the Box and Mask F1 scores. For object detection of three categories at a threshold of 0.5, the average precision increased by 0.7%, 0.4%, and 3.4%, respectively. For instance, with the segmentation of three categories at a threshold of 0.5, the average precision increased by 0.6%, 0.6%, and 3.8%, respectively. The SG-Yolov8n network, which applied both the SPD-CONV convolution module and the GAM global attention mechanism module, showed the same improvement as the S-Yolov8n network in the Box and Mask F1 scores. For object detection of three categories at a threshold of 0.5, the average precision increased by 0.6%, 3.3%, and 2.7%, respectively. For instance, segmentation of three categories at a threshold of 0.5, the average precision increased by 0.8%, 4.2%, and 2.6%, respectively. When all three modules—SPD-CONV, GAM, and Wise-IoU—were applied to the network, the SGW-Yolov8n achieved a 4% improvement in both the Box and Mask F1 scores. For object detection of three categories at a threshold of 0.5, the average precision increased by

0.3%, 4.4%, and 6.3%, respectively. For instance, with the segmentation of three categories at a threshold of 0.5, the average precision increased by 0.4%, 4.4%, and 5%, respectively. The detection mAP also improved from the original 72.2% to a maximum of 75.9%, and the segmentation mAP increased from the original 72.5% to 75.7%. In terms of FPS, the image processing speed of the improved network model decreased slightly from the original 48.96 FPS to 44.37 FPS. Despite this slight decrease, the FPS remains above 30, which still meets the requirements for real-time inference. Regarding the size of the generated weight files, apart from the smaller weight of the original YOLOv8n model, the size of the weight files of the three improved models is around 50 MB. Since the model is deployed on an edge computing platform in actual production, the size of the weight files of the improved models will not affect deployment. Combining the above analyses and Table 1, it concludes that the improved network based on YOLOv8n, SGW-YOLOv8n has the best overall performance among the four network models mentioned above.

From Figure 12, it can be seen that when the eooch is 200, the minimum values of Train Box Loss and Val Box Loss of the SGW-YOLOv8n network model are smaller than those of the other three network models, and the model achieves the optimal effect of the model in advance before the end of the 200 rounds of training and thus ends the training, which indicates that the network model of SGW-YOLOv8n can converge quickly in fewer training rounds, indicating that its learning algorithm and model architecture are very effective and can quickly extract useful features and patterns from the training data; and has strong applicability This ability indicates that the model may have good adaptability and robustness to various datasets, especially on similar data distributions, and does not need too much adjustment to achieve satisfactory results.

