agronomy

MDPI

Article

Lightweight Detection of Broccoli Heads in Complex Field Environments Based on LBDC-YOLO

Zhiyu Zuo *, Sheng Gao, Haitao Peng, Yue Xue, Lvhua Han, Guoxin Ma and Hanping Mao

School of Agricultural Engineering, Jiangsu University, Zhenjiang 212013, China; 2222216041@stmail.ujs.edu.cn (S.G.); 2112116012@stmail.ujs.edu.cn (H.P.); 2222216030@stmail.ujs.edu.cn (Y.X.); hanlh@ujs.edu.cn (L.H.); mgx@ujs.edu.cn (G.M.); maohp@ujs.edu.cn (H.M.)
* Correspondence: zuozy@ujs.edu.cn

Abstract: Robotically selective broccoli harvesting requires precise lightweight detection models to efficiently detect broccoli heads. Therefore, this study introduces a lightweight and high-precision detection model named LBDC-YOLO (Lightweight Broccoli Detection in Complex Environment—You Look Only Once), based on the improved YOLOv8 (You Look Only Once, Version 8). The model incorporates the Slim-neck design paradigm based on GSConv to reduce computational complexity. Furthermore, Triplet Attention is integrated into the backbone network to capture cross-dimensional interactions between spatial and channel dimensions, enhancing the model's feature extraction capability under multiple interfering factors. The original neck network structure is replaced with a BiFPN (Bidirectional Feature Pyramid Network), optimizing the cross-layer connection structure, and employing weighted fusion methods for better integration of multi-scale features. The model undergoes training and testing on a dataset constructed in real field conditions, featuring broccoli images under various influencing factors. Experimental results demonstrate that LBDC-YOLO achieves an average detection accuracy of 94.44% for broccoli. Compared to the original YOLOv8n, LBDC-YOLO achieves a 32.1% reduction in computational complexity, a 47.8% decrease in parameters, a 44.4% reduction in model size, and a 0.47 percentage point accuracy improvement. When compared to models such as YOLOv5n, YOLOv5s, and YOLOv7-tiny, LBDC-YOLO exhibits higher detection accuracy and lower computational complexity, presenting clear advantages for broccoli detection tasks in complex field environments. The results of this study provide an accurate and lightweight method for the detection of broccoli heads in complex field environments. This work aims to inspire further research in precision agriculture and to advance knowledge in model-assisted agricultural practices.

Keywords: deep learning; lightweight; YOLO; broccoli head; object detection

Citation: Zuo, Z.; Gao, S.; Peng, H.; Xue, Y.; Han, L.; Ma, G.; Mao, H. Lightweight Detection of Broccoli Heads in Complex Field Environments Based on LBDC-YOLO. *Agronomy* **2024**, *14*, 2359. https://doi.org/10.3390/agronomy14102359

Academic Editors: Chenglin Wang, Lufeng Luo, Juntao Xiong and Xiangjun Zou

Received: 18 September 2024
Revised: 10 October 2024
Accepted: 11 October 2024
Published: 13 October 2024

1. Introduction

Broccoli (*Brassica oleracea* var. *italica* Plenck) is one of the most common vegetables in the world; the domestication process of broccoli probably began during Roman times [1]. Broccoli and its derived products offer a myriad of health benefits, encompassing antioxidant, anti-inflammatory, anti-cancer, anti-microbial, metabolic disorder regulatory, neuroprotective, and renoprotective effects [2]. Dietary vitamin C is important in an optimal diet, due to its antioxidant activity, which plays an important role in human nutrition [3]; broccoli can be a rich source of these vitamins in the diet. The expanded inflorescences of broccoli are an important feature. The main inflorescence morphometric characteristics, such as weight, height, diameter, shape, inflorescence curvature angle, and its stem diameter, are useful not only for analyzing varietal differences [1] and assessing plant quality [3], but also for non-contact broccoli grading [4].

The broccoli used for fresh consumption in traditional agriculture is harvested manually. The harvest must be selective due to the different stages of maturity that can be found within the same plantation [5]. In China, the largest producer of broccoli, agricultural

workers specializing in this task only cut broccoli at optimal maturity stages, while immature broccoli needs to be rechecked for several days or weeks, making it a labor-intensive industry. However, in recent years, the aging population and the loss of rural labor have brought challenges to the broccoli industry due to labor shortages and rising labor costs [6]. Therefore, the development of broccoli selective harvesting robots capable of automation and unmanned operation is necessary. The target detection system is crucial for broccoli selective harvesting robots, as the accuracy and speed of target detection directly affect the subsequent harvesting efficiency of the robot. Developing a high-speed and accurate target detection method for broccoli is necessary.

Target detection methods used for fruit and vegetable picking include traditional image processing methods and deep learning-based methods. Traditional image processing methods require the use of manually designed features, including color features [7], geometric features [8], texture features [9], and multi-feature fusion features [10], which have been used to detect fruit and vegetable targets in a traditional image processing manner. Pieter M. Blok et al. [11] used texture- and color-based image segmentation techniques to separate broccoli heads from the background, achieving a sensitivity score of 91.2% in a ground truth dataset containing 200 images and 228 broccoli heads. However, its detection effectiveness is susceptible to lighting and camera focus depth, and the average processing time per image is 0.27 s, which is relatively slow.

With the improvement of computer performance, deep learning technology has rapidly developed. Deep learning technology has made many contributions to agriculture. Dhaka et al. [12] successfully demonstrated the role of integrating cutting-edge technologies, such as the Internet of Things (IoT) and deep learning, in automating agriculture through a literature review. The authors believe that integrating the IoT and deep learning models may help develop tools to assist farmers in improving the productivity and quality of crop products. Kundu et al. [13] implemented a framework based on deep learning models to monitor, classify, and predict the severity of corn crop diseases, and estimate crop losses. Additionally, they collected datasets containing healthy corn crops and crops infected with various diseases such as TLB and rust. The model's efficiency in extracting relevant features with fewer parameters, shorter training time, and high accuracy highlight its importance as an auxiliary tool for plant pathology experts. Gangwar et al. [14] developed a deep learning model called "Swift-Lite-CvT" that accurately classifies diseases based on images of tomato leaves. The model achieves the goals of minimizing storage space, reducing training time, and optimizing certain parameters while ensuring model accuracy.

Compared to naked-eye detection and traditional image classification, deep learning-based methods have higher accuracy [15]. Various deep learning neural networks have been used in target detection tasks in agriculture [16,17]. Deep learning-based object detection algorithms can be roughly divided into two-stage and one-stage object detection algorithms. Yan Jianwei et al. [18] used an improved algorithm based on the two-stage algorithm Faster R-CNN to identify prickly pear fruits in natural environments. This algorithm showed high accuracy and recall rates when tested on a dataset containing prickly pear fruits in 11 different states. Although two-stage algorithms often perform slightly better than one-stage algorithms in terms of accuracy, one-stage algorithms have advantages in terms of detection speed and computational requirements. A YOLOv5s model using a simplified channel pruning method was used for apple fruit detection before thinning [19], which maintained good detection performance in environments with varying lighting and clustered fruits. The detection time per frame was only 8 ms, indicating a much faster detection speed than Faster R-CNN. Yolo-CBAM [20] combined YOLOv5 with an attention mechanism and was deployed on a mobile platform for testing after employing a multi-scale training approach. The testing results showed that it is highly competent for real-time field detection tasks of Solanum rostratum Dunal seedlings.

Due to the hardware limitations of mobile platforms, broccoli selective harvesting machines often face computational challenges when deploying target detection algorithms. To address the challenge of insufficient computational power on mobile platforms, many

researchers have been devoted to lightweight models for target detection. Les-YOLO [21] replaces the backbone of YOLOv4-tiny with LesNet and adds the SE (squeeze-and-excitation) attention mechanism. This model retains a single-scale detection head for small object detection and reduces parameters, demonstrating high accuracy, speed, and low computation complexity in pinecone detection tasks compared to other mainstream algorithm models. Li Hui peng et al. [22] used the YOLO-grape model for real-time detection of fresh grapes in complex backgrounds, which had lower recognition time and model size compared to SSD300 and Faster R-CNN with ResNet31 as the backbone.

In recent years, studies related to broccoli recognition have reported high accuracy rates. Blok et al. [11] used a closed box to collect broccoli head images unaffected by ambient light and used texture-based image segmentation techniques to locate the broccoli heads, achieving a sensitivity score of 91.2%. However, limited by the generalization ability of the algorithm, its recall rate was low. In subsequent studies, they introduced deep learning algorithms, achieving better image generalization across multiple broccoli varieties [23]. Kusumam et al. [24] used a 3D vision-based method to detect broccoli heads, achieving a high accuracy rate of 95.2%, and conducted cross-validation on datasets from the UK and Spain, demonstrating good generalization performance. García-Manso et al. [5] used Faster R-CNN to detect and classify broccoli heads. Their data were captured on-site in plantations with uncontrolled natural lighting, and reportedly, the algorithm could accurately locate 97% of the broccoli in the test set. The authors reported that the algorithm's generalization ability could be further improved if images taken under different lighting conditions at different times of the day were included. Chengquan Zhou et al. [4] used an improved deep convolutional neural network to segment broccoli pixels in field environments and grade broccoli according to their new standards. This method demonstrated good robustness to light intensity and noise. Through experiments, the authors believe that their method can contribute to improving broccoli breeding and vegetable trade. Kang Shuo and others [25] proposed a machine vision method based on a semantic segmentation model to detect broccoli heads, determine maturity categories, and accurately locate targets suitable for harvesting. This method integrates an enhanced DeepLabV3+ network model with a self-designed grading module, achieving a high pixel accuracy of 98.56% and an average category prediction accuracy of 70.93%.

