agronomy

Article

Research and Explainable Analysis of a Real-Time Passion Fruit Detection Model Based on FSOne-YOLOv7

Juji Ou, Rihong Zhang *, Xiaomin Li and Guichao Lin

College of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China; oujjcm@163.com (J.O.); lixiaomin@zhku.edu.cn (X.L.); guichaolin@126.com (G.L.)

* Correspondence: zhangrihong@zhku.edu.cn

Abstract: Real-time object detection plays an indispensable role in facilitating the intelligent harvesting process of passion fruit. Accordingly, this paper proposes an FSOne-YOLOv7 model designed to facilitate the real-time detection of passion fruit. The model addresses the challenges arising from the diverse appearance characteristics of passion fruit in complex growth environments. An enhanced version of the YOLOv7 architecture serves as the foundation for the FSOne-YOLOv7 model, with ShuffleOne serving as the novel backbone network and slim-neck operating as the neck network. These architectural modifications significantly enhance the capabilities of feature extraction and fusion, thus leading to improved detection speed. By utilizing the explainable gradient-weighted class activation mapping technique, the output features of FSOne-YOLOv7 exhibit a higher level of concentration and precision in the detection of passion fruit compared to YOLOv7. As a result, the proposed model achieves more accurate, fast, and computationally efficient passion fruit detection. The experimental results demonstrate that FSOne-YOLOv7 outperforms the original YOLOv7, exhibiting a 4.6% increase in precision (P) and a 4.85% increase in mean average precision (mAP). Additionally, it reduces the parameter count by approximately 62.7% and enhances real-time detection speed by 35.7%. When compared to Faster-RCNN and SSD, the proposed model exhibits a 10% and 4.4% increase in mAP, respectively, while achieving approximately 2.6 times and 1.5 times faster real-time detection speeds, respectively. This model proves to be particularly suitable for scenarios characterized by limited memory and computing capabilities where high accuracy is crucial. Moreover, it serves as a valuable technical reference for passion fruit detection applications on mobile or embedded devices and offers insightful guidance for real-time detection research involving similar fruits.

Keywords: passion fruit; YOLOv7; lightweight; reparameterization; explainable

Citation: Ou, J.; Zhang, R.; Li, X.; Lin, G. Research and Explainable Analysis of a Real-Time Passion Fruit Detection Model Based on FSOne-YOLOv7. *Agronomy* **2023**, *13*, 1993. https://doi.org/10.3390/agronomy13081993

Academic Editor: Roberto Marani

Received: 15 June 2023
Revised: 19 July 2023
Accepted: 25 July 2023
Published: 27 July 2023

1. Introduction

Passion fruit, a fruit widely cultivated in tropical and subtropical regions, primarily thrives in the southern provinces of China. It is recognized for its potential health benefits [1] derived from the pulp, peel, and seeds, which have garnered attention for both natural consumption and industrial processing [2]. In recent years, passion fruit has experienced substantial economic growth and an increased market demand [3,4]. However, conventional manual harvesting methods cannot satisfy the demands of modern agriculture. Therefore, the utilization of intelligent robots for passion fruit harvesting has emerged as a promising solution [5]. To ensure the efficiency and quality of robotic harvesting, the real-time detection of passion fruit plays a critical role.

With the advent of deep learning methods, remarkable progress has been made in object detection algorithms. These algorithms can be broadly classified into one-stage and two-stage detection methods [6]. One-stage detection algorithms, such as YOLO [7–9], CenterNet [10], and SSD [11], are renowned for their fast detection speeds and efficient computational performance, making them well-suited for the real-time detection of agricultural targets in complex environments [12]. Researchers have successfully improved these algorithms to achieve high-precision fruit detection in challenging agricultural settings. For instance, Sekharamacntry et al. [13] proposed an enhanced YOLOv5 model for real-time apple detection, achieving an impressive detection accuracy of 97%. Quan et al. [14] employed depth information in a dual-stream, dense, feature fusion network based on YOLO-V4 to predict the fresh weight of weeds, yielding a detection error of approximately 4%. In a similar vein, Lu et al. [15] presented the Swin-transformer-YOLOv5 model for the real-time detection of clustered wine grape bunches, achieving a maximum mAP of 97%. Additionally, Ridho and Irwan [16] developed a real-time strawberry quality assessment model based on an improved SSD, attaining an accuracy of 90% when tested on a robot. On the other hand, two-stage detection methods, such as Faster R-CNN [17] and Mask R-CNN [18], find broader applications in specific scenarios. For instance, Pan et al. [19] achieved the automatic detection of sugarcane seedlings with an average accuracy of 93.67% using an enhanced Faster R-CNN model and a non-maximum suppression algorithm. Zhong et al. [20] combined an improved Faster R-CNN model with depth information to locate clustered chili peppers, achieving an average precision (AP) of 87.30%. Moreover, Kumar and Kukreja [21] proposed a wheat leaf virus detection algorithm based on Mask R-CNN, achieving a remarkable detection accuracy of 97.16%. In summary, one-stage object detection methods are well-suited for tasks requiring real-time performance, while two-stage object detection methods typically offer higher detection accuracy at the expense of increased computational resources and time.

In the domain of fruit object detection, researchers have focused on improving the performance and efficiency of detection models, particularly emphasizing lightweight design and model optimization. Zhang et al. [22] proposed a lightweight apple detection model based on YOLOv4. By incorporating networks such as GhostNet and depth-wise separable convolutions, they constructed a lightweight model that enabled a detection speed of 45.2 fps. Similarly, Shang et al. [23] introduced an improved lightweight detection model for apple blossoms using YOLOv5s. By employing techniques such as ShuffleNetv2 and Ghost modules, they achieved a detection speed of 86.21 fps. Zeng et al. [24] presented a tomato fruit detection algorithm based on an enhanced YOLOv5 model. They utilized MobileNetV3 as the backbone network and combined it with channel pruning methods, resulting in a lightweight model with an average detection speed of 26.5 fps on mobile devices. These works, by reducing model parameters and computational complexity, enhanced detection speeds and adaptability to resource-constrained devices, providing valuable insights and approaches to lightweight fruit object detection models.

The complexity and uncertainty associated with agricultural robot operations in intricate environments necessitate further research into deep learning-based methods for image recognition in passion fruit identification and localization. Recently, numerous deep learning approaches have emerged in the field of passion fruit recognition. Luo et al. [25] employed the lightweight MobileNetV3 network within YOLOv5 in order to enhance the speed of passion fruit detection; yet, optimal accuracy was not achieved. Additionally, Wu et al. [26] introduced DenseNet into YOLOv3 to enhance the detection accuracy of passion fruit in natural environments, albeit at the expense of increased computational demands and memory consumption, thereby impacting inference speeds. Two-stage object detection algorithms, such as those [27,28] based on an improved Faster R-CNN, often exhibit excellent detection performance in complex environments; yet, their complex model structures hinder detection speeds. While one-stage object detection algorithms can improve detection speeds, they may sacrifice a certain degree of accuracy. Conversely, two-stage object detection algorithms excel in complex environments but are restricted by the complexity of their model structures. However, given the intricacy of agricultural robot operations and the uncertainty of the environment, the fast detection characteristics of one-stage algorithms make them more applicable. Premised on the aforementioned issues and background, this study aimed to design a real-time passion fruit detection model for intelligent harvesting robots. A deep learning model termed FSOne-YOLOv7 is proposed to strike a balance between detection accuracy and inference speeds as much as possible. In Figure 1, we illustrate the overall structure and workflow of the FSOne-YOLOv7 model, which includes the enhanced version based on the YOLOv7 architecture, the novel backbone network ShuffleOne (blue region), and the neck network slim-neck (green region). These modified architectures significantly enhance the capabilities of feature extraction and fusion, resulting in improved detection accuracy and speed.

The subsequent sections of this paper are organized as follows: Section 2 presents the introduction and analysis of the white passion fruit dataset, along with the evaluation parameters. In Section 3, the fundamental theory and novel methodologies employed in this research are elaborated upon. Section 4 provides a comprehensive discussion and analysis of the experimental test results obtained in this study. Section 5 engages in a discourse regarding the research findings, while also presenting ideas for further enhancement and improvement. Finally, Section 6 concludes the paper.

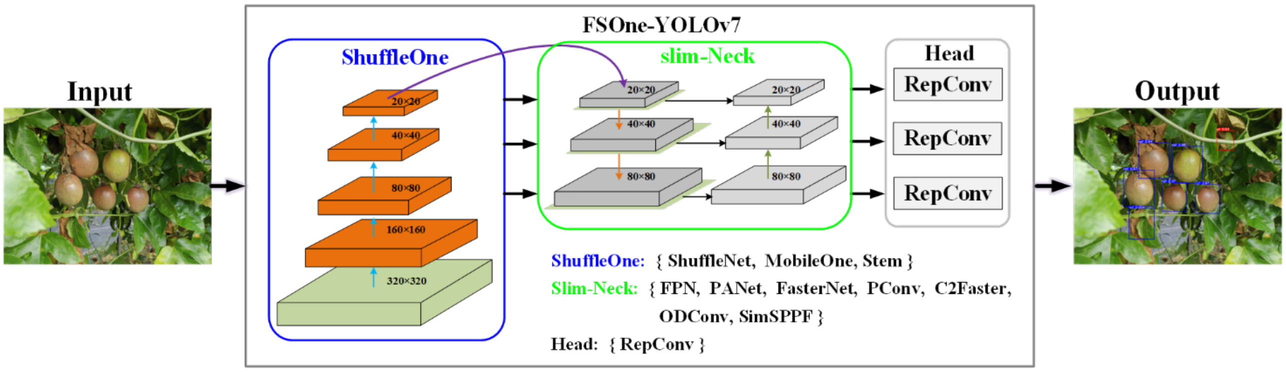

Figure 1. The overall structure and workflow of the FSOne-YOLOv7 model.

2. Materials

2.1. Image Acquisition and Presentation

The passion fruit image dataset utilized in this study was obtained from a purple passion fruit plantation located at the Zhanjiang Institute of Science in Guangdong Province, China. The specific variety employed in the dataset was Purple Fragrance No. 1, and data collection occurred in June 2022. In order to optimize growth conditions and enhance yields by mitigating excessive foliage that could hinder light exposure, a vertical trellis cultivation technique was implemented at the plantation. Pruning activities were carried out during both the growth and fruiting stages of the passion fruit plants. To ensure dataset diversity, a collection of passion fruit growth scenes was obtained within the height range of 0.5 to 1.3 m and depth range of 0.3 to 0.8 m. A total of 2560 images were captured for the dataset. Figure 2 showcases close-up perspectives of purple passion fruit at various stages of ripeness. In Figure 2a, the passion fruit is depicted in its unripe stage, distinguished by a greenish tint on the fruit's skin. In Figure 2b, the fruit is showcased during the ripening stage, undergoing a transition from green to purple. It is generally considered suitable for harvesting when the fruit initiates a color change, taking into account the necessary time for transportation and sales. Figure 2c portrays the fruit in its mature stage, where the majority of the purple passion fruit's skin exhibits a reddish-purple hue. Lastly, Figure 2d displays the fruit in its fully ripe stage, characterized by a deep purple coloration prevalent across the skin.

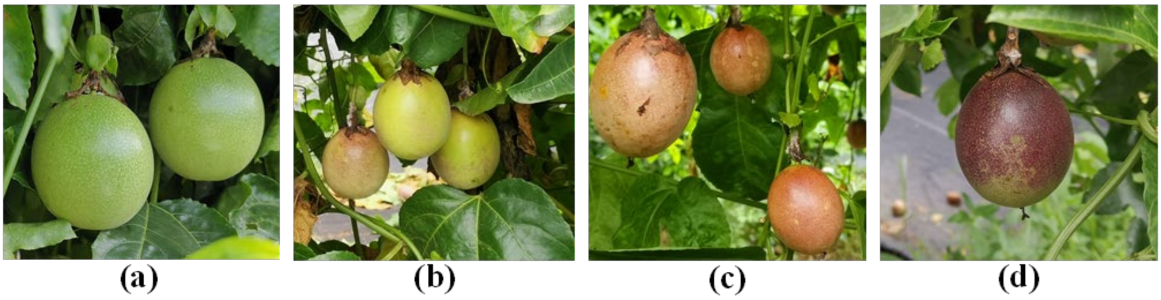

Figure 2. Close-up view of the passion fruit dataset: (**a**) unripe passion fruit, (**b**) unripe passion fruit, (**c**) ripe passion fruit, and (**d**) fully ripe passion fruit.