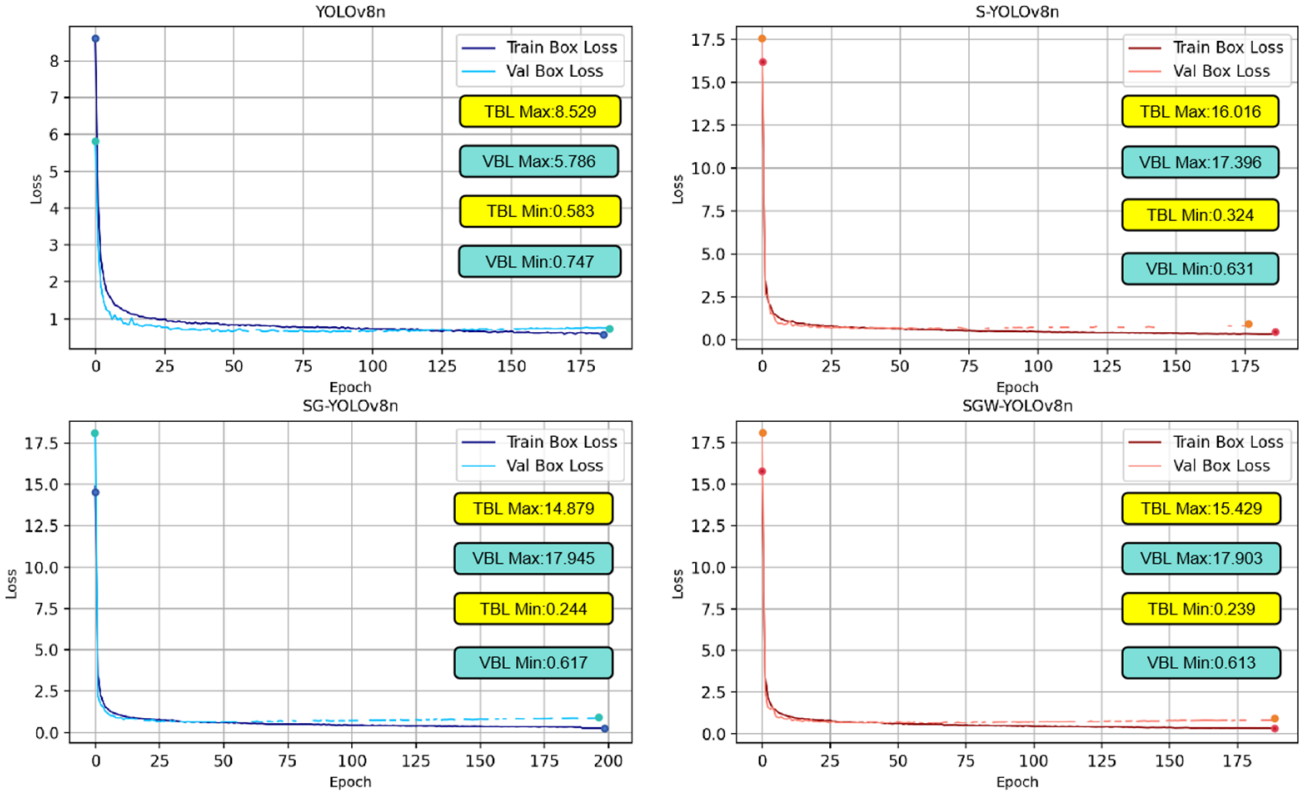

Figure 12. Loss function image of each model in the training process.

As shown in Figure 13, after training, the YOLOv8n model achieved a Box PR mAP@50 value of 0.722 and the Mask PR mAP@50 value of 0.725. The S-YOLOv8n model achieved a Box PR mAP@50 value of 0.737 and the Mask PR mAP@50 value of 0.741. The SG-YOLOv8n model achieved a Box PR mAP@50 value of 0.744 and the Mask PR mAP@50 value of 0.75.

The SGW-YOLOv8n model achieved a Box PR mAP@50 value of 0.759 and the Mask PR mAP@50 value of 0.757. Except for the YOLOv8n model, the other three models reached a stable performance stage around the 100th training epoch. Both their Box PR mAP@50 (Precision-Recall for Bounding Boxes) and the Mask PR mAP@50 (Precision-Recall for Segmentation Masks) metrics were higher than those of the YOLOv8n model, with the SGW-YOLOv8n achieving the highest values among all models. To further verify the practical effectiveness of SGW-YOLOv8n, an additional batch of image data was collected based on the previously prepared dataset. From this additional batch, three images were selected to conduct inference experiments using the optimal weight files (bast.pt) generated by the four aforementioned network models. The original graph and the inference result graph of each model occupy one row, respectively, and the experimental results are shown in Figure 14 below.