Unfortunately, existing research on broccoli detection has not focused on the computational complexity of the recognition algorithms used. Considering our future goal of developing a low-cost, all-weather selective broccoli harvesting machine, in practice, the broccoli harvester needs to simultaneously harvest multiple rows of broccoli, processing multiple rows of image data, which can put significant computational pressure on the mobile platform's hardware due to its limited processing power. Therefore, for achieving our future goals, it is important to focus on the lightweight nature of the recognition algorithms. Moreover, accurate detection of broccoli heads still faces challenges in the complex field environment. In open-air broccoli fields with dense planting, during the harvesting period, the leaves of mature and semi-mature broccoli plants are long and lush, causing high occlusion and uneven lighting on the broccoli heads. Additionally, it is nearly impossible to distinguish between broccoli heads and leaves based on color features, especially under insufficient lighting conditions. Furthermore, to match the high-speed actions of mechanical broccoli harvesting, the detection model also needs to have real-time detection capabilities.

To address these challenges, this paper introduces LBDC-YOLO (Lightweight Broccoli Detection in Complex Environment—You Look Only Once), a lightweight broccoli detection model designed to address challenges in complex field environments. By applying the Slim-neck design to the YOLOv8 algorithm, the model reduces computational load and parameter count. Triplet Attention is integrated into the backbone network, and BiFPN replaces the original neck structure to enhance multi-scale feature integration. A diverse dataset was created to improve image variety. Key contributions include a mixed dataset of broccoli heads under a variety of conditions and a detection model that achieves a

detection accuracy of 94.44% while significantly reducing complexity and parameters and demonstrating robustness to light, occlusion, shadows, and water droplets.

2. Materials and Methods

2.1. Data Acquisition

2.1.1. Raw Image Data Acquisition

The broccoli data used in this study were obtained from open-field broccoli farms located in Xiangshui County, Yancheng City, Jiangsu Province, China. The pH value of the soil in the planting area is 7.2–8.0, the content of organic matter is ≥1.6%, and the quick-acting potassium is ≥140 mg/kg. The planting area is located in the climatic zone of the transition from north subtropical zone to south warm zone, which belongs to the typical oceanic monsoon climatic zone, with an average annual precipitation of 922.1 mm, and an average annual sunshine of 2346.8 h. The broccoli varieties used were "Xiyingmen" (Sakata Seedling (Suzhou) Corporation, Suzhou, China) and "Lvsheng" (Huihe Seed Industry Corporation, Shanghai, China); all broccoli seedlings came from Shumei Agricultural Technology Corporation (Yancheng, China). Broccoli plants were manually planted on 10 August. The plants were planted on a 100 m long and 1 m wide ridge, with two rows planted on one ridge, 0.4 m between plants, 0.6 m between rows, and 1.4 m between ridges. Fertilizer was applied about 15 days after planting, 10 kg~15 kg urea per mu; fertilizer was applied once after the buds appeared, 20 kg~25 kg compound fertilizer with ≥45% total nutrient content of nitrogen, phosphorus and potassium per mu. Artificial channel irrigation was used to keep the soil dry and wet alternately, and water was drained in time after rain. Watering was controlled for 7 days before harvesting, and the plants were watered moderately after fertilization.

The image acquisition devices used were the Intel Realsense D435i camera (Intel Corporation, Delaware, DE, United States) and Huawei Nova 8 Pro mobile phone (Huawei Corporation, Shenzhen, China), placed at a height of 100–110 cm above the ground and capturing images from directly above the broccoli.

The heads of broccoli in the field environment are easily affected by factors such as lighting, occlusion, shadows, and water droplets. Broccoli heads affected by different lighting conditions can be observed as follows: on sunny days, sunlight shines directly (Figure 1a), resulting in strong illumination and distinct shadow boundaries; on cloudy and rainy days, sunlight is partially or fully blocked by clouds, resulting in soft and even lighting (Figure 1b). Broccoli heads occluded by leaves have fewer visible features as the level of occlusion increases (Figure 1c). Broccoli heads affected by shadows occur when shadows cast by leaves intersect with sunlight passing through branches, creating partial shadows on the broccoli heads (Figure 1d). In densely planted areas, broccoli leaves surround the heads, creating a wall-like enclosure where direct sunlight cannot reach, resulting in shadowy surroundings Figure 1e). Broccoli heads affected by water droplets exhibit unique characteristics compared to dry ones, as raindrops hang and attach to the broccoli heads (Figure 1f).

During data collection, various influencing factors were taken into consideration. Images were captured under three different weather conditions, sunny, cloudy, and rainy, and during three different time periods, early morning (6:00–8:00), midday (10:00–14:00), and dusk (17:30–18:30), with sunrise at 6:02 and sunset at 17:37. In total, 839 valid images were collected, providing rich and comprehensive image data.

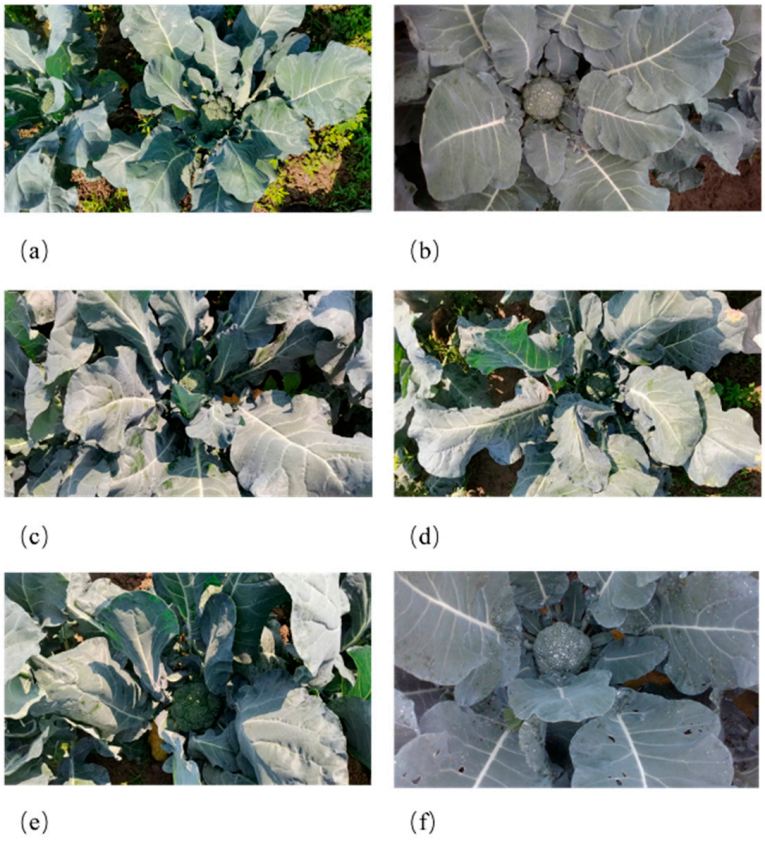

Figure 1. Different conditions of broccoli heads. (**a**) Broccoli heads in direct sunlight; (**b**) broccoli heads in soft and even light; (**c**) occluded broccoli heads; (**d**) broccoli heads in partial shadows; (**e**) broccoli heads in complete shadow; (**f**) wet broccoli heads.

2.1.2. Data Preprocessing and Dataset Construction

In this study, the Labelme (v4.6.0) [26] software was used to manually annotate the collected original valid images using rectangular bounding boxes. The bounding boxes were aligned with the visible parts of the broccoli heads. The label files were outputted in JSON format and we converted it to TXT format with the codes we made with Python. The 839 images obtained were divided into training set, validation set, and test set, which accounted for 80%, 10%, and 10% of the total number of images, respectively. Data augmentation was performed on the training set using two methods: brightness adjustment and noise addition, resulting in an expanded training set of 2013 images. The images and labels were stored in the YOLO dataset format.

2.2. LBDC-YOLO Detection Model

This study is based on the YOLOv8 network architecture but makes improvements specifically for the characteristics of broccoli while considering the needs of the detection task. Thus, the LBDC-YOLO model is proposed. Firstly, considering the limited computational power when deploying the model on mobile platforms, the Slim-neck design paradigm is applied to the model's neck network to balance the model's complexity and accuracy. Secondly, the Triplet Attention mechanism is embedded into the model's backbone, specifically into the C2f-Triplet module in the diagram. The notation H × W × C in the diagram represents Height × Width × Channels, which indicates the size of the input or output tensors in each process. This captures cross-dimensional interaction information between spatial dimensions and channel dimensions. This enables the model to better focus its attention on the target, improving the model's recognition and localization capabilities at a lower computational cost, compensating for and further enhancing the loss of model accuracy introduced by Slim-neck. Finally, the PAN-FPN structure is replaced with BiFPN

to obtain a more streamlined architecture and allow features at more important scales to receive more attention. The network structure of LBDC-YOLO is illustrated in Figure 2.