As demonstrated in Figure 2, the growth environment of passion fruit is characterized by its complexity, showcasing diverse visual attributes. The fruit exhibits distinct color features at different stages of ripeness. In Figure 2a, the fruit is situated within a growth environment where its color closely resembles that of the background. Figure 2b,c illustrate the visual characteristics of passion fruit, encompassing variations in fruit size. In addition, Figure 3 provides additional elucidation of the diverse growth environments for passion fruit. The close-up image on the right side of Figure 3a depicts passion fruit partially concealed by surrounding leaves. Similarly, the close-up image on the right side of Figure 3c reveals passion fruit heavily obstructed by branches and leaves. The close-up images on both sides of Figure 3b showcase growth environments where passion fruit is obstructed by branches and leaves, with dense or overlapping fruits present. The close-up image on the left side of Figure 3c portrays a growth environment where the passion fruit's skin color closely resembles the background color. Finally, the close-up image on the left side of Figure 3a highlights the visual attribute of varying fruit sizes within passion fruit.

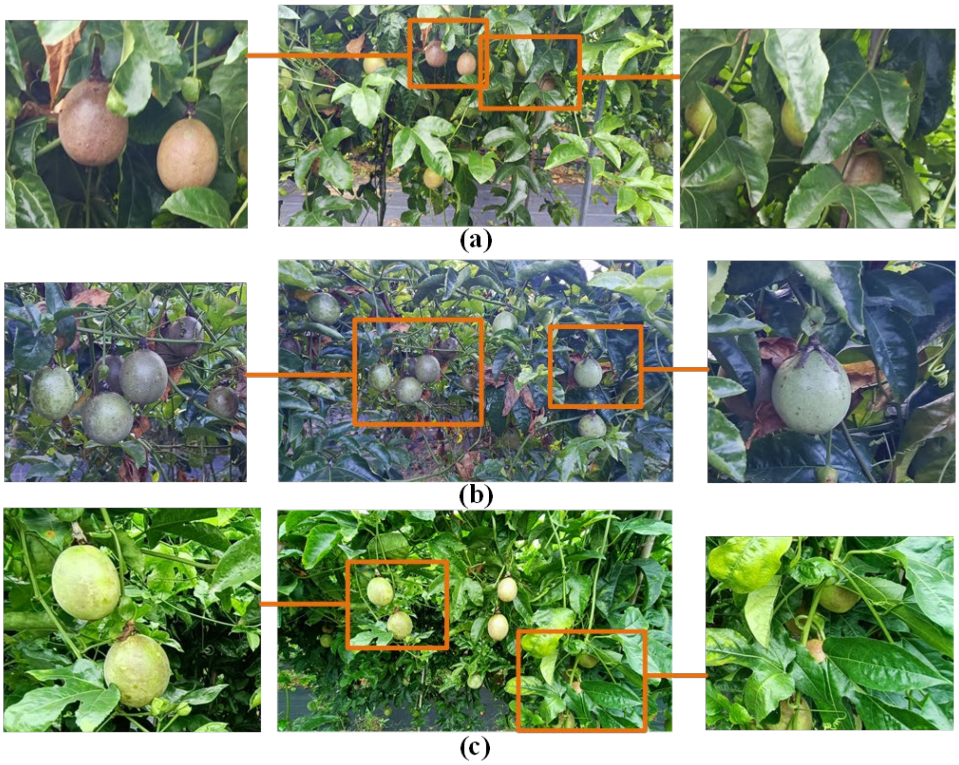

Figure 3. Passion fruit data examples and some close-up images. (**a**) Occluded and varying sizes, (**b**) Occluded and dense, (**c**) Occluded and similar background color.

2.2. Dataset Augmentation and Annotation

The base dataset comprised 2000 high-quality images that were carefully selected from the collected image data. Among these, 400 images were randomly allocated as the test set, while the remaining 1600 images constituted the base training set. To enhance the model's generalization ability, a variety of data augmentation techniques were applied to a subset of the base training set. These techniques encompassed brightness adjustment, mirror flipping, Gaussian blur, noise addition, and mix-up augmentation. Moreover, image cropping was employed to extract 560 valid samples from the original data with unstable quality, ensuring the preservation of the data's authenticity. Specific details are provided in Table 1. Following the augmentation process, the total number of images in the training set increased to 3280.

Table 1. Passion fruit data expansion parameter description.

Data Source	Number of Images	Image Size (Pixels)	Enhancement Method	Number of Images (after Enhancement)
Basic training set	1120	4608 × 2592	Brightness adjusting	222
			Mirror flipping	222
			Gaussian blur	148
			Adding noise	148
			Mix-up	742
Image data of unstable quality	560	1920 × 1080	Image cropping	480

The real-time detection and classification of passion fruit ripeness are of the utmost importance in achieving intelligent harvesting. Harvesting fully ripe fruits promptly is essential to prevent overripening and decay. Moreover, the ripeness of passion fruits has a significant impact on post-harvest storage. Fruits that have reached optimal ripeness are more suitable for storage and long-distance transportation, whereas unripe fruits possess inferior texture and flavor, reducing their market value. Therefore, it is advisable to refrain from harvesting unripe passion fruits, but those that exhibit a color change or that have turned purple can be harvested. To accomplish this objective, we utilized the labeling tool LabelImg for manual annotation, employing the minimum bounding rectangle method for each passion fruit instance. During the annotation process, we assigned the passion fruit labels into two categories: "pf" (representing unripe fruits) and "rpf" (representing ripe fruits). For a comprehensive breakdown of the specific categories and their corresponding quantities, please refer to Table 2.

Table 2. Passion fruit type and quantity description.

Total Number of Images	Data Type	Passion Fruit Category	Number of Passion Fruit Categories
3962	Training set (3562)	Images of immature passion fruits	5550
		Images of ripe passion fruits	9446
	Test set (400)	Images of immature passion fruits	804
		Images of ripe passion fruits	1293

3. Methods

3.1. FSOne-YOLOv7 Network Model

YOLOv7 [9] represents a single-stage model for object detection, surpassing the performance of other algorithms in the YOLO series in terms of both speed and accuracy. This accomplishment can be attributed to its advanced network architecture and the utilization of sophisticated training strategy techniques. Illustrated in Figure 4, the YOLOv7 model inherits the fundamental YOLO object detection network, which comprises three key components: the backbone, neck, and head. The backbone component of YOLOv7 plays a pivotal role in feature extraction. It incorporates a stem element and four sequentially connected blocks, employing the highly efficient ELAN network module [29]. For feature fusion, the neck component leverages the Path Aggregation Network (PANet) [30]. As for feature decoding, the head component employs a structural reparameterization approach based on gradient flow propagation path analysis. This includes the integration of a redesigned reparameterization convolution network known as RepConv. Collectively, these three components are synergistically combined in YOLOv7 to yield enhanced performance in various object detection tasks.

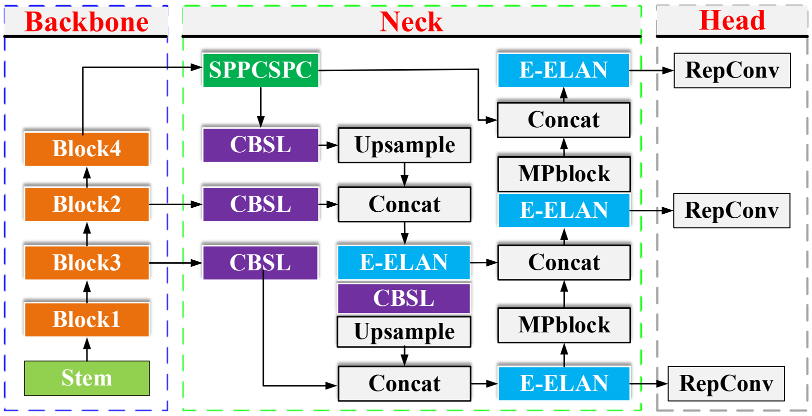

Figure 4. Network architecture diagram of YOLOv7.

The network model proposed in this study, FSOne-YOLOv7, showcases an architecture based on YOLOv7, as depicted in Figure 5. Notably, it introduces a redesigned backbone termed ShuffleOne, which encompasses a sequence of ShuffleOne blocks, effectively replacing the original backbone. Additionally, a streamlined and efficient slim-neck replaces the initial neck component. The ShuffleOne block, renowned for its lightweight nature and high efficiency, serves as the building block within ShuffleOne. Its primary objective is to swiftly capture the diverse appearance features of passion fruit, particularly in intricate environments. The ShuffleOne's stem comprises a mere three CBSL structures, while each block comprises a stride = 2 ShuffleOne block and a variable number of stride = 1 ShuffleOne blocks. A CBSL structure encompasses a convolution layer (Conv), batch normalization layer (BN), and Silu activation function layer (Silu). Moreover, the slim-neck is carefully crafted by considering various convolution methods, feature fusion structures, and spatial pyramid pooling structures (SPP) [31], effectively replacing the original neck. Within the slim-neck, the E-ELAN module is substituted with the C2Faster module to minimize redundant computations and memory access. The ODConv module is introduced to augment the acquisition of contextual and spatial information. Eventually, the original SPPCSPC module is replaced with SimSPPF to achieve accelerated inference speeds. In conclusion, this study presents the FSOne-YOLOv7 model, which offers real-time detection capabilities specifically tailored for passion fruit. This is achieved by substituting the original backbone and neck of YOLOv7 with ShuffleOne and a slim-neck, respectively.

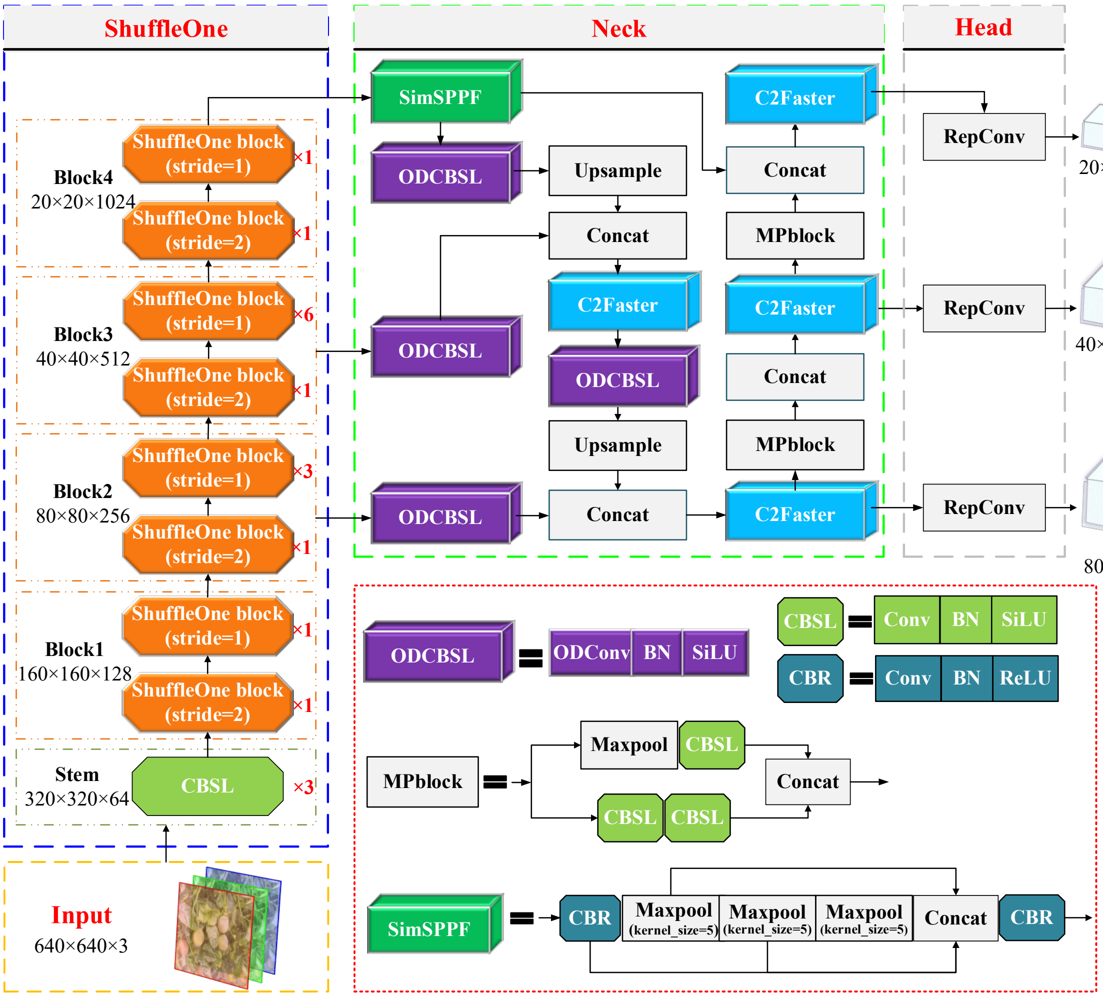

Figure 5. Network architecture diagram of FSOne-YOLOv7.