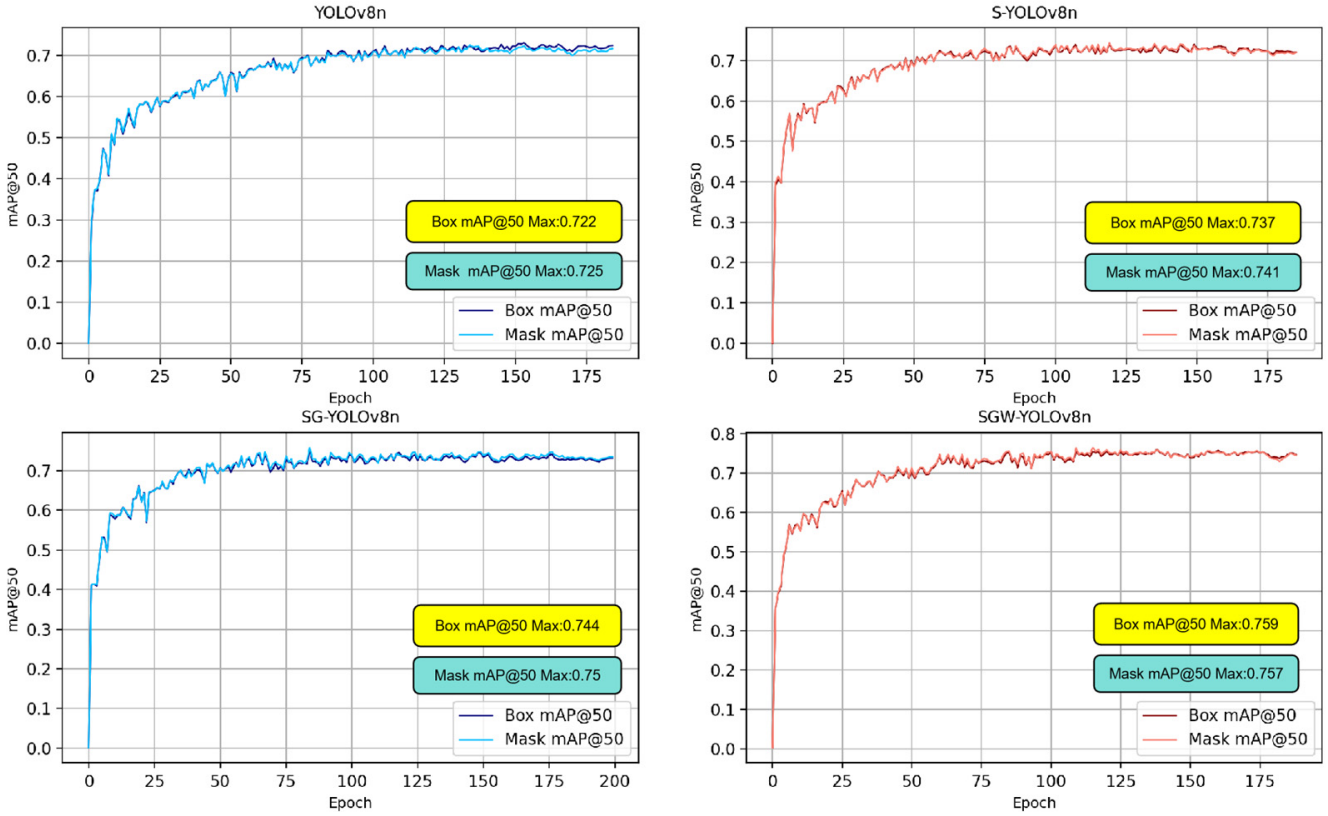

Figure 13. mAP change of each model in the training process.

It can be clearly seen from the second photo in Figure 14 that the image reasoning effect of the three improved network models is better than that of the original YOLOv8 model. The original YOLOv8 only detects two apples with the label category of Obvious, while the three improved network models not only improve the detection accuracy of the three label categories of apples, but the SGW-YOLOv8 model also detected two additional apples labelled Occluded, and the SGW-YOLOv8 model performed significantly better than the other three models in the three photos because it detected an additional apple labelled Occluded. These reasoning results show that the SGW-YOLOv8 model is significantly better than the original YOLOv8n network model for fruit detection in outdoor unstructured environments, especially for the detection of small occluded objects.

Figure 14. Actual reasoning effect of each model.

4.2. Comparison with Other Networks

In order to verify the differences between the SGW-YOLOv8n model and other different target detection and instance segmentation models, the following experiments were conducted in this study: the data sets prepared in the early stage of the experiment were trained with different models in Table 2 in turn, and the parameters such as the number of training rounds and batch size were set to be consistent, and the same hardware and system environment was used for training. The training results are shown in Table 2.