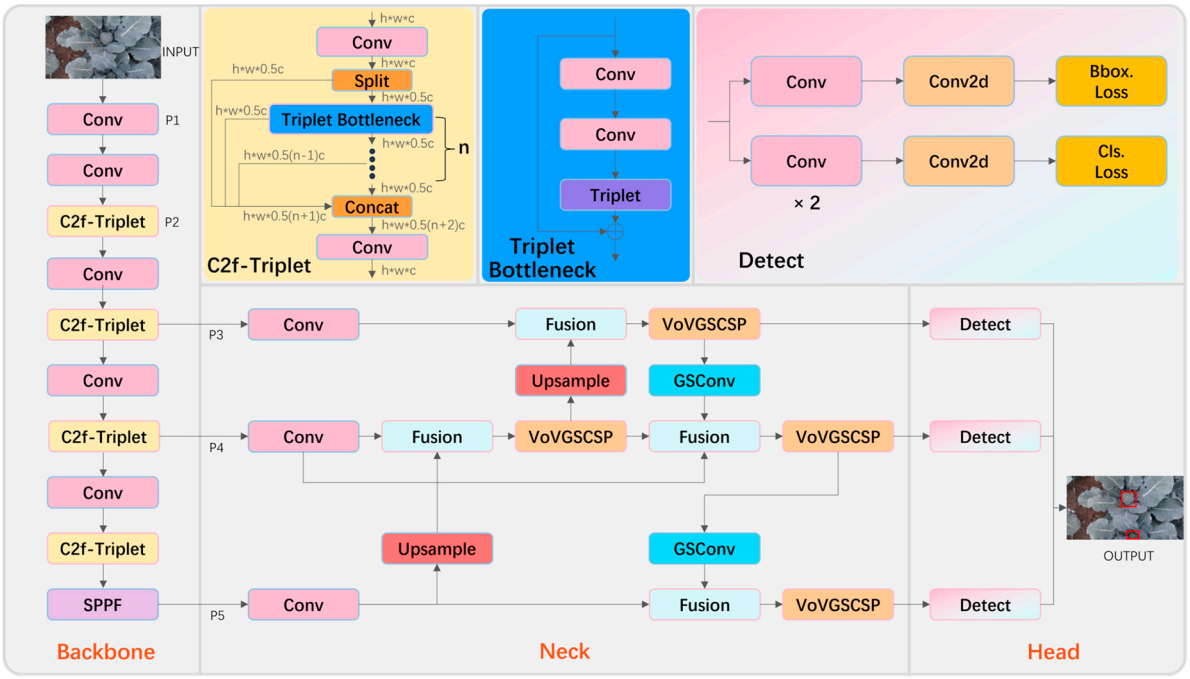

Figure 2. Network structure of LBDC-YOLO. Red rectangles in the output image indicate broccoli heads detected by the model. The different colored boxes in the figure represent modules with different functions.

2.2.1. YOLOv8

YOLOv8 [27] is one of the state-of-the-art object detection algorithms, which utilizes the one-stage detection method of YOLO [28]. It consists of a backbone, neck, and head structure. The main innovation of YOLOv8 lies in the adoption of the C2f module for the backbone and neck, aiming to enhance the network's depth and receptive field, thereby improving the feature extraction capabilities. The C2f module in YOLOv8 is inspired by the C3 module in YOLOv5 [29] and the E-ELAN (Extended Efficient Layer Aggregation Networks) concept in YOLOv7 [30]. Compared to the C3 module in YOLOv5, the C2f module introduces more residual connections, allowing YOLOv8 to obtain richer gradient flow information while maintaining a lightweight design. The neck component of YOLOv8 utilizes the PAN-FPN (Path Aggregation Network—Feature Pyramid Network) structure, which combines multi-scale features through both top-down and bottom-up fusion strategies. By integrating spatial information from shallow features and semantic information from deep features, the PAN-FPN structure ensures excellent detection capabilities for objects of various scales. Therefore, we consider YOLOv8 to be a suitable base model for broccoli detection tasks.

2.2.2. Slim-Neck

When deploying object detection models on mobile platforms such as broccoli selective harvesting machines, there can be limited hardware resources and insufficient computational power. These conditions may severely degrade the efficiency of the model or even render it inoperable. To address these challenges while maintaining high detection accuracy, it is important to minimize the computational complexity of the model. In this study, we applied the Slim-neck design paradigm and utilized GSConv to replace the convolution modules in the neck section and the convolution in the C2f module of the model. This effectively reduces the computational complexity of the model [31].

GSConv can reduce computational costs while preserving the feature extraction and fusion capabilities of standard convolution as much as possible. This contributes to the lightweight design of the model and improves its performance when deployed on edge devices with limited computational power. Figure 3 illustrates the principle of GSConv, which first applies Standard Convolution (SC) to the input image. Then, Depth-wise Separable Convolution (DSC) is performed on the convolution result, and the two parts are simply added together. The information generated by SC and DSC is fused using the shuffle operation. In general, the time complexity of a convolutional operation is defined by FLOPs (Floating-Point Operations). Therefore, the time complexities of SC, DSC, and GSConv (excluding biases) are, respectively, as follows [31]:

$$Time_{SC} \sim O(W \times H \times K_1 \times K_2 \times C_1 \times C_2) \quad (1)$$

$$Time_{DSC} \sim O(W \times H \times K_1 \times K_2 \times 1 \times C_2) \quad (2)$$

$$Time_{GSConv} \sim O\left[W \times H \times K_1 \times K_2 \times \left(\frac{C_2}{2}\right) \times (C_1 + 1)\right] \quad (3)$$

where W represents the width of the output feature map; H represents the height of the output feature map; $K_1 \times K_2$ represents the size of the convolutional kernel; C_1 represents the number of channels in each convolutional kernel; C_2 represents the number of channels in the output feature map; W, H, K_1, K_2, C_1, and C_2 are all positive integers. It is evident that when C_1 is greater than 1, the time complexity of DSC is the lowest, while GSConv falls between SC and DSC. In the calculation process, SC maximally preserves the hidden connections among each channel, while DSC completely cuts off these connections in exchange for significantly reduced complexity. GSConv, which combines SC and DSC, can preserve these connections to the greatest extent with lower time complexity, thus maintaining a higher capacity for feature extraction while reducing computational cost.

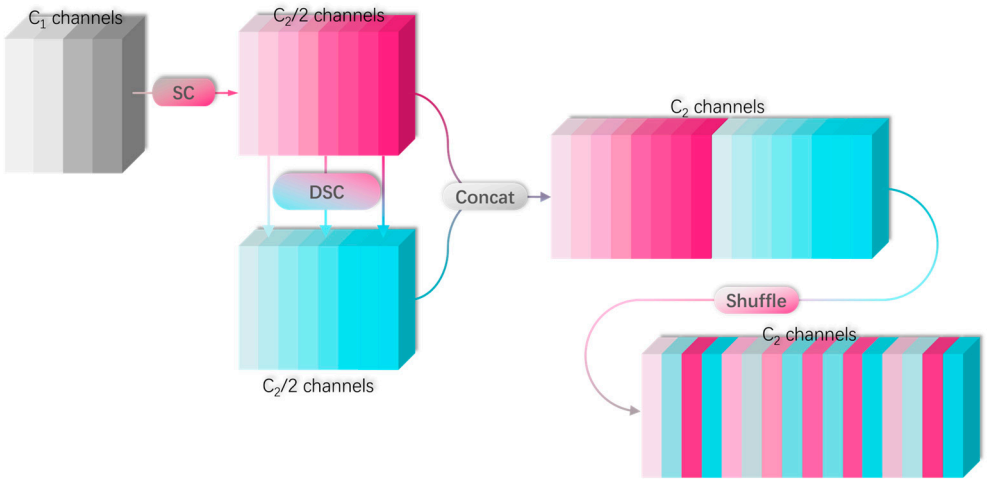

Figure 3. Schematic diagram of GSConv principle.

2.2.3. Triplet

In densely planted open-air broccoli fields, the leaves of broccoli plants in the harvesting period are long and lush. The florets of each broccoli plant exhibit varying degrees of occlusion and uneven lighting. Furthermore, the color of the broccoli florets closely resembles that of the background leaves, with the similarity becoming more pronounced under insufficient lighting conditions. All these present challenges to the feature extraction and target localization abilities of the object detection model. Therefore, this study adopts a strategy of incorporating an attention mechanism to enhance the feature extraction and target localization capabilities of the model.

The Triplet Attention mechanism captures cross-dimensional interactions using an innovative three-branch structure to calculate attention weights [32]. This is a cost-effective yet effective attention mechanism. Traditional methods for computing channel attention involve calculating a weight and then uniformly scaling the feature map using that weight. However, this method has a drawback: typically, to calculate the weight of the channels, the input tensor is spatially downsized to a single pixel through global average pooling, resulting in significant loss of spatial information. The interdependence between the channel dimension and spatial dimension is also absent when computing attention on a per-pixel channel. Although a later proposed CBAM (Convolutional Block Attention Module) [33] model based on spatial and channel dimensions attempted to alleviate spatial interdependence issue, there still remains a problem that channel attention and spatial attention are separate and their computations are independent. Based on establishing spatial attention, Misra et al. [32] introduced the concept of cross-dimension interaction, which captures the interaction between the spatial dimension and the input tensor's channel dimension, solving this problem.