3.2. *ShuffleOne Backbone Network*

3.2.1. ShuffleNet

ShuffleNet [32] is an advanced network architecture that leverages group convolution and introduces the channel shuffle operation. These techniques facilitate the transfer of cross-channel information, enhancing the extraction of features while concurrently reducing computational and parameter costs. Building upon ShuffleNet, ShuffleNetV2 [33] further refines the network structure by considering factors such as memory access cost (MAC) and the level of network parallelization. To improve efficiency, ShuffleNetV2 introduces the “Channel Split” operation, which segregates the input feature map into multiple branches based on channel count. Each branch then undergoes distinct convolution operations. The results from each branch are subsequently merged and subject to channel shuffling, enhancing interactivity and complexity among the features and, in turn, improving overall model performance. By employing ShuffleNet or ShuffleNetV2, it becomes possible to achieve model compactness and acceleration in resource-constrained environments without compromising accuracy. These network architectures have showcased remarkable performance in image recognition and object detection tasks, making them highly valuable in scenarios where performance is restricted, such as mobile and embedded devices.

Figure 6a,b represent the cases of stride = 1 and stride = 2, respectively, within the ShuffleNetV2 block. When stride = 1, the input features undergo a “Channel Split” operation, dividing them into two branches. The left branch remains unchanged, while the right branch consists of three convolutional layers. To reduce MAC, the input and output channels of these layers are kept the same. Instead of using 1 × 1 grouped convolutions (1 × 1 GConv), 1 × 1 convolutions are employed to mitigate fragmentation and enhance network parallelism and MAC. Following the 1 × 1 convolution, there are 3 × 3 depthwise convolutions (3 × 3 dConv) and another 1 × 1 convolution. The features from the left and right branches are then concatenated using the Concat operation, maintaining the same number of output feature channels. Finally, the concatenated features undergo the "Channel Shuffle" operation to generate the final output. When stride = 2, there is no splitting of the input features. The left branch consists of a 3 × 3 depthwise convolution (3 × 3 dConv) and a 1 × 1 convolution. It is worth noting that "dConv" refers to a depthwise convolution, while "1 × 1 Conv" denotes a pointwise convolution. The combination of 3 × 3 dConv and 1 × 1 Conv forms a depthwise separable convolution module.

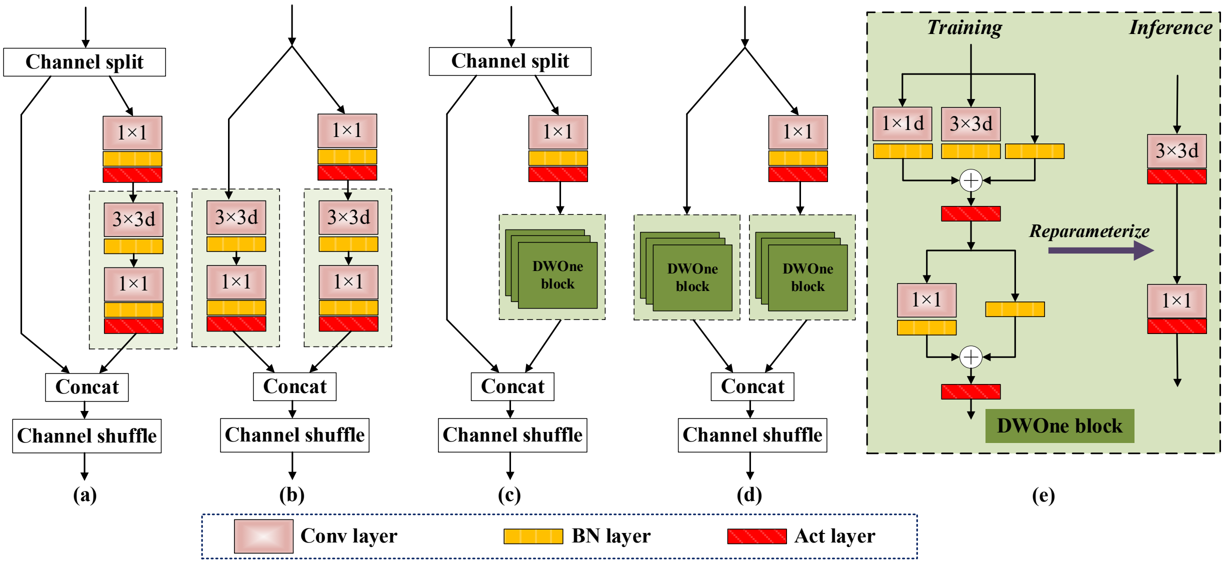

Figure 6. The structure diagrams of the ShuffleNetV2 block and ShuffleOne block. (**a**,**b**) are the basic unit structure diagrams of the ShuffleNetV2 block with stride = 1 and stride = 2, respectively. (**c**,**d**) are the basic unit structures of the ShuffleOne block with stride = 1 and stride = 2, respectively. (**e**) represents the different structures of the DWOne block during training and inference. This module serves as the basic unit for the structural reparameterization of the depth-wise separable convolution (DWConv). The sizes of the convolutional kernels are denoted as 1 × 1 and 3 × 3, and the "d" in (**e**) indicates the depth-wise convolution operation. The act layer applies the ReLU activation function.

3.2.2. DWOne Block

Network models characterized by the presence of multiple branches inherently possess the capability to augment model representation [34–37]. Motivated by prior studies on structural reparameterization [38–40], the introduction of linear convolutions during the training phase serves to compensate for the multi-branch structure, thereby leading to an improvement in model accuracy. During the inference stage, the multi-branch structure is reconfigured into a single-branch structure through reparameterization, resulting in a reduction in the number of convolution operations performed and a decrease in model memory usage. Therefore, the efficiency of the inference speed is enhanced. The DWOne blocks closely resemble the MobileOne blocks proposed in [41], with the fundamental block still adhering to the 3 × 3 depth convolution, followed by a 1 × 1 point convolution structure. The distinction lies in the elimination of cumbersome over-parameterized branches, instead incorporating batch normalization alongside a reparameterized skip connection that eschews the replication of structural branches, as illustrated in Figure 6e.

The convolution operation can be expressed for an input feature map $X_1 \in R^{C \times H \times W}$, output feature map $X_2 \in R^{C \times H \times W}$, and convolution kernel size of $K \times K$ using the weight matrix $W_{Conv} \in R^{C \times C \times k \times k}$ and an optional bias term $b_{Conv} \in R^{C}$. In the context of the convolution operator symbol defined as *, the convolution operation can be denoted as follows:

$$X_2 = W_{Conv} * X_1 + b_{Conv} \tag{1}$$

BN layer operation involves a sequence of steps applied to each individual feature in the feature map. This includes subtracting the mean and dividing it by the standard deviation, followed by adding the mean and multiplying it by the standard deviation. In this context, the accumulated mean, standard deviation, scaling factor, and bias are denoted as μ, σ, γ, and β, respectively. To prevent division by zero, a very small number ε is added to the denominator. The linear representation of the BN operation on X_2 is as follows:

$$BN(X_2) = \gamma \frac{X_2 - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{2}$$

The output feature map X_2 resulting from the convolution operation is incorporated into formula (1) and subsequently decomposed to derive the following expression:

$$BN(X_2) = \gamma \frac{W_{Conv}}{\sqrt{\sigma^2 + \varepsilon}} * X_1 + \beta + \gamma \frac{b_{Conv} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{3}$$

To simplify formula (3), we can make the following expression:

$$\hat{W} = \gamma \frac{W_{Conv}}{\sqrt{\sigma^2 + \varepsilon}}, \hat{b} = \beta + \gamma \frac{b_{Conv} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{4}$$

The convolutional layer can then be re-expressed as a convolution performed after the operation of the BN layer:

$$X_2 = \hat{W} * x + \hat{b} \tag{5}$$

$$W = \sum_i^N \hat{W}, b = \sum_i^N \hat{b} \tag{6}$$

The skip connection can be considered a 1 × 1 convolutional operation, where the identity matrix serves as the kernel. In the case of a skip connection combined with a BN layer, it can be simplified as a convolutional operation with a 1 × 1 kernel. Hence, during the inference phase, the BN layers associated with all branches within the DWOne block are merged with the preceding convolutional layers to form a new convolutional layer. Figure 6e illustrates the DWOne block, which consists of three branches in the upper part and two branches in the lower part. After fusing the convolutional and BN layers, the upper part yields a 3 × 3 kernel, two 1 × 1 kernels, and three bias vectors. The lower part obtains two 1 × 1 kernels and two bias vectors. Following a similar approach as described in [41], the upper part of the block generates a new 3 × 3 kernel by zero-padding the 1 × 1 kernel and adding it to the center of the 3 × 3 kernel. By applying Equation (6), the final kernel and bias vectors can be computed separately for the upper and lower parts of the DWOne block. Herein, N represents the number of branches. Through these computations, the multi-branch structure is transformed into a single-branch structure during the inference phase.

3.2.3. Integration of the ShuffleNetV2 Block and DWOne Block: ShuffleOne Block

The primary objective of ShuffleNetV2 is to minimize the number of floating-point operations (FLOPs). In order to achieve this, the output channels of the depthwise convolution in each branch are not expanded when utilizing depthwise and pointwise convolutions. During the training phase of the DWOne block, the multi-branch structure compensates for the model's accuracy. However, during the inference phase, a single-branch structure enhances the model's inference speed. In Figure 5, We have replaced the dashed box depicted in Figure 5a,b with the DWOne block, resulting in the structures illustrated in Figure 5c,d The ShuffleNetV2 block, integrated with the DWOne block, is now referred to as the ShuffleOne block. Similarly, Figure 5c,d in the figure correspond to the ShuffleOne block in cases where the stride is equal to 1 and 2, respectively.

3.3. Simple and Efficient Slim-Neck

3.3.1. C2Faster

To minimize computational redundancy and optimize MAC in the neck section, while taking into account convolutional techniques and feature fusion structures, we propose a C2Faster module as a replacement for the original E-ELAN module. This substitution is carried out without disturbing the original gradient pathways. By studying the approaches that enhance the learning capabilities of CNNs in ELAN [29] and CSPNet [42], we have devised the network structure depicted in Figure 6b. The C2Faster module is composed of 1 × 1 convolutional layers at the beginning and end serving for channel expansion and contraction, respectively. In the middle, the “Channel split” operation is initially employed to partition the channels. The Faster Block [43] is utilized as the primary branch for channel flow. The concept of gradient routing is employed to propagate the feature information of each Faster Block and concatenate it with the preceding split information. Finally, an additional 1 × 1 convolutional layer is employed to adjust the output channels.

The integration of the Faster Block brings forth enhanced feature extraction capabilities and improved latency performance. This block encompasses a partial convolution (PConv) layer along with two consecutive 1 × 1 convolutional layers, resulting in an inverted residual structure, as demonstrated in Figure 7a. The Conv 1 × 1 layer effectively leverages information from all channels, while the channel expansion in the middle layer necessitates the inclusion of a “Shortcut” to reuse input features and mitigate the issue of gradient divergence in deep networks. The PConv operation, depicted in Figure 7, follows a straightforward, swift, and efficient convolutional approach. It selectively applies the conventional convolution operation to a subset of input channels for spatial feature extraction while keeping the remaining input channels unaltered. This technique effectively reduces redundant computations and minimizes memory access requirements.