Table 2. Comparison of the effectiveness of SGW-YOLOv8 with other node detection and segmentation models.

Model	Precision (%)	Recall (%)	F1 Score (%)	Box mAP@50 (%)	Mask mAP@50 (%)	FPS	Weigth Size (MB)
Yolov5	67.2	68.1	67.65	68.4	72.5	32.28	14.9
Yolov6	69.3	66.2	67.71	67.9	_	30.94	8.28
Yolov7	66.8	63.5	65.11	64.8	_	28.63	71.3
Yolact	44.8	45.7	45.25	45.1	54.4	23.44	413
Mask R-CNN	43.2	42.6	42.90	43.2	53.1	26.57	483
SGW-Yolov8n	78	78	74	75.9	75.7	44.37	52.3

Table 2 provides the training results of different object detection and instance segmentation models (YOLOv5, YOLOv6, YOLOv7, YOLACT, Mask R-CNN, and SGW-YOLOv8n) including multiple evaluation indexes. Deploy the six different models mentioned above on a computer and set the same training parameters, such as the number of training epochs and batch size. Then, these six models will be used to train on the same dataset separately. In terms of precision and recall, SGW-YOLOv8n performs the best among all models, showing its strong ability to identify the correct target and reduce missed detection, which is also verified by the F1 score. The comprehensive performance of SGW-YOLOv8n is the highest among all models. In the Box mAP @ 50 (%) metric, YOLOv5 is slightly better, indicating that it performs best on bounding box localisation accuracy for object detection. For instance, in terms of segmentation, Mask R-CNN and YOLACT's Mask mAP @ 50 (%) are comparable, but Mask R-CNN is slightly better, indicating that it is slightly more accurate in instance segmentation. In terms of processing speed, SGW-YOLOv8n leads the way with 44.37 FPS, which makes it ideal for application scenarios requiring fast response. In addition, in terms of weight size, Mask R-CNN and YOLACT are much larger than other models, which reflects that they increase the model complexity and computational burden to deal with instance segmentation tasks, so Mask R-CNN and YOLACT are not suitable for deployment on edge computing platforms for real-time reasoning. Overall, although each model has its own characteristics and advantages, SGW-YOLOv8n is the most appropriate choice in terms of detection and segmentation accuracy, image processing speed and weight size.

4.3. Conclusions

The comparative experiments of the aforementioned different network models demonstrate that the improved SGW-YOLOv8n network model outperforms other models in terms of inference speed, detection accuracy, and segmentation accuracy. Additionally, its weight size meets the deployment requirements for edge computing platforms. In future research, we plan to test the model under different lighting conditions (morning, noon, and evening) to observe whether the SGW-YOLOv8n network model's generalisation capability and robustness in various outdoor environments are also effectively enhanced. The proposed network model provides a valuable reference for the research on fruit detection and segmentation in complex unstructured outdoor environments, promoting the development and application of orchard-picking robots.

Author Contributions: Conceptualization, T.W. and T.L.; methodology, T.W. and T.L.; validation, W.H. (Wenlei Huang), T.L. and Z.G.; visualization, T.W. and W.H. (Wenkai Han); data curation, T.W. and W.H. (Wenkai Han); form analysis, T.W. and W.H. (Wenlei Huang); investigation, T.W. and Z.G.; writing—original draft preparation, T.W. and T.L.; writing—review and editing, T.L., Z.M. and T.W.; funding acquisition, Z.M. and T.L.; supervision, Z.M. and T.L., project administration T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by BAAFS Youth Research Foundation [QNJJ202318], Innovation Capacity Building Project [KJCX20240502], International Science and Technology Cooperation Platform [2024-08].

Institutional Review Board Statement: Not applicable.