The Triplet structure used in this study is shown in Figure 4. The *Z-pool* layer is used to capture cross-dimensional dependencies between the first and second dimensions of the input tensor to reduce the size of the tensor's zeroth dimension to 2. Its formula can be represented as follows:

$$Z\text{-}pool(\chi) = [MaxPool_{0d}(\chi), AvgPool_{0d}(\chi)] \tag{4}$$

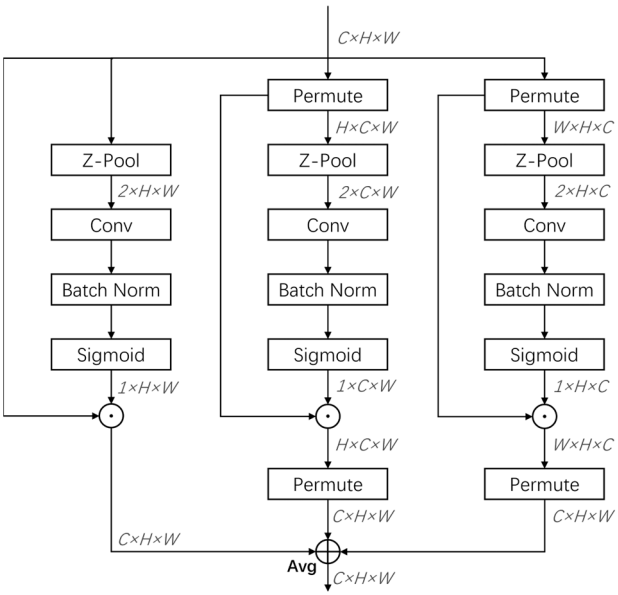

Figure 4. Structure of Triplet.

In the equation, χ represents the input tensor; the subscript $0d$ represents the zeroth dimension; *MaxPool* represents the max pooling operation; and *AvgPool* represents the average pooling operation. As such, the *Z-pool* layer can capture rich cross-dimensional interaction information between the first and second dimensions while reducing the tensor depth and compressing computational cost.

The first branch is the spatial attention computation branch. The input tensor is first passed through the *Z-pool* layer, where the channel number of the input tensor is reduced to 2, resulting in a simplified tensor shape of $2 \times H \times W$. It then goes through standard convolutional layers and batch normalization layers, resulting in an intermediate tensor of $1 \times H \times W$. This tensor is then passed through a sigmoid activation layer to obtain spatial attention weights of shape $1 \times H \times W$, which are applied to the input tensor.

The resulting tensor maintains the original shape of the input tensor and contains spatial attention information.

The second branch captures cross-dimensional interaction information between the channel dimension C and the spatial dimension W. The input tensor is first rotated counterclockwise 90° along the w-axis, resulting in a tensor shape of $H \times C \times W$. It then passes through the *Z-pool* layer, simplifying it to $2 \times C \times W$. This is followed by standard convolutional layers and batch normalization layers, outputting an intermediate tensor of $1 \times C \times W$. The tensor then undergoes a sigmoid activation layer to obtain attention weights. These weights are applied to the rotated tensor of shape $H \times C \times W$, which is then rotated clockwise 90° along the w-axis to maintain the same shape as the input tensor.

The cross-dimensional interaction between the channel dimension C and the spatial dimension H is captured by the third branch. The input tensor is rotated counterclockwise by 90 degrees along the H axis, resulting in a tensor of shape $W \times H \times C$. Then, this tensor goes through standard convolutional layers and batch normalization layers, and the output is passed through a sigmoid activation layer to generate attention weights of shape $1 \times H \times C$. These attention weights are applied to the rotated tensor $W \times H \times C$, which is then restored to its original shape by rotating clockwise 90 degrees along the H axis.

Finally, the three branches obtain three different $C \times H \times W$ tensors, which are aggregated using average pooling, resulting in a tensor that contains cross-dimensional interaction information across the three dimensions.

2.2.4. BiFPN

Broccoli in field environments often has high occlusion rates, and the degree of target occlusion varies with significant variations in size. In this study, the neck network structure of the model is improved to enhance its ability to fuse features of different scales. YOLOv8 adopts PAN-FPN for multi-scale feature fusion. PAN-FPN has a bidirectional feature network structure that fuses features from deep to shallow and from shallow to deep, effectively integrating multi-scale features. However, it involves a large computation cost due to the numerous cross-level connections and does not consider the unequal contributions of input features with different resolutions to the fused output features. Therefore, this study replaces the original PAN-FPN feature fusion network in YOLOv8 with the Weighted Bi-directional Feature Pyramid Network (BiFPN) [34] to address these issues.

The BiFPN structure is shown in Figure 5. The BiFPN structure enables efficient bidirectional cross-scale connections and weighted feature fusion. It has two main characteristics:

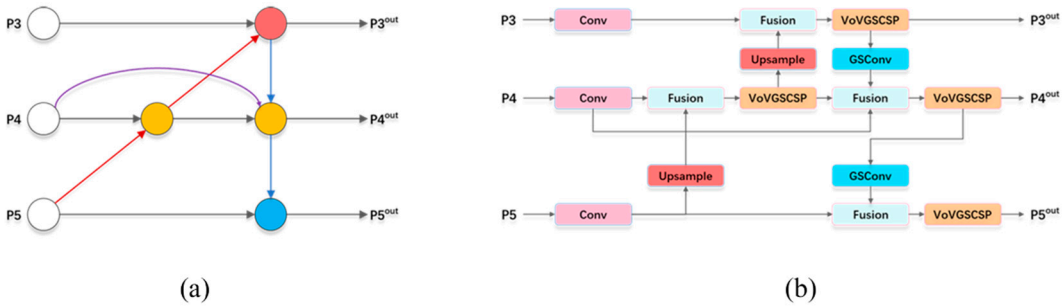

Figure 5. Structure of BiFPN. (**a**) Simplified structure of BiFPN; (**b**) BiFPN structure in LBDC-YOLO.

(a) Optimized cross-scale connection structure

In a bidirectional feature network, nodes without feature fusion and only one input edge contribute relatively less to the fusion of different features [34]. Therefore, based on PAN-FPN, BiFPN removes nodes with only one input edge to achieve a more streamlined network structure. In addition, BiFPN also adds skip connections between input and output nodes of the same scale, which enables the fusion of more features without significantly increasing computational cost.

(b) Weighted feature fusion

Since features from different scales have different resolutions, their contributions to the output features are usually unequal. BiFPN uses a weighted fusion method, which adds an additional weight for each input and lets the network learn the importance of each input characteristic to address this issue. Fast normalized fusion:

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i \tag{5}$$

In the equation, O represents the output; I represents the input; and ϵ represents the learning rate.

Among them, $w_i \geq 0$ is ensured by applying the ReLU activation function after each w_i. The learning rate is set to ϵ = 0.0001 to avoid numerical instability. This way, the value of each weight is constrained between 0 and 1, and, compared to the commonly used normalization operation softmax, it is much more computationally efficient.

2.3. The Environment of the Experiment

2.3.1. Model Operating Environment

The operating environment of the model in this experiment is shown in Table 1.

Table 1. Operating environment.

Configuration	Parameters
CPU	Intel Xeon Silver 4110 2.1 GHz (Intel Corporation, Delaware, DE, USA)
GPU	NVIDIA Quadro P4000 (NVIDIA Corporation, Delaware, DE, USA)
DRAM	8 GB
RAM	64 GB
Operating system	Windows10 professional workstation edition
Python version	3.9.16
PyTorch version	1.13.1
Torchvision version	0.14.1
CUDA version	11.7

In this paper, all experiments were conducted with a predetermined number of 500 epochs, a batch size of 8, an input resolution of 1280 × 1280, and a patience value of 50 epochs.

2.3.2. Evaluation Metrics

In this paper, P (precision), R (recall), and AP (Average Precision)$_{0.5}$ and $AP_{0.5–0.95}$ are used to evaluate the detection performance of the model.

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

Among them, TP (True Positive) represents the number of true positive detections, which is the number of positive targets correctly detected as positive; FP (False Positive) represents the number of false positive detections, which is the number of negative targets incorrectly detected as positive; FN (False Negative) represents the number of false negative detections, which is the number of positive targets incorrectly detected as negative. P can be used to measure the detection accuracy of the model, and R can be used to measure the effectiveness of the detection model. $AP_{0.5}$ and $AP_{0.5–0.95}$ can reflect the detection performance of the model. The concept of IoU (Intersection over Union) needs to be

introduced when calculating. In object detection, IoU is used to measure the accuracy of the detection boxes, and its calculation formula is as follows:

$$IoU = \frac{A \cap B}{A \cup B} \tag{8}$$

Here, *IoU* is the value of IoU; *A* represents the prediction box, and *B* represents the ground truth box.

When computing, set the IoU threshold to evaluate each detection box. Those that exceed the threshold with any ground truth box are considered as True Positives, those that do not reach the threshold are considered as False Positives, and any ground truth boxes that do not reach the threshold with any detection box are considered as False Negatives.

$AP_{0.5}$ refers to the Average Precision calculated with an IoU threshold set at 0.5, while $AP_{0.5–0.95}$ refers to the mean *AP* calculated at different IoU thresholds (from 0.5 to 0.95, in steps of 0.05).