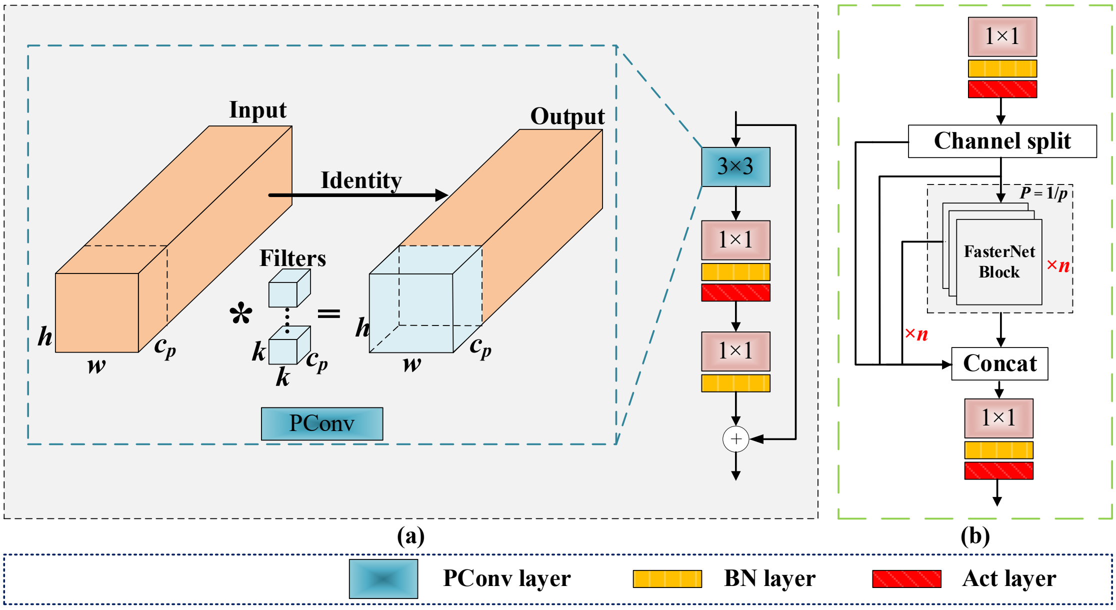

Figure 7. (**a**) FasterNet block structure diagram; (**b**) C2Faster structure diagram. * represents convolution operation.

Assuming that C_{in} and C_{out} are equal and denoted as c, the number of FLOPs for the convolution operation is $h \times w \times k^2 \times c^2$ and, for the DWConv, it is $h \times w \times k^2 \times c$. In the case of PConv with a partial ratio $P = 1/p$ where p represents the fraction of channels to be convolved, the number of channels involved in the convolution operation is c_P. The FLOPs for PConv can be calculated as $h \times w \times k^2 \times c_P{}^2$ and the FLOPs for PConv are *1/p* times the FLOPs for Conv. Comparatively, DWConv has a smaller number of FLOPs. Both DWConv and Conv have the same number of MAC:

$$h \times w \times 2c + k^2 \times c^2 \approx h \times w \times 2c \tag{7}$$

which is higher than that of a Pconv:

$$h \times w \times 2c_P + k^2 \times c_P{}^2 \approx h \times w \times 2c_P \tag{8}$$

When using DWConv (typically followed by PWConv) instead of Conv, it is often necessary to increase the output dimension of DWConv by a multiplier to compensate for the potential loss in accuracy. However, this increase in the output dimension also results in higher memory access, which inevitably leads to delays in the network's inference speed. On the other hand, selecting the FasterNet block helps achieve lower latency and higher throughput. In general, PConv has lower FLOPs compared to conventional convolutions but higher FLOPs than DWConv. However, PConv outperforms both convolutions and DWConv in terms of utilizing computational capabilities on devices. In the C2Faster module illustrated in Figure 7, the partial ratio $P = 1/p$ represents the proportion of PConv within the FasterNet block, where n denotes the number of FasterNet blocks or parallel gradient flow branches. In this particular study, in order to reduce the computational complexity and inference time of the detector, the values $p = 8$ and $n = 1$ were set in the ablation experiments.

3.3.2. Omni-Dimensional Dynamic Convolution: ODConv

In lightweight backbone networks, the pervasive utilization of lightweight convolution kernels can impose constraints on the network's apprehension of global contextual features. Despite the implementation of depthwise separable convolutions for computational reduction, spatial convolution operations in the spatial dimension may engender the dissipation of high-frequency information, thereby impinging on the model's proficiency in capturing high-frequency intricacies and textures in images. In order to augment the model's capacity for feature fusion, this study introduces dynamic convolution to establish interconnections between the output features of varying scales within the Shuffle-One module residing in the slim-neck.

Conventional methods of dynamic convolution [44,45] typically employ attention mechanisms in a singular dimension of the kernel space to dynamically modulate the convolution kernels' weights. In contrast, ODConv [46] employs complementary attention mechanisms across all four dimensions of the kernel space to dynamically regulate the convolution kernels' weights. As depicted in Figure 8, the input x undergoes an initial global average pooling (GAP) operation, followed by processing through fully connected (FC) layers and activation functions (ReLU). Diverging from conventional dynamic convolutions, ODConv features four parallel header branches, each outfitted with an FC layer and either a Softmax or Sigmoid function. This process can be expressed mathematically using Equation (9). By incorporating dynamic convolutions, the model gains the ability to adaptively adjust the weights of the convolution kernels based on the input data's features, thereby enhancing the fusion capability of the features. Therefore, the model's perceptual and capturing aptitude of high-frequency information, such as image details and textures, is improved, leading to enhanced performance and accuracy.

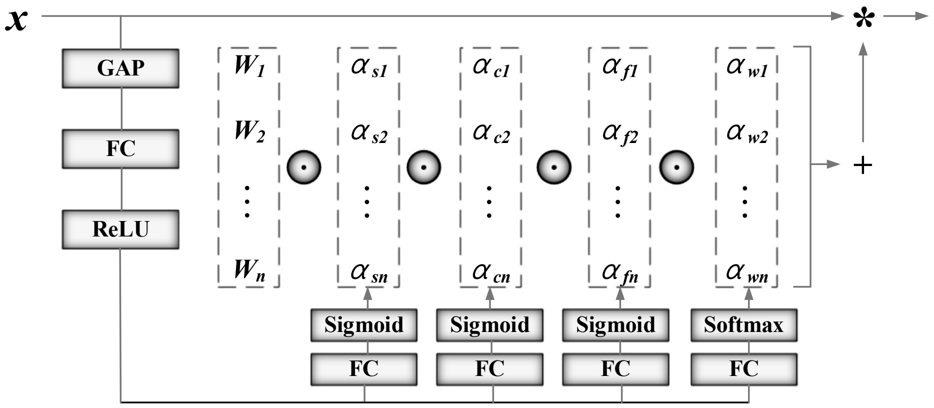

Figure 8. A schematic of an omni-dimensional dynamic convolution [46]. GAP: global average pooling; FC: fully connected layer; W_i: convolution kernel; α_{si}: kernel space dimension position; α_{ci}: channel mode of input channel dimension; α_{fi}: filter mode of output channel dimension; α_{wi}: the kernel dimension of the convolution kernel space and the kernel dimension of the space where i ranges from 1 to n; the symbols '+' and '*' represent addition and convolution operations respectively..

$$y=\left(\alpha_{\omega 1}\odot\alpha_{f1}\odot\alpha_{c1}\odot\alpha_{s1}\odot W_1+\cdots+\alpha_{\omega n}\odot\alpha_{fn}\odot\alpha_{cn}\odot\alpha_{sn}\odot W_{1n}\right)*x \tag{9}$$

Within the ODConv framework, the attention mechanism operates across four dimensions of the kernel space: position, channel, filter, and kernel dimensions. These complementary attention mechanisms contribute to a convolution operation that incorporates different attention at each step, thereby influencing each dimension of the input in distinct ways. This approach facilitates an enhanced utilization of spatial information and yields superior performance in capturing intricate contextual details. By integrating ODConv into the slim-neck architecture, the detrimental effects stemming from the extensive usage of lightweight convolution kernels and depthwise convolutions are effectively alleviated. This fortifies the model's ability to capture both contextual information and spatial dimensionality, thereby enhancing the accuracy and robustness of the detection task.

3.3.3. Spatial Pyramid Pooling

In conventional convolutional neural networks (CNNs), it is a common practice to resize input images to a fixed dimension to facilitate efficient data processing. However, practical applications often encounter images of various sizes, which impose constraints on traditional CNNs. To overcome this limitation, He et al. [31] introduced a technique termed SPP. SPP aims to convert feature maps of arbitrary sizes into fixed-sized feature vectors, thereby enhancing the performance of tasks such as image classification and object detection. The approach of SPP involves pooling image features at multiple scales by constructing grids, allowing for the capture of contextual information across different scales. Despite the notable success in handling inputs of varying sizes, SPP has certain limitations. When confronted with images containing intricate structures or local objects, the utilization of fixed-sized grids for feature pooling may result in the loss of spatial information. In addition, SPP entails computations across the entire feature map, which gives rise to high computational complexity and time consumption. Therefore, these factors render SPP less suitable for real-time scenarios or resource-constrained environments.

To address these challenges, several advancements have been made by researchers to enhance the SPP method. These improvements involve the proposal of more efficient pooling algorithms, which effectively reduce computational complexity and time overheads. These enhanced approaches [47–49] empower SPP-based models to effectively handle inputs of different sizes while achieving superior performance in tasks such as image classification and object detection.

To further enhance efficiency and speed, we conducted an exploration of different types of SPP blocks, including SPPF [47] and SimSPPF [48]. SimSPPF [49] is a simplified variant of SPPF. To evaluate their performance, we compared SPPF and SimSPPF with the SPPCSPC employed in the original YOLOv7 model. A randomly sized input tensor with dimensions [8,1024,20,20] was generated, and it was passed through these three structures to obtain output tensors with dimensions [8,512,20,20]. The inference time required for 100 iterations of computation was measured and the experimental results are presented in Table 3. Notably, SPPCSPC exhibited the highest number of parameters and the longest inference time, while SPPF and SimSPPF had an equal number of parameters. However, SimSPPF demonstrated a shorter inference time. To achieve faster inference speeds, this study opted to utilize SimSPPF instead of SPPCSPC in the original YOLOv7 model. In the subsequent experiments, we will provide further evidence showcasing the superiority of selecting SimSPPF.

Table 3. Inference speed comparison of different spatial pyramid pooling structures.

Input Size	Output Size	Structure	Params	Inference Speed (s) [1]
[8,1024,20,20]	[8,512,20,20]	SPPF [47]	1,574,912	0.253
		SimSPPF [48]	1,574,912	0.176
		SPPCSPC [49]	7,609,344	1.498

[1] Calculate the inference time required to iterate 100 rounds.

4. Experimental Results and Analysis

4.1. Model Training and Evaluation

4.1.1. Training Platform and Network Initialization

This study adopted a specific set of platform parameters for model training and testing purposes. Detailed information pertaining to these parameters can be found in Table 4. The PyTorch deep learning framework was selected as the primary platform, accompanied by the Python 3.8 programming language. Acceleration and image processing capabilities were provided through the use of libraries such as CUDA 1.10.0, cuDNN, and OpenCV. During the training phase, the input image size was fixed at 640 pixels × 640 pixels. The model underwent training for a total of 300 epochs, with a batch size of 16 for each epoch. Stochastic gradient descent (SGD) was employed as the optimization function, utilizing a momentum factor of 0.937. The initial learning rate was set to 0.01, and a weight decay coefficient of 0.0005 was applied to regulate the model's complexity and prevent overfitting.

Table 4. Hardware and software configuration.

Configuration	
CPU	Intel Core i5-12400F CPU@2.5 GHz
GPU	GeForce RTX 3060 12 G GDDR6
Operating System	Window11 64 bit
Deep Learning Framework	Pytorch

The chosen set of parameters mentioned above played a crucial role in effectively optimizing the model's weights and parameters during the training process. This optimization led to notable improvements in the model's performance and accuracy for object detection tasks. The selection of these specific platform parameters was guided by a series of iterative experiments and careful adjustments. The goal was to achieve the optimal performance of the model on the given task and dataset.