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

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

References

1. Zhang, Q.; Shi, F.; Abdullahi, N.M.; Shao, L.; Huo, X. An empirical study on spatial–temporal dynamics and influencing factors of apple production in China. *PLoS ONE* **2020**, *15*, e0240140. [CrossRef]
2. Shi, L.; Shi, G.; Qiu, H. General review of intelligent agriculture development in China. *China Agric. Econ. Rev.* **2019**, *11*, 39–51. [CrossRef]
3. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef]
4. Gené-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V.; Gregorio, E. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. *Comput. Electron. Agric.* **2020**, *169*, 105165. [CrossRef]
5. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural object detection with You Look Only Once (YOLO) algorithm: A bibliometric and systematic literature review. *arXiv* **2024**, arXiv:2401.10379. [CrossRef]
6. Swathi, Y.; Challa, M. YOLOv8: Advancements and innovations in object detection. In *International Conference on Smart Computing and Communication*; Springer Nature: Singapore, 2024; pp. 1–13.
7. Yang, S.; Wang, W.; Gao, S.; Deng, Z. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin transformer. *Comput. Electron. Agric.* **2023**, *215*, 108360. [CrossRef]
8. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Method for identifying litchi picking position based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [CrossRef]
9. Zhang, L.; Luo, P.; Ding, S.; Li, T.; Qin, K.; Mu, J. The grading detection model for fingered citron slices (Citrus medica 'fingered') based on YOLOv8-FCS. *Front. Plant Sci.* **2024**, *15*, 1411178. [CrossRef]
10. Wang, F.; Tang, Y.; Gong, Z.; Jiang, J.; Chen, Y.; Xu, Q.; Hu, P.; Zhu, H. A lightweight Yunnan Xiaomila detection and pose estimation based on improved YOLOv8. *Front. Plant Sci.* **2024**, *15*, 1421381. [CrossRef]
11. Wang, X.; Liu, J. Vegetable disease detection using an improved YOLOv8 algorithm in the greenhouse plant environment. *Sci. Rep.* **2024**, *14*, 4261. [CrossRef]
12. Zhou, S.; Zhou, H. Detection based on semantics and a detail infusion feature pyramid network and a coordinate adaptive spatial feature fusion mechanism remote sensing small object detector. *Remote Sens.* **2024**, *16*, 2416. [CrossRef]
13. 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]
14. Zhang, C.; Kang, F.; Wang, Y. An improved apple object detection method based on lightweight YOLOv4 in complex backgrounds. *Remote Sens.* **2022**, *14*, 4150. [CrossRef]
15. Ma, Z.; Dong, Y.; Xia, Y.; Xu, D.; Xu, F.; Chen, F. Wildlife real-time detection in complex forest scenes based on YOLOv5s deep learning network. *Remote Sens.* **2024**, *16*, 1350. [CrossRef]
16. Yuan, H.; Huang, K.; Ren, C.; Xiong, Y.; Duan, J.; Yang, Z. Pomelo tree detection method based on attention mechanism and cross-layer feature fusion. *Remote Sens.* **2022**, *14*, 3902. [CrossRef]
17. Zhu, Y.; Zhou, J.; Yang, Y.; Liu, L.; Liu, F.; Kong, W. Rapid target detection of fruit trees using UAV imaging and improved light YOLOv4 algorithm. *Remote Sens.* **2022**, *14*, 4324. [CrossRef]
18. Ni, J.; Zhu, S.; Tang, G.; Ke, C.; Wang, T. A small-object detection model based on improved YOLOv8s for UAV image scenarios. *Remote Sens.* **2024**, *16*, 2465. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhad, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; 2016; pp. 779–788.
20. Howard, A.G. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
21. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
22. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
24. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

25. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Cv, A.; Adam, H. Deeplabv3+: Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*; Ferrari, V., Hebert, M., Sminchisescu, C., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2018.
26. Lin, G.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2017**, *42*, 1228–1242.
27. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
28. Bochkovskiy, A. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
29. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2022; pp. 443–459.
30. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561.
31. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.