In addition, we also use model size, number of parameters, Floating-Point Operations (FLOPs), and FPS (Frames Per Second) to evaluate the complexity of the model. Model size mainly affects the ROM occupation, the number of parameters can be used to measure the complexity of the model, and FLOPs represents the computational cost required for one detection by the model. FPS evaluates the detection speed of the model, indicating the number of images the model can process per second.

3. Results

3.1. Different Attention Mechanisms

After introducing the Slim-neck structure into the neck network of YOLOv8, the model's accuracy slightly decreased. Therefore, an attention mechanism was added to the main backbone network of the model to enhance its feature extraction capability and compensate for the loss in accuracy. To demonstrate the effectiveness of the Triplet Attention mechanism used in this paper, the Triplet, SE (Squeeze-and-Excitation), CA (Channel Attention), CBAM (Convolutional Block Attention Module), and ECA (Efficient Channel Attention) mechanisms were added individually at the same positions in the baseline model, to compare test results with YOLOv8-Slim-neck. The performance of models using different attentional mechanisms is shown in Table 2.

Table 2. Performance comparison of different attentional mechanisms.

Model	*P*/%	*R*/%	$AP_{0.5–0.95}$/%	Params/M	FLOPs/G
YOLOv8n-Slim-neck	96.50	97.70	93.69	2.796	7.3
+SE	96.19	97.03	94.26	2.799	7.3
+CBAM	97.40	96.16	93.98	2.824	7.3
+CA	96.78	97.00	94.33	2.799	7.3
+ECA	96.22	96.77	94.08	2.796	7.3
+Triplet	96.60	97.08	94.37	2.798	7.3

The incorporation of diverse attention mechanisms is evident in enhancing the model's detection accuracy. In terms of the magnitude of accuracy improvement, Triplet is undoubtedly the best choice. Considering the impact of attention mechanisms on computational complexity, Triplet remains the optimal choice as it yielded a 0.68 percentage point *AP* improvement with an increase of only about two thousand parameters. The inclusion of the Triplet Attention mechanism allows the model to focus more on the relevant key areas related to the target when processing input images, thereby enhancing the model's feature extraction capability and ultimately improving its detection ability.

3.2. Experiments on the Model

3.2.1. Model Training Results

When the model was trained, the input image size was set to 1280 × 1280, the batch size was set to 8, the initial learning rate was set to 0.001, and the decay index was set to 0.937. After 360 epochs of training, the model converged, and the curve of the change of $AP_{0.5–0.95}$ is shown in Figure 6.

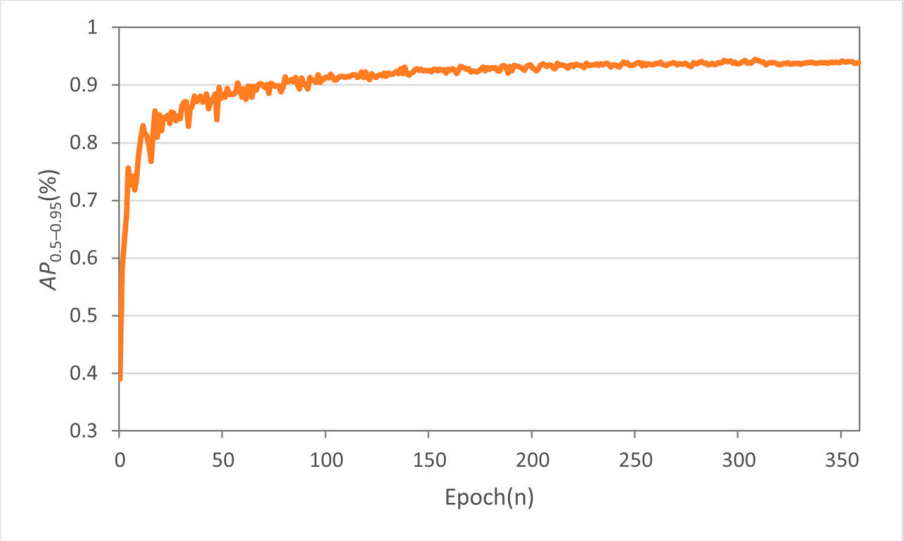

Figure 6. $AP_{0.5–0.95}$ curve of LBDC-YOLO.

Figure 6 illustrates the dynamic variations in $AP_{0.5–0.95}$ across the entire training process of the model. During the initial 100 rounds of training, the model's $AP_{0.5–0.95}$ rapidly fluctuated. After 100 rounds, the model began to slowly converge and approach stability, with $AP_{0.5–0.95}$ eventually stabilizing at around 95%, indicating that the training of the model was successful.

The training results of YOLOv8n and LBDC-YOLO are shown in Table 3. Compared to the baseline model, LBDC-YOLO achieved an improvement of 0.47 percentage points in $AP_{0.5–0.95}$, while reducing the model size by 44.4%. The parameter count was decreased by 47.8% and the computational load was reduced by 32.1%. The lightweight nature of the improved model was significantly enhanced, and the model accuracy benefited from the introduction of the attention mechanism and BiFPN (Weighted Bi-directional Feature Pyramid Network) structure, leading to an increase rather than a decrease.

Table 3. Model training results.

Model	*P*/%	*R*/%	$AP_{0.5–0.95}$/%	Size/MB	Params/M	FLOPs/G
YOLOv8n	96.18	96.67	93.97	6.3	3.006	8.1
LBDC-YOLO	97.65	95.82	94.44	3.5	1.928	6.5

3.2.2. Performance Comparison after Introducing Slim-Neck

As shown in Table 4, after applying Slim-neck, the size of the model reduced by 6.3%, the parameter count decreased by 7.0%, and the computational load decreased by 9.9%. Although there was a slight $AP_{0.5–0.95}$ decrease of 0.28 percentage points due to information loss caused by the DSC operation in GSConv, the computational complexity of the model was significantly reduced.

Table 4. Performance comparison after introducing Slim-neck.

Model	$AP_{0.5–0.95}$/%	Size/MB	Params/M	FLOPs/G
YOLOv8n	93.97	6.3	3.006	8.1
YOLOv8n-slim-neck	93.69	5.0	2.796	7.3

3.2.3. Performance Comparison after Introducing BiFPN

In this paper, BiFPN is used to replace the PANet structure in YOLOv8. The network test results are shown in Table 5.

Table 5. Performance comparison after introducing BiFPN.

Model	*P*/%	*R*/%	$AP_{0.5\sim0.95}$/%	Size/MB	Params/M	FLOPs/G
YOLOv8n	96.18	96.67	93.97	6.3	3.006	8.1
YOLO-BiFPN	96.05	97.39	94.19	4.3	1.992	7.1

After using BiFPN, the model's average precision increased from 93.97% to 94.19%. The model size reduced from 6.3 MB to 4.3 MB, the parameter count decreased from approximately 3 million to approximately 2 million, and the computational load decreased from 8.1 billion FLOPs to 7.1 billion FLOPs. The BiFPN structure removes nodes that contribute little, resulting in significant progress in lightweighting the model. The refined architecture, complemented by cross-scale connections and weighted feature operations introduced by BiFPN, augments the model's ability to fuse features and significantly enhances its accuracy.

3.2.4. Ablation Experiments

To evaluate the effectiveness of each step of the proposed improvement strategies in broccoli detection, improvement modules were gradually added in the ablation experiments, and the results are shown in Table 6.

Table 6. Results of the ablation experiment.

Number	Basic	Slim-Neck	Triplet	BiFPN	$AP_{0.5–0.95}$	Params/M	FLOPs/G	FPS
0	√				93.97%	3.006	8.1	21.6
1	√	√			93.69%	2.796	7.3	25.1
2	√	√	√		94.37%	2.798	7.3	22.8
3	√	√	√	√	94.44%	1.928	6.5	23.0

Compared to the baseline model, the proposed model in this paper improved $AP_{0.5–0.95}$ by 0.47 percentage points, while reducing the parameter count and computational load by 47.8% and 32.1%, respectively.

After applying the Slim-neck paradigm to the neck of YOLOv8, the model's computational complexity significantly decreased with a reduction of 7.0% in parameter count and 9.9% in computational load. However, this led to a decrease in model accuracy, with $AP_{0.5–0.95}$ decreasing by 0.28 percentage points.

By incorporating the lightweight Triplet Attention mechanism, the model's feature extraction capability was enhanced without significantly increasing the parameter count. This resulted in a 0.68 percentage point increase in $AP_{0.5–0.95}$. Lastly, the introduction of the BiFPN network structure in the neck improved the model's multi-scale feature fusion capability while reducing the complexity of the neck's feature fusion network. This led to a 0.07 percentage point increase in $AP_{0.5–0.95}$, a 31.1% reduction in parameter count, and a 11.0% reduction in computational load.

Each step of the improvement strategy has been effective, significantly reducing the computational complexity of the model while maintaining its accuracy.

3.3. Comparisons with Other Lightweight Models

The improved model proposed in this paper is compared with other commonly used lightweight detection models to validate its superiority. The results of the comparative experiments are shown in Table 7.

Table 7. Performance comparison of different lightweight models.