4.1.2. Model Evaluation Method and Results

This study employed a comprehensive set of metrics to evaluate the performance of passion fruit object detection. These metrics included precision (P), average precision

(AP), recall (R), and mean average precision (mAP). The calculation formulas for these metrics are provided below:

$$\mathrm{P} = \frac{T_P}{T_P + F_P} \tag{10}$$

$$\mathrm{R} = \frac{T_P}{T_P + F_N} \tag{11}$$

$$\mathrm{AP} = \int_0^1 \mathrm{P}(\mathrm{R})d\mathrm{R} \tag{12}$$

$$\mathrm{mAP} = \frac{1}{nc}\sum_{k=1}^{nc} \mathrm{AP}_k \tag{13}$$

In passion fruit object detection, T_P represents the number of predicted boxes classified as positive and exhibiting an overlap with the ground truth boxes. F_P represents the number of predicted boxes classified as positive but lacking any overlap with the ground truth boxes. F_N represents the number of ground truth boxes that are not predicted. P serves to assess the model's accuracy in predicting diverse varieties of passion fruit, denoting the ratio of accurately predicted positive samples to the total number of predicted positive samples. R measures the model's ability to detect actual positive samples, ensuring that the model can predict all targets within different types of passion fruit. *nc* denotes the number of categories within the samples, while *k* pertains to a specific category under evaluation. AP refers to the area beneath the precision–recall curve, serving as a metric for measuring the passion fruit object detection model's ability to detect different categories. mAP signifies the average of AP values across all categories and provides a comprehensive evaluation of the model's detection performance for passion fruit across different categories. Additionally, the real-time detection capability of the model plays a vital role in passion fruit harvesting. Therefore, this study also incorporated inference time and the number of parameters as performance metrics to evaluate the additional aspects of the passion fruit object detection model. The utilization of these metrics enabled a holistic assessment of the model's precision, recall, and detection aptitude in the performance of passion fruit recognition tasks, while concurrently taking into account crucial factors like real-time performance and parameter quantity.

To ensure the reliability and stability of evaluating the proposed FSOne-YOLOv7 model, we employed a robust five-fold cross-validation strategy. This approach involved dividing the training dataset into five subsets, with each subset used once and only once as a validation set in rotation, while the remaining four subsets formed the corresponding training sets. By utilizing this method, we ensured both the randomness and balance of the dataset, effectively addressing biases arising from uneven data distribution. Table 5 showcases the evaluation metrics of the FSOne-YOLOv7 model on the test set for each fold, including P, R, and mAP, alongside the mean and standard deviation values for each metric. The final two rows present the mean and standard deviation values across all five folds, providing a comprehensive assessment of the FSOne-YOLOv7 model's performance.

Table 5. Experimental results of cross-validation for FSOne-YOLOv7.

Fold	P (%)	R (%)	AP (%)		mAP (%)
			pf	rpf	
Fold1	83.6	82.3	86.2	92.9	89.55
Fold2	84.4	83.0	87.3	93.4	90.35
Fold3	84.4	81.9	87.1	92.7	89.90
Fold4	82.5	84.5	86.8	93.5	90.15
Fold5	86.8	79.9	87.7	93.2	90.45
Average Value	84.34	82.32	87.02	93.14	90.08
Standard Deviation	1.58	1.68	0.56	0.34	0.36

Based on the results of five-fold cross-validation, our model achieved an average P of 84.34% with a standard deviation of 1.58. The mAP was 90.08% with a standard deviation of 0.36. The average R was 82.32% with a standard deviation of 1.68. These results demonstrate the model's stability and consistency in object detection, indicating its favorable performance. The five-fold cross-validation evaluation confirmed the robustness and reliability of our model.

4.2. Feature Information Analysis Experiment

When evaluating the performance of ShuffleOne and the slim-neck in FSOne-YOLOv7, we employed gradient-weighted class activation mapping (Grad-CAM) [50] heatmaps as a means of analysis. Grad-CAM is a gradient-based method for generating class activation maps that assign importance values to individual neurons based on the flow of gradient information into the final convolutional layer of a CNN. This process facilitates attention-based decision-making regarding specific regions. Specifically, by computing the gradients of the target class and element-wise multiplying them with their corresponding feature maps, we derived the weights for the activation map. These weights were then spatially averaged, underwent nonlinear processing through rectified linear units (ReLU), and ultimately produced the class activation map.

The Grad-CAM method serves as a valuable tool for researchers to visually comprehend the relevance of individual neurons within a convolutional neural network, specifically in relation to their impact on network prediction outcomes. By examining the generated heatmap, researchers can gain intuitive insights into the model's focus on specific regions and analyze its performance across different categories and locations. The visualizations of Grad-CAM heatmaps provide a deeper understanding of how the backbone and slim-neck structures within the FSOne-YOLOv7 model operate in passion fruit recognition tasks, offering valuable insights for further refinement and optimization. The computation of Grad-CAM can be expressed by the following equation:

$$L_{\text{Grad-CAM}}^{C} = ReLU\left(\sum_{k} \alpha_{k}^{c} A^{k}\right) \tag{14}$$

where $L_{\text{Grad-CAM}}^{C}$ signifies the Grad-CAM of class c and $ReLU$ denotes the rectified linear unit operation, which sets negative values in the activation map to 0. A represents the feature map and A^k refers to the k channel data within the feature map A. Additionally, α_k^c denotes the importance weight of the feature map Ak for the target class c, which is computed based on gradient information, as depicted in the following manner:

$$\alpha_{k}^{c} = \frac{1}{Z} \sum_{u} \sum_{v} \frac{\partial y^{c}}{\partial A_{uv}^{k}} \tag{15}$$

Herein, $\frac{\partial y^c}{\partial A_{uv}^k}$ represents the gradient of the target class c with respect to the element A_{uv}^k, y^c denotes the score predicted by the model for class c prior to applying the Softmax function, and A_{uv}^k represents the data within the feature map A at channel k with coordinates (u,v). The normalization factor Z is computed as the global average pooling of the gradients. Subsequently, the ReLU-activated and weighted feature map is upsampled to match the dimensions of the input image. The upsampled class activation map is then normalized for visualization purposes. In this study, the Grad-CAM technique was employed to generate heatmaps, which were superimposed onto the original images. This visualization approach enabled a graphical representation of the model's focus on passion fruit, with regions displaying darker red hues indicating greater attention by the model.

Similar to YOLOv7, the FSOne-YOLOv7 model also generates three levels of feature outputs: shallow, medium, and deep. However, both models differ in their emphasis on these feature outputs when detecting mature and immature passion fruit. In the case of mature passion fruit, both models concentrate their feature attention on the shallow- and medium-level outputs. However, both models differ in their emphasis on these feature outputs when detecting mature and immature passion fruit. In the case of mature passion fruit, both models concentrate their feature attention on the shallow- and medium-level outputs. However, when it comes to immature passion fruit, FSOne-YOLOv7 places greater emphasis on the shallow- and medium-level feature outputs, as opposed to the medium and deep-level ones, which distinguishes it from YOLOv7. This heightened sensitivity of FSOne-YOLOv7 to the specific details of the target contributes to the improved accuracy of Grad-CAM heatmaps in capturing contour edges, colors, and other intricate features of passion fruit. Figure 9 showcases the results of YOLOv7 and FSOne-YOLOv7 on different levels of feature outputs. The top section of the figure presents visualizations of Grad-CAM heatmaps for detecting immature passion fruit, while the bottom section displays the results for detecting mature passion fruit. Observing Figure 9, it becomes evident that FSOne-YOLOv7 outperforms the YOLOv7 model by effectively filtering out background information and optimizing computational resources for the detection of immature passion fruit. Additionally, when detecting mature passion fruit, FSOne-YOLOv7 demonstrates a superior ability to accurately and precisely locate the fruit's center position within the shallow-level feature outputs, thereby expediting the localization process.

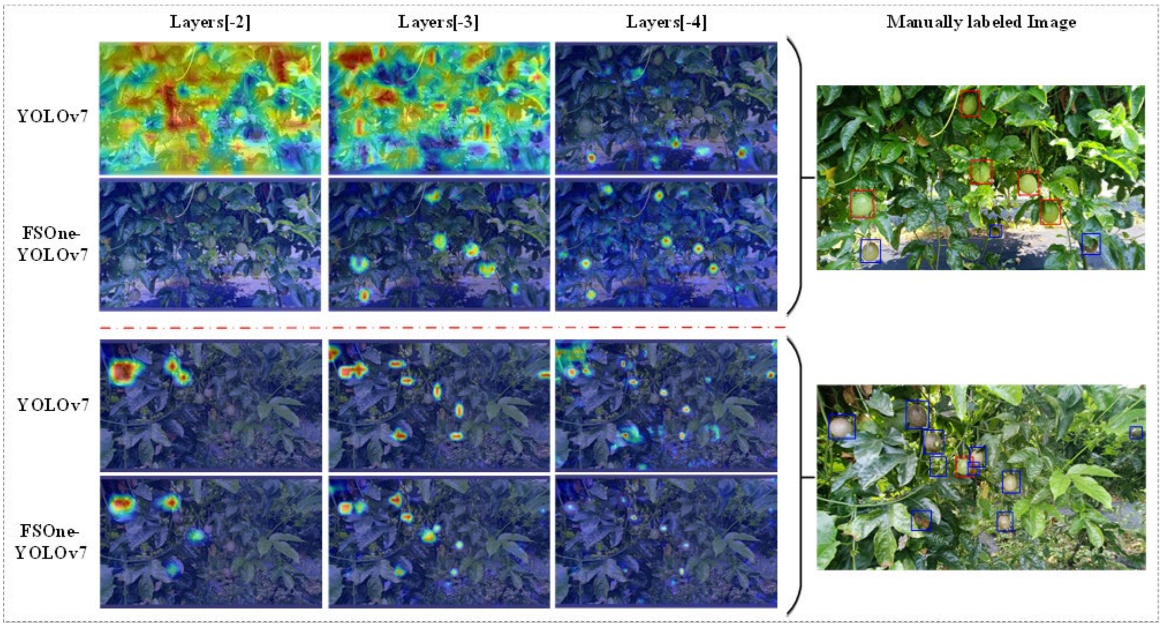

Figure 9. Comparison of Grad-CAM heatmap visualizations for different output scales between YOLOv7 and FSOne-YOLOv7. Layers [−2], [−3], and [−4] represent the deep, middle, and shallow feature outputs of the model.

To demonstrate the efficacy of the FSOne-YOLOv7 model, we conducted a visual comparison of feature maps using the Grad-CAM technique between YOLOv7 and FSOne-YOLOv7. Figure 10 presents enhanced Grad-CAM heatmaps with a specific focus on the target passion fruit. In the case of immature passion fruit, which bears a resemblance to the background color, the models tended to expand the attention area extensively to capture intricate details such as contour edges and color characteristics. When YOLOv7 detected immature passion fruit, its feature map unavoidably exhibited increased attention to the background information, both in terms of coverage and intensity. Conversely, the feature map generated by FSOne-YOLOv7 achieved superior coverage of the passion fruit region while effectively filtering out background features that bore a similarity to the fruit's color. When detecting ripe passion fruit, FSOne-YOLOv7's feature map primarily concentrated on the central region of the fruit, facilitating precise localization and accurate bounding box placement. In contrast, YOLOv7's feature map displayed a broader distribution of attention across the fruit, encompassing more background features. This increased inclusion of background features consumed more computational resources, while the concentrated feature information in FSOne-YOLOv7 enhanced the accuracy and stability of passion fruit detection. Therefore, in comparison to YOLOv7, FSOne-YOLOv7 exhibited higher detection accuracy and faster inference speeds in detecting passion fruit targets. These results serve to further substantiate the advantages offered by the FSOne-YOLOv7 model.

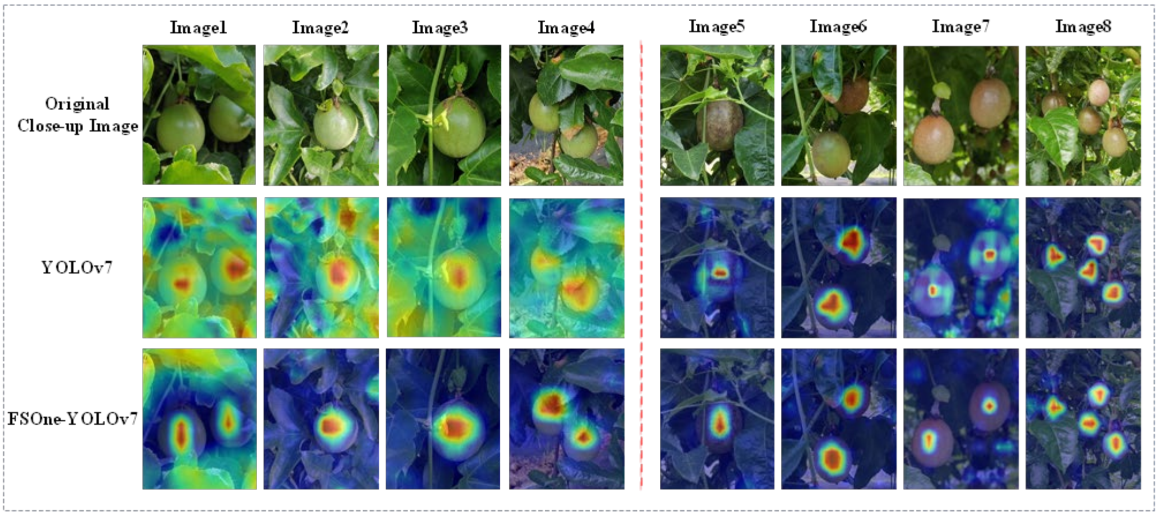

Figure 10. YOLOv7 and FSOne-YOLOv7 visualization comparison close-up images. The first row consists of close-up images of passion fruits. Image1 to Image4 depict immature passion fruit, while Image5 to Image8 represent ripe passion fruit. The second and third rows display the optimal Grad-CAM heatmaps from the output feature layers of YOLOv7 and FSOne-YOLOv7, respectively.