Model	$AP_{0.5–0.95}$/%	Params/M	FLOPs/G	FPS
LBDC-YOLO	94.44	1.928	6.5	23.0
YOLOv8n [27]	93.97	3.006	8.1	21.6
YOLOv8s [27]	94.08	9.799	24.2	18.6
YOLOv5n [29]	93.90	2.503	7.1	29.2
YOLOv5s [29]	94.39	9.142	24.0	23.2
YOLOv7-tiny [30]	90.21	2.796	7.3	31.5

The results indicate that the LBDC-YOLO model performs better than other models overall. Compared to YOLOv8n, YOLOv8s, YOLOv5n, YOLOv5s, and YOLOv7-tiny, LBDC-YOLO achieved $AP_{0.5–0.95}$ increases of 0.47 percentage points, 0.36 percentage points, 0.54 percentage points, 0.05 percentage points, and 4.23 percentage points, respectively. The improved model has a significantly lower parameter count and computational load compared to other models. Compared to YOLOv8n, YOLOv8s, YOLOv5n, YOLOv5s, and YOLOv7-tiny, LBDC-YOLO achieved a reduction in parameter counts by 35.86%, 80.32%, 22.98%, 78.91%, and 31.04%, respectively, and computational load by 19.75%, 73.14%, 8.45%, 72.91%, and 10.96%, respectively. LBDC-YOLO establishes itself as a frontrunner in lightweight design among other models. Taking into account both accuracy and lightweight design, the enhanced model stands out as the preferred choice for broccoli detection tasks.

3.4. Model Visualization Analysis

The crux of object detection resides in feature extraction. Given the constrained interpretability of neural network computations, to intuitively scrutinize the alterations in the model's feature extraction capabilities resulting from the improvement strategies outlined in this paper, we employ Grad-CAM for visual analysis. This is manifested through the generation of class activation maps for the detection model.

For each model, the hierarchy of shallow, intermediate, and deep feature maps is visualized, generating three heatmaps. The shallow feature maps have high spatial resolution, which benefits the extraction of features from smaller-scale objects and provides more accurate position information. The deep feature maps have larger receptive fields, which benefits the extraction of features from larger-scale objects and provides richer semantic information. The intermediate feature maps are between the shallow and deep layers. In the heatmap, the redder a certain region is, the greater its contribution to the detection.

For the original base model YOLOv8n, and the YOLOv8n model with Slim-neck introduced, in Figure 7c–h, it can be observed that there are many hotspot regions outside the target areas. This indicates that the network focuses on many irrelevant features, and these excessively attended irrelevant feature regions ultimately negatively impact the model's detection capability. After introducing the Triplet Attention mechanism, Figure 7i–k clearly show that the heatmaps are more concentrated on the target areas, and the noise outside the target areas is significantly reduced. This indicates that the model's "attention" becomes more focused, with less attention dispersed on irrelevant feature regions, resulting in improved feature extraction capability. With the introduction of BiFPN, the hotspot regions comprehensively cover the broccoli targets. In the heatmaps of larger and medium scales, shown in Figure 7m,n, the noise almost disappears, further enhancing the model's feature extraction capability.

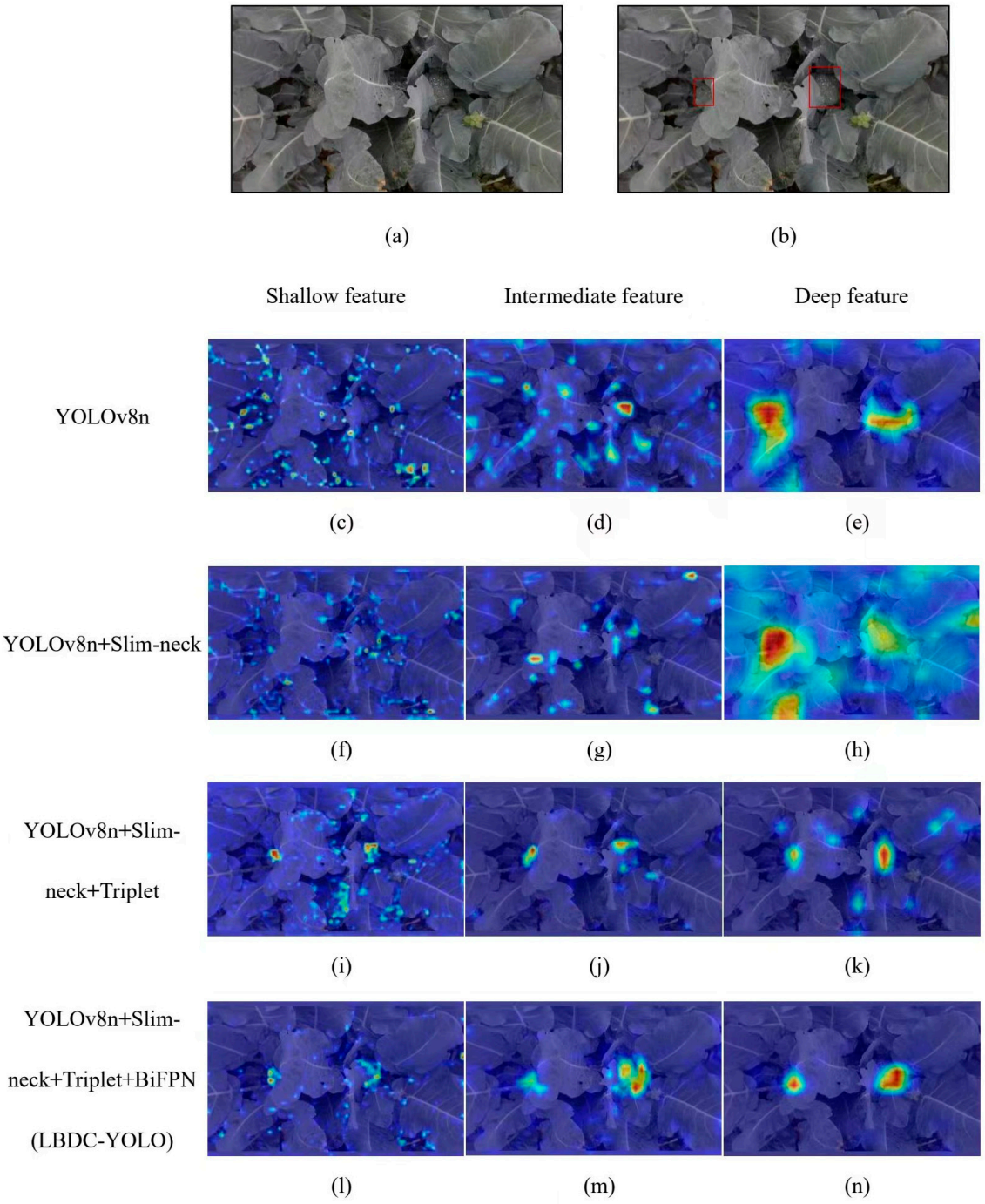

Figure 7. Visualization results of the model. (**a**) Original image; (**b**) original image with annotations; (**c**–**e**) are visualization heatmaps of shallow, intermediate, and deep feature maps of the YOLOv8n model, respectively; (**f**–**h**) are visualization heatmaps of shallow, intermediate, and deep feature maps of the YOLOv8n model with Slim-neck, respectively; (**i**–**k**) are visualization heatmaps of shallow, intermediate, and deep feature maps of the YOLOv8n model with Slim-neck and Triplet, respectively; (**l**–**n**) are visualization heatmaps of shallow, intermediate, and deep feature maps of the LBDC-YOLO model, respectively.

3.5. Model Detection Effect Analysis

The improved model exhibits good robustness in complex field environments affected by multiple factors, and its detection and localization capabilities have been enhanced based on YOLOv8n. The improved model demonstrates superior broccoli object detection ability

under the influence of shadows, occlusions, and dim lighting conditions. The detection results of the two models in typical scenarios are shown in Figure 8.

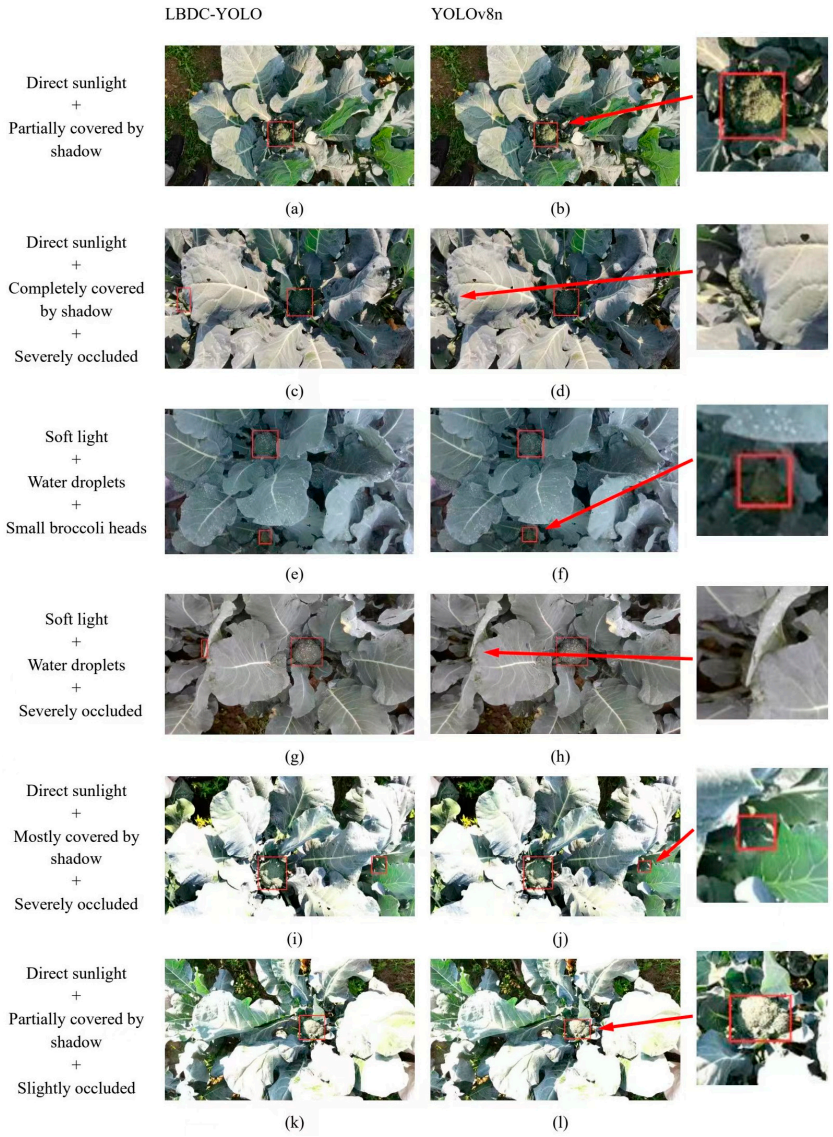

Figure 8. LBDC-YOLO and YOLOv8n model detection results. (**a**,**c**,**e**,**g**,**i**,**k**) show the detection results using the LBDC-YOLO model; (**b**,**d**,**f**,**h**,**j**,**l**) show the detection results using the YOLOv8n model. The environmental effects on the broccoli head in each image are listed in the first column. The red squares in the figures are model detection results, and the red arrows are used to indicate the position of the local zoom in the original figure.

In Figure 8a,b,k,l, clear shadows appear on the broccoli heads under direct sunlight. The illuminated areas have distinct features, while the darker areas have relatively blurry features. In this case, the YOLOv8n model mistakenly considers the boundary between light and dark regions as the detection box boundary. It can be observed that the YOLOv8n model fails to recognize the features of the darker parts of broccoli, resulting in the exclusion of the images where the broccoli heads are in the shadow from the detection boxes. On the other hand, the improved model generates more accurate detection boxes. In Figure 8c,d,g,h–j, there are objects that are largely occluded. The visible parts of these objects are small in scale and have irregular shapes, posing challenges for feature extraction and detection of small objects. The results show that the improved model performs better in detecting and accurately localizing these objects. In Figure 8e,f, there are small targets under dim lighting conditions. Due to the limitations of lighting, water droplets, and image resolution, these

targets almost blend in with the surrounding leaves. Although both models detect the target, the YOLOv8n model mistakenly includes the surrounding leaves in the detection box, while the improved model is not affected.

In field environments, broccoli heads are susceptible to the influence of lighting, occlusions, shadows, and water droplets. These influencing factors have a negative impact on the detection performance of the model. The proposed model in this study exhibits robustness under the influence of various factors, and there are two main reasons for this. Firstly, the proposed improvement strategy enhances the model's attention to the target, improves the fusion capability of position and semantic information, and strengthens the model's resistance to environmental interference. Therefore, even when the input image contains broccoli heads under different lighting conditions, various occlusion levels, different degrees of shadow coverage, and water droplet effects, the improved model can still extract effective features such as surface textures of the broccoli heads. Secondly, the training dataset used for model training is rich and diverse.

3.6. Testing on Independent Datasets of Others

To evaluate the robustness of our model, we tested it on publicly available datasets constructed by Blok et al. [11] and Kusumam et al. [24], where the LBDC-YOLO model was trained only using our own dataset. The LBDC-YOLO (trained additionally) model was further trained with a small subset of images from the corresponding independent datasets, and the experimental results are shown in Table 8.

Table 8. Different dataset test results.

Model	Dataset	$AP_{0.5}$/%	$AP_{0.5–0.95}$/%
LBDC-YOLO	Ours	98.77	94.44
LBDC-YOLO	Blok's	98.39	85.74
LBDC-YOLO (trained additionally)	Blok's	99.02	89.42
LBDC-YOLO	Kusumam's	44.54	33.03
LBDC-YOLO (trained additionally)	Kusumam's	99.31	86.62

In our experiments, the LBDC-YOLO model demonstrated good generalization capabilities on Blok et al.'s dataset [11]. Although the model's $AP_{0.5–0.95}$ dropped from 94.44% to 85.74% when applied to a new independent dataset, the model maintained a high performance similar to the original dataset on the $AP_{0.5}$ metric, showing the model's adaptability and robustness to different data characteristics. This further validates the model's effectiveness for object detection tasks. Subsequently, we conducted additional training on the LBDC-YOLO model with a small number of images (400 training images, 100 validation images) from the dataset constructed by Blok et al. [11], and the results after testing showed a significant improvement in $AP_{0.5–0.95}$, indicating that the issue of decreased precision in detection boxes in the independent dataset was mitigated. Moreover, these subtle performance changes also highlighted the model's stability in handling different types of data, providing a reliable foundation for future optimization and applications.

In the dataset constructed by Kusumam et al. [24], the large background differences, motion blur, and image variations caused by different shooting methods resulted in unsatisfactory performance of the LBDC-YOLO model trained on our own dataset, with only 44.54% $AP_{0.5}$ and 33.03% $AP_{0.5–0.95}$. However, after additional training of the LBDC-YOLO model with a small number of images (352 training images, 88 validation images) from the dataset, satisfactory results were achieved on images not used for training. This shows that although our collected dataset considered different environmental factors, such as weather, lighting, and shadows, the single shooting method used during collection and the similar background colors of the collection sites led to a lack of diversity in these aspects. This directly resulted in poor performance of the LBDC-YOLO model trained on our dataset when facing images with large background differences, motion blur, and different shooting methods. The excellent performance of the LBDC-YOLO model after additional

training with a few images further proves the significant impact of a diverse dataset on model performance.

4. Discussions

4.1. Discussion on Real-Time Detection Capability

The broccoli harvesting robot needs to continuously detect broccoli while moving, and it can only carry out the harvesting operation after detecting and locating broccoli targets. Therefore, the detection speed is a crucial factor affecting the efficiency of broccoli harvesting. In this study, LBDC-YOLO not only reduces the computational complexity of the model and improves detection accuracy, but also achieves a detection rate of 23.0 FPS on our device, meeting the real-time detection requirement. From the experimental results, the addition of the Triplet Attention mechanism in the improvement strategy has reduced the detection rate by 2.3 FPS. In future research, if further improvement of the detection speed is needed, optimization can be started from the Triplet Attention mechanism, which has greatly impacted the detection speed.

4.2. Limitations and Potential Applications

The object detection model proposed in this paper is not limited to the detection of broccoli. With its strong feature extraction capability, good multi-scale feature fusion ability, low computation, and low parameter requirements, a well-trained model supported by a sufficient and diverse dataset can be applied to detect various fruit and vegetable targets in mobile platform tasks. When detecting small targets, the model can also incorporate the features from the p2 layer of the backbone network into the neck network to enhance the detection performance of small targets.

To further achieve the all-weather operation of the broccoli selective harvesting robot, the vision system must be able to handle detection tasks in all lighting conditions, including nighttime artificial lighting environments. Several factors that may significantly affect the detection results need to be considered in the research: (1) Different types of artificial lighting sources, such as point sources, ring lights, and flat panel lights, can affect the occurrence of shadows and uniformity of illumination. (2) The brightness of the light source affects target visibility and the intensity of light reflected by water droplets. (3) The intensity of different wavelengths of light generated by the light source may have an impact.

During the real field operation of the broccoli harvesting robot, the camera frequently experiences shaking due to the uneven terrain and vibration from the power source. The high-frequency shaking seriously affects the image quality captured by the camera and subsequently impacts the detection performance. Future research will focus on algorithmic and mechanical stabilization to address this issue.

The broccoli images used in this study only include two varieties. Considering the diverse varieties of broccoli and the morphological differences and cultivation practices among different varieties, the detection model proposed in this paper may not necessarily achieve optimal performance in all broccoli detection tasks. It is important to gather representative images of other broccoli varieties for future research and further optimize the detection model based on that. Agricultural robots are transforming agricultural practices; agricultural robotics technology will revolutionize agriculture, reduce operational costs, and increase productivity in the face of labor shortages [35]. Given the good performance of the model proposed in this study, if future research can address its limitations, it may contribute even more significantly to the development of agricultural robots.

5. Conclusions

In this paper, we propose a target detection model named LBDC-YOLO, derived from the enhanced YOLOv8, designed to address multiple interfering factors in complex field environments and reduce the computational burden on computing devices. The model embraces the Slim-neck design paradigm to mitigate computational complexity. The incor-

poration of the Triplet Attention mechanism enhances the model's detection capabilities across various interfering factors. Additionally, the BiFPN structure is introduced to further optimize the model's lightweight nature and facilitate the fusion of multi-scale features.