To offer a more comprehensive assessment of the benefits of ShuffleOne and the slim-neck module in the domains of feature extraction and fusion, a comparative analysis between YOLOv7 and FSOne-YOLOv7 models is presented in Figure 11. The utilization of Grad-CAM heatmaps revealed that FSOne-YOLOv7 surpasses YOLOv7 in terms of its proficiency in feature extraction. The backbone component of the YOLOv7 model exhibited shortcomings in effectively eliminating extraneous background information. Conversely, the FSOne-YOLOv7 model surmounted this limitation by adeptly filtering out superfluous background details, thereby accentuating the distinctive appearance features specific to passion fruit. These findings substantiate the notion that ShuffleOne confers benefits upon feature extraction while augmenting the interpretability of the exhibited results. Appraising the results presented for each block, it becomes evident that the FSOne-YOOv7 model initially emphasized the elliptical visual attributes inherent to passion fruit, gradually discarding background traits that resembled the fruit's color, progressively intensifying the attention surrounding the fruit area and ensuring maximal concentration of attention upon the fruit. Finally, by capitalizing on the fruit's appearance characteristics, attention was directed towards the salient regions of interest. In addition, the distinct modules within slim-neck manifested divergent influences on feature fusion. Slim-neck1 manifested commendable results in fusion, whereas slim-neck2 introduced ODConv to bolster the fusion of contextual information, thereby expanding and focalizing attention upon both unripe and ripe regions of passion fruit. Moreover, slim-neck3 further optimized the results of fusion by accentuating the emphasis on the central position of the fruit. These results unequivocally demonstrate the pivotal roles played by ShuffleOne and slim-neck in feature extraction and fusion, substantiating their capacity to enhance the performance of the FSOne-YOOv7 model.

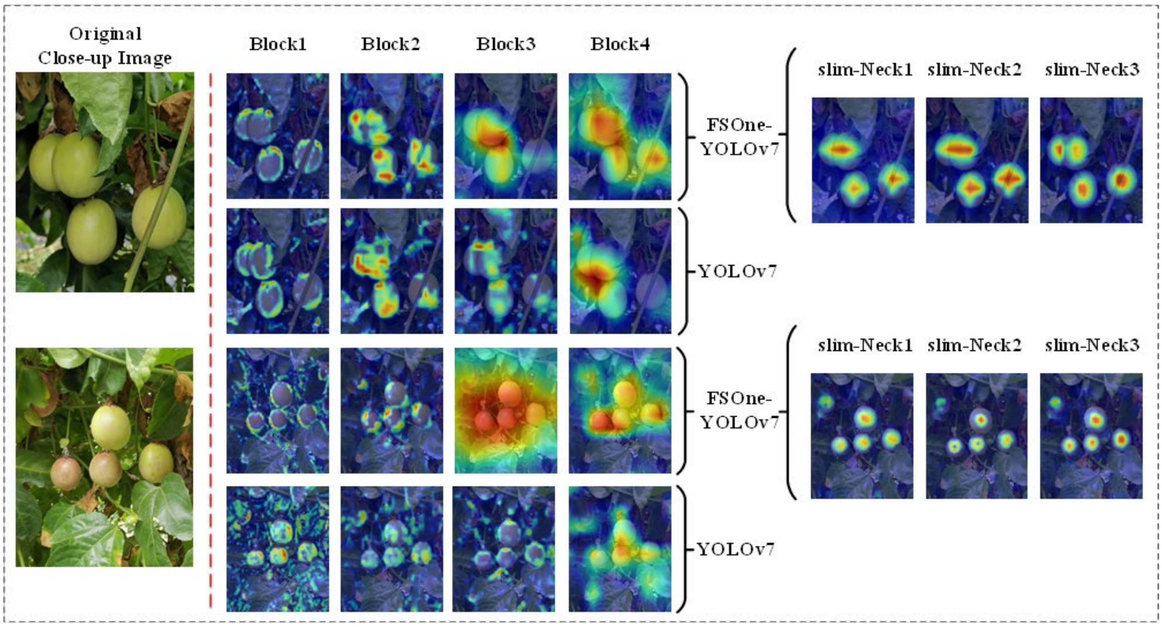

Figure 11. Visual effect diagram of the effect of different modules on the model. The left section displays close-up images of the original passion fruit. In the middle section, each row represents the visualization results of a model, and each column represents the Grad-CAM effect of each block layer in the backbone of FSOne-YOLOv7 and YOLOv7. On the right side, the optimal Grad-CAM effect images are shown for FSOne-YOLOv7 using different slim-neck configurations (slim-neck1 represents the introduction of only C2Faster, slim-neck2 represents the introduction of C2Faster and ODConv, and slim-neck3 represents the introduction of C2Faster, ODConv, and SimSPPF).

In conclusion, the incorporation of ShuffleOne yielded substantial benefits in terms of proficient feature extraction, thereby enabling the FSOne-YOOv7 model to efficiently eliminate extraneous background information and focus on essential attributes specific to passion fruit. The distinct modules within the slim-neck exert diverse influences on feature fusion, thereby optimizing the attention regions of the model and facilitating the more precise detection of both unripe and ripe passion fruit. Consequently, the FSOne-YOLOv7 model demonstrated superior performance in feature extraction and fusion, resulting in heightened detection accuracy and accelerated inference speeds during the detection of passion fruit.

4.3. Ablation Experiments and Parameter Design of the Model

To ensure the objectivity and accuracy of the evaluation, the test set, which comprised data that were not utilized during model training or validation, served as the benchmark for the final performance assessment of the passion fruit detection model. The training set was employed to determine the optimal hyperparameters and model structure. In this study, a comprehensive set of experiments was carried out on the test set to showcase the superiority of all modules. During the testing phase, the input images were standardized to dimensions of 640 × 640 pixels and an IoU (intersection over union) threshold of 0.65 was applied.

The selection of the backbone network in this study took into account both computational efficiency and trainability as these factors are crucial for practical applications. In particular, a backbone network with fewer parameters and computational requirements is deemed more suitable. Therefore, ShuffleOne was chosen as the backbone network for the model, leveraging its advantages in computational efficiency and trainability. In addition, a comprehensive set of ablation experiments was conducted on the backbone and slim-neck modules of the model. These experiments specifically focused on ShuffleOne, C2Faster module, SimSPPF module, and ODConv module. The primary objective was to assess the contribution of each module to the model's performance and determine their respective roles in the task of passion fruit detection. Through a comparative analysis of the experimental results, the effectiveness and performance enhancements brought about by the model could be further validated. In conclusion, a series of experiments and ablation analyses were performed on the test set, serving as an objective means to evaluate the performance of the passion fruit detection model and demonstrate the superiority of the various modules. These experimental findings provide valuable insights for future model refinement and optimization endeavors.

The experimental results presented in Table 6 showcase the utilization of ShuffleOne as the foundational framework for YOLOv7, which culminated in a reduction in model parameters to 25.3 M. Moreover, this approach yielded a notable 3.45 percentage point increase in mAP and a reduction of 19.6% in single-frame inference time compared to the original YOLOv7 configuration. Of particular significance is the discernible enhancement in the detection accuracy of nascent passion fruit. By incorporating the C2Faster module within the neck (using default values of d = 1 and *p* = 8), the model parameters were further diminished to 19.5 M, accompanied by a 3 percentage point elevation in mAP and a decrease of 25.1% in single-frame inference time. However, the introduction of the C2Faster module, despite its positive impact on the model's inference speed, resulted in diminished detection performance for immature passion fruit when contrasted with the model that exclusively employed ShuffleOne as the foundational framework.

Table 6. Effects of each module on test dataset.

YOLOv7	ShuffleOne	Slim-Neck			Param	AP (%)		mAP	Speed
		C2Faster	ODConv	SimSPPF	(M)	pf	rpf	(%)	(ms) [1]
√					36.5	81.1	90.2	85.60	9.2
√	√				25.3	86.4	91.7	89.05 (+3.45)	7.4
√	√	√			19.5	84.7	92.5	88.60 (+3.00)	6.9
√	√	√	√		19.6	86.5	92.9	89.70 (+4.10)	7.2
√	√	√	√	√	13.6	87.7	93.2	90.45 (+4.85)	6.5

[1] Average inference speed per image for the test set.

Henceforth, the ODConv module is introduced with the purpose of augmenting the model's capacity to comprehend contextual information, thereby engendering an ameliorated detection performance for fledgling and mature passion fruit. Remarkably, in comparison to the initial YOLOv7 configuration, a notable upsurge of 4.1 percentage points in mAP was observed. This introduction of ODConv served as a compensatory measure to mitigate the impact of the C2Faster module on the model's accuracy. Nevertheless, it should be acknowledged that ODConv entails additional parameters and computational costs when juxtaposed with conventional convolutions (Conv), thus resulting in an increase in the inference time of the model. To tackle this issue, we further incorporated the SimSPPF structure, which exhibited a reduction in parameters, thus propelling enhancements in both model inference speed and the adept utilization of feature space information. Intriguingly, in contrast to the original YOLOv7 model, an augmented mAP of 4.85 percentage points was witnessed, accompanied by a substantial decrease in single-frame inference time of 28.3%.

The incorporation of the spatial pyramid pooling (SPP) structure facilitated the extraction of features at various receptive field sizes, thereby concurrently bolstering the inference speed of the model. The results presented in Table 7 substantiate the claim that different SPP structures engender diverse effects on the model, with SimSPPF yielding the most favorable performance overall. In the FSOne-YOLOv7 model, employing SimSPPF and SPPF, respectively, led to an equivalent number of parameters (13.6 M). However, the utilization of SimSPPF yielded superior results in terms of mAP and lower single-frame inference times. Within the FSOne-YOLOv7 model, utilizing SPPCSPCS produced a detection AP that surpassed that of SimSPPF by 0.1 percentage points specifically for mature passion fruit. Nevertheless, this alternative led to a reduction of 0.7 percentage points in mAP in comparison to SimSPPF, accompanied by longer single-frame detection times. By electing to adopt SimSPPF, which offers superior detection performance, as a replacement for the SPPCSPCS structure in the original YOLOv7 model, further enhancements could be attained in terms of the model's inference speed and its adeptness in leveraging feature space information.

Table 7. Comparison of pyramid pooling structures.

Model	SPPF	SimSPPF	SPPCSPC	Param	AP (%)		mAP	Speed
					pf	rpf	(%)	(ms) [1]
FSOne-YOLOv7	√			13.6 M	86.2	92.9	89.55	6.9
		√		13.6 M	87.7	93.2	90.45	6.5
			√	19.6 M	86.2	93.3	89.75	7.3

[1] Average inference speed per image for the test set.

Predicated on the aforementioned considerations, a brief ablation study was carried out to investigate the impact of varying the number of FasterNet blocks (n) in the C2Faster module and the ratio (p) of PConv. The study demonstrated that when $n = 1$ and $p = 8$, the model attained superior mAP and accelerated inference speed. The detailed findings are summarized in Table 8. It was observed, based on prior knowledge and expertise, that employing a large value for p could result in the degradation of PConv, rendering it akin to a traditional Conv. Additionally, increasing the number of FasterNet blocks (d) in the C2Faster module led to the inadequate extraction of spatial features and prolonged inference times. The experimental results validate that the utilization of the slim-neck module further enhances the computational efficiency and inference speed of the passion fruit object detection model, thereby bolstering its performance.

Table 8. Ablation study on the number of FasterNet blocks (d) and the partial ratio (p) of PConv.