Experimental results on a field dataset reveal that the improved model surpasses other commonly employed target detection algorithms, achieving an impressive Average Precision ($AP_{0.5–0.95}$) of 94.44%. Notably, the model boasts a parameter size and computational cost of only 1.9 M and 6.5 GFLOPs, respectively, while maintaining a high detection efficiency of 23.0 FPS on our device. These findings underscore this lightweight model's efficacy for real-time broccoli detection, particularly in complex environments.

At the same time, the limitations of this study should not be overlooked. Despite the good performance of the model proposed in this paper, there are areas for improvement, such as the decline in detection efficiency, the need for enhanced robustness against camera shake, and the limited dataset of broccoli varieties. Exploring different improvement strategies and adding a richer variety of broccoli images to the dataset may help enhance the overall performance of the model.

In future research, we aim to explore detection capabilities in nighttime artificial light conditions to achieve all-weather detection. Furthermore, the research plan encompasses efforts towards model acceleration and on-site testing during practical deployment scenarios.

Author Contributions: Conceptualization, S.G.; Data curation, Z.Z.; Formal analysis, S.G., H.P. and H.M.; Funding acquisition, Z.Z.; Investigation, S.G., H.P., Y.X. and H.M.; Methodology, Z.Z. and S.G.; Project administration, S.G.; Resources, Z.Z., S.G., H.P. and H.M.; Software, S.G., L.H. and G.M.; Supervision, Z.Z.; Validation, Z.Z., L.H. and G.M.; Visualization, S.G.; Writing—original draft, S.G.; Writing—review and editing, Z.Z. and S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the earmarked fund for China Agriculture Research System (CARS-23-D03), the General Program of Basic Science (Natural Science) Research in Higher Education Institutions of Jiangsu Province (23KJB210004) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD-2023-87).

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to zuozy@ujs.edu.cn.

Acknowledgments: The authors would like to acknowledge the support of the School of Agricultural Engineering at Jiangsu University, the reception of the Xiangshui County Agricultural Machinery Extension Station during the data collection period, and the assistance of Defang Lin, the contractor of the broccoli fields.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Treccarichi, S.; Ben Ammar, H.; Amari, M.; Cali, R.; Tribulato, A.; Branca, F. Molecular Markers for Detecting Inflorescence Size of *Brassica oleracea* L. Crops and *B. oleracea* Complex Species (n = 9) Useful for Breeding of Broccoli (*B. oleracea* var. *italica*) and Cauliflower (*B. oleracea* var. *botrytis*). *Plants* **2023**, *12*, 407. [CrossRef]
2. Li, H.; Xia, Y.; Liu, H.-Y.; Guo, H.; He, X.-Q.; Liu, Y.; Wu, D.-T.; Mai, Y.-H.; Li, H.-B.; Zou, L.; et al. Nutritional values, beneficial effects, and food applications of broccoli (*Brassica oleracea* var. *italica* Plenck). *Trends Food Sci. Technol.* **2022**, *119*, 288–308. [CrossRef]
3. López-Berenguer, C.; Martínez-Ballesta, M.; Moreno, D.A.; Carvajal, M.; Cristina, G.-V. Growing Hardier Crops for Better Health: Salinity Tolerance and the Nutritional Value of Broccoli. *J. Agric. Food Chem.* **2009**, *57*, 572–578. [CrossRef] [PubMed]
4. Zhou, C.; Hu, J.; Xu, Z.; Yue, J.; Ye, H.; Yang, G. A monitoring system for the segmentation and grading of broccoli head based on deep learning and neural networks. *Front. Plant Sci.* **2020**, *11*, 402. [CrossRef] [PubMed]
5. García-Manso, A.; Gallardo-Caballero, R.; García-Orellana, C.J.; González-Velasco, H.M.; Macías-Macías, M. Towards selective and automatic harvesting of broccoli for agri-food industry. *Comput. Electron. Agric.* **2021**, *188*, 106263. [CrossRef]
6. Zhu, Q.; Yang, H. Who are Engaging in Agriculture? -Investigations and Recognition to the Agricultural Labor Force. *J. China Agric. Univ. (Soc. Sci. Ed.)* **2011**, *28*, 162–169. [CrossRef]

7. Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik* **2014**, *125*, 5684–5689. [CrossRef]
8. Kelman, E.; Linker, R. Vision-based localisation of mature apples in tree images using convexity. *Biosyst. Eng.* **2014**, *118*, 174–185. [CrossRef]
9. Fu, L.; Duan, J.; Zou, X.; Lin, G.; Song, S.; Ji, B.; Yang, Z. Banana detection based on color and texture features in the natural environment. *Comput. Electron. Agric.* **2019**, *167*, 105057. [CrossRef]
10. He, L.; Wu, F.; Du, X.; Zhang, G. Cascade-SORT: A robust fruit counting approach using multiple features cascade matching. *Comput. Electron. Agric.* **2022**, *200*, 107223. [CrossRef]
11. Blok, P.M.; Barth, R.; van den Berg, W. Machine vision for a selective broccoli harvesting robot. *IFAC-Pap.* **2016**, *49*, 66–71. [CrossRef]
12. Dhaka, V.S.; Kundu, N.; Rani, G.; Zumpano, E.; Vocaturo, E. Role of Internet of Things and Deep Learning Techniques in Plant Disease Detection and Classification: A Focused Review. *Sensors* **2023**, *23*, 7877. [CrossRef] [PubMed]
13. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayaka, S.C.; Vocaturo, E.; Zumpano, E. Disease detection, severity prediction, and crop loss estimation in MaizeCrop using deep learning. *Artif. Intell. Agric.* **2022**, *6*, 276–291. [CrossRef]
14. Gangwar, A.; Dhaka, V.S.; Rani, G.; Khandelwal, S.; Zumpano, E.; Vocaturo, E. Time and Space Efficient Multi-Model Convolution Vision Transformer for Tomato Disease Detection from Leaf Images with Varied Backgrounds. *Comput. Mater. Contin.* **2024**, *79*, 117–142. [CrossRef]
15. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
16. Zeng, T.; Li, S.; Song, Q.; Zhong, F.; Wei, X. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. *Comput. Electron. Agric.* **2023**, *205*, 107625. [CrossRef]
17. Qiang, J.; Liu, W.; Li, X.; Guan, P.; Du, Y.; Liu, B.; Xiao, G. Detection of citrus pests in double backbone network based on single shot multibox detector. *Comput. Electron. Agric.* **2023**, *212*, 108158. [CrossRef]
18. Yan, J.; Zhao, Y.; Zhang, L.; Su, X.; Liu, H.; Zhang, F.; Fan, W.; He, L. Recognition of Rosa roxbunghii in natural environment based on improved Faster RCNN. *Trans. Chin. Soc. Agric. Eng.* **2019**, *35*, 143–150. [CrossRef]
19. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
20. Wang, Q.; Cheng, M.; Huang, S.; Cai, Z.; Zhang, J.; Yuan, H. A deep learning approach incorporating YOLO v5 and attention mechanisms for field real-time detection of the invasive weed Solanum rostratum Dunal seedlings. *Comput. Electron. Agric.* **2022**, *199*, 107194. [CrossRef]
21. Cui, M.; Lou, Y.; Ge, Y.; Wang, K. LES-YOLO: A lightweight pinecone detection algorithm based on improved YOLOv4-Tiny network. *Comput. Electron. Agric.* **2023**, *205*, 107613. [CrossRef]
22. Li, H.; Li, C.; Li, G.; Chen, L. A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. *Biosyst. Eng.* **2021**, *212*, 347–359. [CrossRef]
23. Blok, P.M.; van Evert, F.K.; Tielen, A.P.M.; van Henten, E.J.; Kootstra, G. The effect of data augmentation and network simplification on the image-based detection of broccoli heads with Mask R-CNN. *J. Field Robot.* **2021**, *38*, 85–104. [CrossRef]
24. Kusumam, K.; Krajník, T.; Pearson, S.; Duckett, T.; Cielniak, G. 3D-vision based detection, localization, and sizing of broccoli heads in the field. *J. Field Robot.* **2017**, *34*, 1505–1518. [CrossRef]
25. Kang, S.; Li, D.; Li, B.; Zhu, J.; Long, S.; Wang, J. Maturity identification and category determination method of broccoli based on semantic segmentation models. *Comput. Electron. Agric.* **2024**, *217*, 108633. [CrossRef]
26. Wada, K. *Labelme: Image Polygonal Annotation with Python*, v4.6.0; Python Software Foundation: Wilmington, DE, USA, 2021. Available online: https://github.com/wkentaro/labelme/releases/tag/v4.6.0 (accessed on 6 November 2022).
27. Jocher, G.; Chaurasia, A.; Qiu, J. *YOLO by Ultralytics*, v. 8.0.0; Ultralytics: Los Angeles, CA, USA, 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 2 March 2023).
28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
29. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J. *YOLOv5 by Ultralytics*, (Version 7.0); Ultralytics: Los Angeles, CA, USA, 2020. Available online: https://doi.org/10.5281/zenodo.3908559 (accessed on 2 July 2023).
30. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
31. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424. [CrossRef]
32. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to attend: Convolutional triplet attention module. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 3139–3148. [CrossRef]
33. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 3–19. [CrossRef]

34. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790. [CrossRef]
35. Vocaturo, E.; Rani, G.; Dhaka, V.; Zumpano, E. AI-Driven Agriculture: Opportunities and Challenges. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023; pp. 3530–3537. [CrossRef]