Model	Number of FasterNet Blocks (n)	Partial Ratio (p)	AP (%)		mAP (%)	Speed (ms) [1]
			pf	rpf		
FSOne-YOLOv7	1	2	86.6	92.6	89.60	6.7
	1	4	87.2	93.2	90.20	6.5
	1	8	87.7	93.2	90.45	6.5
	2	2	86.7	92.6	89.65	7.3
	2	4	87.1	92.8	89.95	7.1
	2	8	85.9	92.8	89.40	6.9

[1] Average inference speed per image for the test set.

4.4. Comparison Experiment with Different YOLO Models

To demonstrate the superiority of FSOne-YOLOv7 for the passion fruit detection task, a comprehensive comparison was conducted (see Table 9) involving YOLO versions (specifically YOLOv5l, YOLOv7, and YOLOv8l) alongside the proposed FSOne-YOLOv7 model specifically designed for passion fruit detection. The results indicate that FSOne-YOLOv7 outperforms other models in both detection accuracy and speed.

Compared to the earlier YOLOv5l version, YOLOv7 exhibited smaller model parameters and weight size, resulting in improved detection speeds. Additionally, when compared to the recent YOLOv8l, YOLOv7 demonstrated superior performance in terms of both detection accuracy and speed. Rigorous empirical analysis using a passion fruit detection task and dataset highlighted the substantial advantages and potential of YOLOv7 in this specific context. Consequently, the selection of YOLOv7 as the foundational research model aligned perfectly with the practical requirements of this study and significantly contributed to the establishment of a robust and reliable model.

Table 9. Comparative analysis of FSOne-YOLOv7 with different versions of YOLO models.

Models	Param	Weight Size	P (%)		AP (%)		mAP (%)	Speed (ms) [1]
			pf	rpf	pf	rpf		
YOLOv5l	46.1 M	92.9 MB	79.6	86.3	82.4	89.9	86.15	12.4
YOLOv7	36.5 M	75.6 MB	78.8	86.1	81.1	90.2	85.60	9.2

YOLOv8l	43.6 M	87.7 MB	78.9	86.9	81.0	89.3	85.15	15.6
FSOne-YOLOv7	13.6 M	29.2 MB	82.7	91.2	87.7	93.2	90.45	6.5

[1] Average inference speed per image for the test set.

In comparison to the original YOLOv7, FSOne-YOLOv7 exhibited a reduced parameter count of 13.6 M and weight volume of 29.2 M, amounting to approximately 2/5 of YOLOv7. Notably, FSOne-YOLOv7 demonstrated enhancements across various facets of detection performance. The precision rates for immature and ripe passion fruit were measured at 82.7% and 91.2%, respectively, denoting an improvement of 3.9% and 5.1%, respectively.–The AP for immature and ripe passion fruit recorded values of 87.7% and 93.2%, respectively, indicating individual improvements of 6.6% and 3%, respectfully. The mAP reached 90.45%, showcasing an advancement of 4.85%. Lastly, the model achieved a detection speed of 58.2 fps, representing an approximate boost of 35.7% in speed.

Figure 12 illustrates the comparative results of the ablation detection experiments conducted between ShuffleOne and slim-neck. Image1 and Image2 comprise 3 and 5 mature passion fruit targets, respectively, while Image3 and Image4 consist of a combined total of 7 unripe passion fruit targets and 16 mature passion fruit targets. The quantities and proportions of unripe and mature passion fruit differ across the four images. In Image1 and Image2, which encompass a limited number of passion fruit targets, both YOLOv7 and One-YOLOv7 failed to detect the same mature passion fruit target due to substantial occlusion caused by leaves. In contrast, FSOne-YOLOv7 successfully identified all passion fruit targets. In Image3 and Image4, which contain a greater number of passion fruit targets, YOLOv7 missed a cumulative total of 4 mature passion fruits and erroneously classified 5 unripe passion fruits as mature. One-YOLOv7 missed 3 mature passion fruits and 1 unripe passion fruit, with only one misclassification of unripe passion fruit as mature. Conversely, FSOne-YOLOv7 only failed to detect 3 mature passion fruits. The detection results reveal that YOLOv7 is susceptible to overlooking or misclassifying passion fruit targets, particularly when multiple targets are present. The introduction of ShuffleOne into YOLOv7 enhanced the model's ability to extract features and diminished the frequency of misclassifications. Moreover, the integration of ShuffleOne and slim-neck into YOLOv7 reinforced the model's feature extraction and fusion abilities, enhanced its detection performance for unripe passion fruit, and further reduced the instances of missed detections. In conclusion, in scenarios involving a limited or substantial number of passion fruits, FSOne-YOLOv7 with ShuffleOne and slim-neck exhibited superior resilience and generalization capabilities.

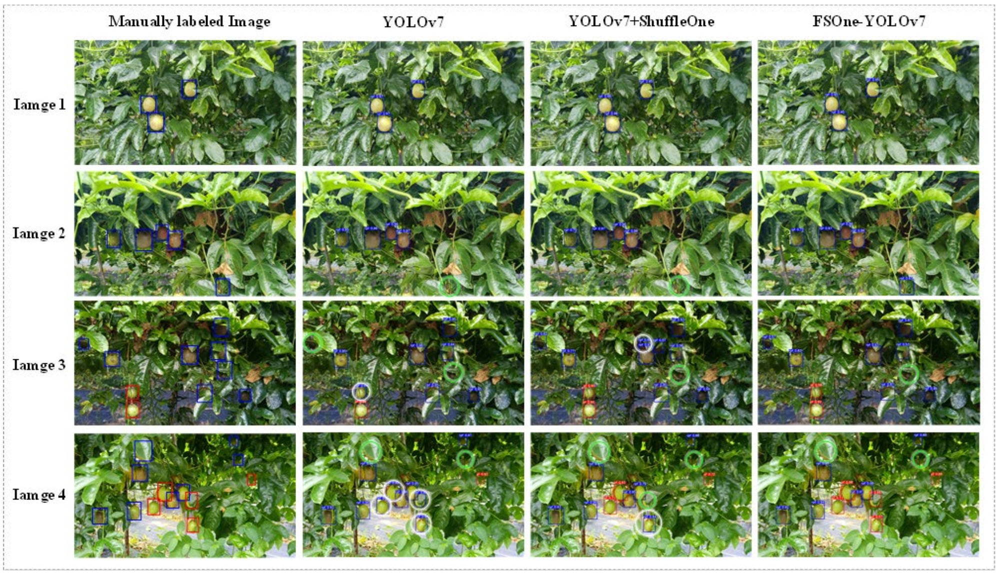

Figure 12. A comparison of the ablation detection results between ShuffleOne and slim-neck. The red bounding boxes represent the predicted results for unripe passion fruit, while the blue bounding

boxes represent the predicted results for ripe passion fruit. The green circles indicate missed passion fruit targets and the white circles represent false detections of passion fruit. "YOLOv7 + ShuffleOne" refers to the introduction of ShuffleOne to YOLOv7, while FSOne-YOLOv7 represents the incorporation of both ShuffleOne and slim-neck into YOLOv7.

4.5. Comparative Experiments of Different Detection Methods

Comparisons were performed between FSOne-YOLOv7 and other object detection techniques, namely, Faster R-CNN and SSD, in order to assess the practicality and adaptability of FSOne-YOLOv7 in various application scenarios. The results, as presented in Table 10, reveal that the proposed FSOne-YOLOv7 model showcased the most impressive performance in terms of detection accuracy. With a parameter size of 13.6 M, FSOne-YOLOv7 is approximately 1/10th and 1/2 the size of Faster R-CNN and SSD, respectively. The weight size of the model is a mere 29.2 MB. In the testing dataset consisting of 400 images, the FSOne-YOLOv7 model achieved a mAP of 90.45% and a detection frame rate of 58.2 fps. Comparing these results to Faster R-CNN and SSD, the proposed FSOne-YOLOv7 model exhibited a mAP improvement of 10 percentage points and 4.4 percentage points, respectively, while also enhancing the detection speed by roughly 2.6 times and 1.5 times, respectively. Among the five detection methods, the model attained the highest P of 82.7% and 91.2% for detecting unripe and fully mature passion fruit, respectively.

The comparison of passion fruit detection performance between FSOne-YOLOv7 and different detection methods is presented in Figure 13. Image1, Image2, and Image3 are in a backlit environment and consist of a limited number of 3, 5, and 8 passion fruit targets, respectively. In Image1, due to the presence of backlighting, Faster-RCNN erroneously identified a background leaf as mature passion fruit and misclassified an unripe passion fruit as mature. Additionally, SSD failed to detect one mature passion fruit. In Image2 and Image3, all models demonstrated satisfactory detection performance, with only one instance of Faster-RCNN missing a mature passion fruit in Image3. Image4 is in a backlit environment comprising 8 unripe passion fruits and 11 mature passion fruits. Faster-RCNN mistakenly classified a background element as an unripe passion fruit, while SSD missed two unripe passion fruits and two mature passion fruits. Similarly, Image5 was also taken in a backlit environment during the evening, which is why it appears to have slightly insufficient lighting. Faster-RCNN overlooked one mature passion fruit and exhibited two false positive detections. SSD failed to detect two unripe passion fruits and three mature passion fruits, while also misclassifying one unripe passion fruit as mature. Similarly, FSOne-YOLOv7 missed two mature passion fruits and misclassified one unripe passion fruit as mature.

Table 10. Comprehensive comparison results of FSOne-YOLOv7 with different detection methods.

Models	**Param**	**Weight Size**	**P (%)**		**AP (%)**		**mAP**	**fps** [1]
	(M)	(MB)	**pf**	**rpf**	**pf**	**rpf**	**(%)**	
FasterRCNN	137.1	546.9	68.1	78.1	74.7	86.2	80.45	16.2
SSD	26.3	105.2	75.9	82.8	81.2	90.9	86.05	23.3
FSOne-YOLOv7	13.6	29.2	82.7	91.2	87.7	93.2	90.45	58.2

[1] The average number of detection frames per image in the test set.

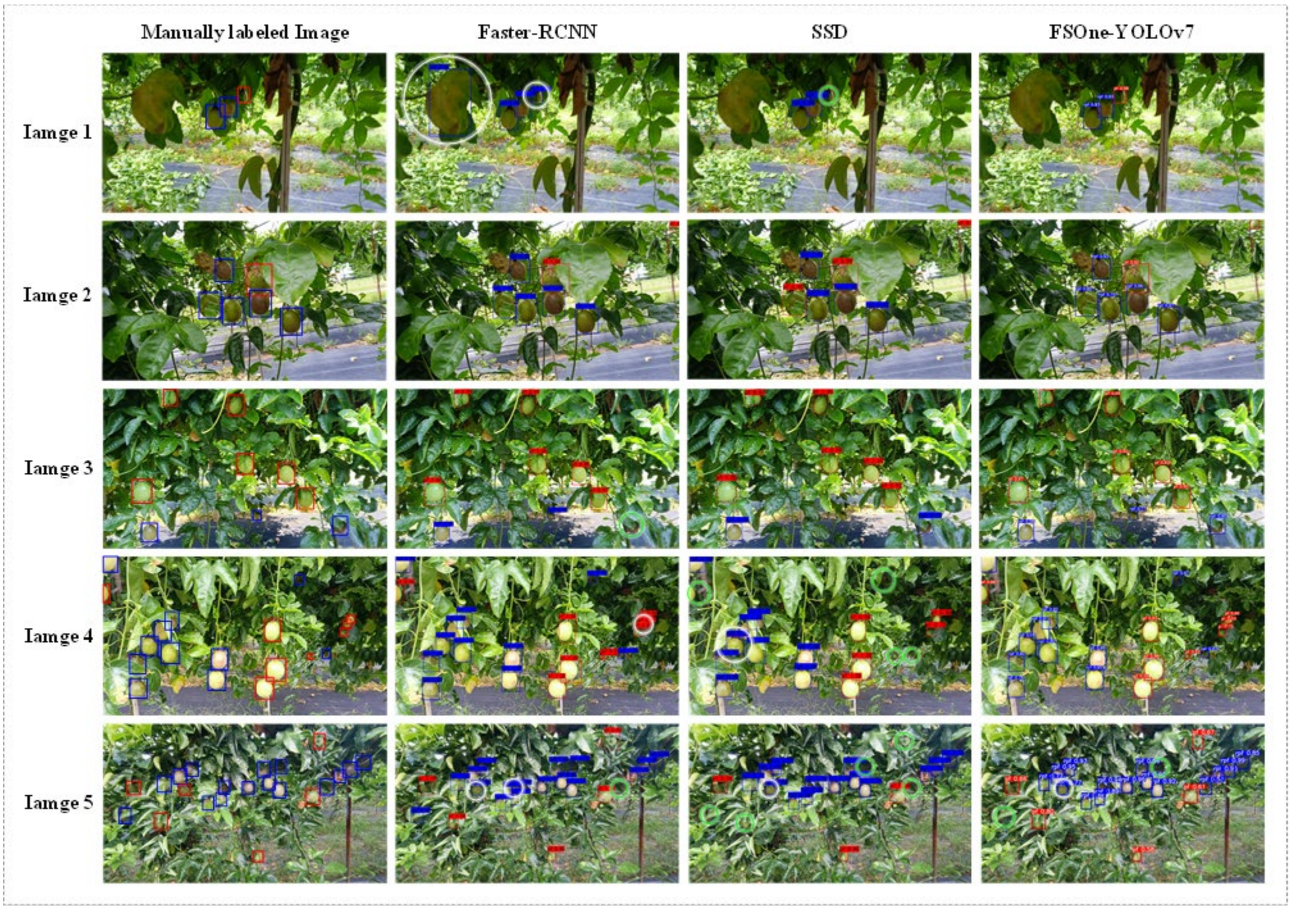

Figure 13. Comparison of detection performance among different methods. Red rectangular boxes represent the predicted results for unripe passion fruit, while blue rectangular boxes represent the predicted results for mature passion fruit. The green circles indicate missed passion fruit targets and the white circles represent falsely detected passion fruit objects.

FSOne-YOLOv7 adopted a lightweight ShuffleOne backbone network and employed a multi-branch structure during training to enhance feature extraction capabilities. During inference, it sped up the process by reparameterizing into a single-branch structure, resulting in smaller parameters and an overall smaller model size. This enabled FSOne-YOLOv7 to effectively extract feature information from passion fruit. In comparison, SSD and Faster RCNN used deeper VGG backbone networks, leading to slower detection speeds and a higher likelihood of losing fine details in complex environments, thereby affecting target localization and classification accuracy. When considering the detection of different quantities of passion fruit, the two-stage detection method, Faster-RCNN, exhibited a lower false-negative rate but slower detection speed. In backlit environments, Faster-RCNN showed more false positive detections. On the other hand, the one-stage detection method, SSD, displayed the fewest false-positive detections but tended to have the most false negative detections in dense and large-quantity scenarios. Compared to Faster-RCNN and SSD, FSOne-YOLOv7 demonstrated superior detection accuracy and speed in backlit environments and also performed well when detecting a large number of or densely packed passion fruit targets.

Both FSOne-YOLOv7 and SSD belong to the one-stage target detection algorithm, capable of directly predicting the positions and categories of targets. In contrast, Faster-RCNN adopts a two-stage detection process, first using the region proposal network (RPN) to generate candidate target boxes and then conducting classification and regression. However, due to the irregular shape or dense arrangement of passion fruit, RPN may not accurately generate all candidate boxes, leading to lower intersection over union

(IOU) in complex scenes and even possible instances of missed or false detections. In backlit and direct light environment tests, Faster-RCNN was prone to misclassifying unripe passion fruit as ripe when light was insufficient and may suffer from low IOU when the light is too intense. In contrast, FSOne-YOLOv7 is less sensitive to insufficient light conditions, yielding higher IOU between predicted boxes and ground truth annotations. As for SSD, when using the feature pyramid, different scales of feature maps are obtained through upsampling or interpolation, leading to information loss or blurring. SSD utilizes a fixed-scale feature extractor, making it challenging to effectively capture information from targets of different scales and levels in complex environments. Hence, in backlit and direct light environment tests, SSD performed poorly in detecting obscured or smaller passion fruit. In contrast, FSOne-YOLOv7 utilizes a feature pyramid network (FPN) to handle targets at different scales and optimizes the model's focus area by employing C2Faster, ODConv, and SimSPPF in the slim-neck, better integrating contextual information and demonstrating improved performance when detecting obscured or smaller passion fruits.

In the overall evaluation, when compared to the different detection methods, FSOne-YOLOv7 demonstrated a reduced occurrence of missed and false positive detections for passion fruit. It showcased superior performance and proved more suitable for addressing the challenges posed by the complex growth environment and diverse appearance characteristics of passion fruit.

5. Discussion

Passion fruit, as a vine fruit, possesses distinct characteristics that set it apart from other fruits such as apples, cherry tomatoes, citrus fruits, and strawberries. Its natural environment includes dense foliage, which often obscures the fruit and poses a challenge for detection. Moreover, the visual appearance of passion fruit undergoes changes throughout its ripening process. Therefore, the complex growth environment and diverse visual characteristics of passion fruit present obstacles to real-time detection. While there have been notable research advancements in the real-time detection of passion fruit in complex environments, limited attention has been given to the real-time detection of passion fruit at different stages of ripeness. This study addressed this gap by categorizing passion fruit into two classes: immature and mature, for detection purposes. A comparative analysis of the accuracy data obtained from the experimental results revealed that the model exhibited superior performance in detecting mature passion fruit compared to immature ones.

Another challenge arises from the influence of light intensity on passion fruit detection. The implementation of a two-layer vertical trellis system for high-density planting of purple passion fruit has helped to partially address the problem of inadequate illumination. However, ensuring the consistent and accurate detection of passion fruit under different light intensities remains crucial. Future research should focus on optimizing image processing techniques to further diminish the impact of light variations, thereby enhancing detection accuracy. These techniques have the potential to bolster the robustness of the detection algorithm, enabling finer classification and differentiation of passion fruit based on their maturity stages. Ultimately, this will lead to improved accuracy and stability in fruit detection.

Integrating the FSOne-YOLOv7 model into agricultural equipment, such as harvesting robots, holds great potential for advancing agricultural automation and intelligence. However, this integration may face practical challenges. Controlled-environment agriculture, like greenhouse farming, differs significantly from traditional open-field methods, with varying light intensity and spectra that could affect the model's accurate recognition and classification of passion fruit. Additionally, complex obstructions, such as vegetation and support structures, demand higher detection capabilities from the model, making it crucial to adapt and optimize the FSOne-YOLOv7 model for these unique environments. Factors like equipment motion, vibration, and noise may also impact the model's detection accuracy and stability, necessitating technical adjustments and optimizations.

Addressing data transmission and real-time processing challenges is essential to ensure the model's timely response and accurate detection results. As a result, integrating the FSOne-YOLOv7 model into agricultural equipment requires comprehensive consideration, extensive testing, and optimization to develop feasible application solutions.

In larger-scale agricultural environments and scenarios involving the simultaneous detection of multiple fruits, the scalability and adaptability of the FSOne-YOLOv7 model become increasingly vital. As farm sizes expand and the number of fruit trees and complex orchard areas increase, the model's computational and memory resource requirements will escalate. To address these challenges, distributed computing or model distillation techniques can be employed to alleviate the model's burden, ensuring it remains efficient and accurate in handling large-scale data. Furthermore, exploring multi-model fusion methods combining different target detection models can cater to the diverse detection needs of various crops and complex scenarios. These optimization measures will enhance the scalability and application potential of the FSOne-YOLOv7 model in larger-scale agricultural environments, providing robust support for the intelligent development of the agricultural sector.

In conclusion, the FSOne-YOLOv7 model is of great significance in agriculture. It excels in detecting ripe passion fruit across different maturity stages and can be further optimized for varying light conditions. Its integration into agricultural equipment may face challenges but promises automation and intelligence in farming. For larger-scale agriculture and multiple fruit detection, optimization and fusion methods enhance its scalability and application potential, providing a powerful solution for efficient and accurate fruit detection. With continuous research and development, the model's capabilities can be further expanded, contributing to the continued growth and advancement of precision agriculture.

6. Conclusions

In conclusion, this study addresses the challenges associated with the complex growth environment and diverse visual characteristics of passion fruit through the introduction of FSOne-YOLOv7, a real-time detection model. This model achieves a favorable trade-off between detection accuracy and inference speeds, with several notable contributions, as summarized below:

(1) The design of a novel backbone network named ShuffleOne and an efficient slim-neck module based on YOLOv7. These enhancements facilitate model miniaturization and acceleration, enabling the improved detection of the diverse visual characteristics of passion fruit. The efficacy of the model is validated through feature analysis, ablation experiments, and comparative studies.

(2) The utilization of the Grad-CAM heat map visualization technique to analyze the feature information of ShuffleOne and the slim-neck module in the FSOne-YOLOv7 model. The experimental results demonstrate that ShuffleOne effectively filters out irrelevant background regions, while the slim-neck module integrates contextually relevant feature information, thereby reducing computational resources and inference time requirements.

(3) The FSOne-YOLOv7 model achieves significant results in passion fruit detection. It attains a mAP of 90.45%, detects frames at a rate of 58.2 fps, exhibits an average inference time of 6.7 ms per frame, contains 13.6 million parameters, and has a size of 29.2 MB. In comparison to the original YOLOv7 model, the proposed model improves mAP by 4.35 percentage points, increases inference speed by 25.1%, reduces the parameter count by 62.7%, and has a volume approximately 2/5 that of YOLOv7. Moreover, the detection accuracy for immature and mature passion fruit is enhanced by 6.6 and 3 percentage points, respectively. When compared to other object detection methods, the FSOne-YOLOv7 model demonstrates advantages in terms of smaller model parameters and volume, as well as superior mAP and detection speeds. Specifically, it outperforms Faster-RCNN and

SSD models by improving mAP by 10 and 4.4 percentage points, respectively, and increasing detection speeds by approximately 2.6 and 1.5 times.

(4) The experimental results validate the advantages of the proposed FSOne-YOLOv7 model in terms of detection accuracy and inference speeds, rendering it suitable for harvest operations. Its low computational complexity and compact model size make it particularly applicable in scenarios with limited memory and computing capacity, where accuracy is of the utmost importance. Therefore, the model can serve as a valuable technical reference for future applications in mobile or embedded devices, offering an effective target detection method for passion fruit production in the field of agriculture, while also providing valuable research insights for similar fruit detection tasks.

Author Contributions: Conceptualization, R.Z. and X.L.; Formal analysis, J.O.; Investigation J.O. and R.Z.; Methodology, J.O. and X.L.; Software, J.O.; Validation, G.L.; Writing—original draft, J.O.; Supervision, R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (grant no. 32101632), Featured Innovation Projects of Guangdong Province of China (grant 2021KTSCX049), and 2023 Guangdong Provincial Science and Technology Innovation Strategy Special Project (grant pdjh2023b0260).

Data Availability Statement: Our dataset is available at https://drive.google.com/drive/folders/1xR9o4qoUmehiEzPSEILyGI1nzXxFTmIu?usp=drive_link, accessed date: 18 July 2023.

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

References

1. Fonseca, A.M.A.; Geraldi, M.V.; Junior, M.R.M.; Silvestre, A.J.; Rocha, S.M. Purple passion fruit (*Passiflora edulis* f. edulis): A comprehensive review on the nutritional value, phytochemical profile and associated health effects. *Food Res. Int.* **2022**, *160*, 111665.
2. Faleiro, F.G.; Junqueira, N.T.V.; Junghans, T.G.; Jesus, O.; Miranda, D.; Otoni, W.C. Advances in passion fruit (*Passiflora* spp.) propagation. *Rev. Bras. Frutic.* **2019**, *41*, e155.
3. Zhao, L.; Wu, L.; Li, L.; Zhu, J.; Chen, X.; Zhang, S.; Li, L.; Yan, J.-K. Physicochemical, structural, and rheological characteristics of pectic polysaccharides from fresh passion fruit (*Passiflora edulis* f. *flavicarpa* L.) peel. *Food Hydrocoll.* **2023**, *136*, 108301.
4. Shi, M.; Ali, M.M.; He, Y.; Ma, S.; Rizwan, H.M.; Yang, Q.; Li, B.; Lin, Z.; Chen, F. Flavonoids accumulation in fruit peel and expression profiling of related genes in purple (*Passiflora edulis* f. *edulis*) and yellow (*Passiflora edulis* f. *flavicarpa*) passion fruits. *Plants* **2021**, *10*, 2240.
5. Zhou, H.; Wang, X.; Au, W.; Kang, H.; Chen, C. Intelligent robots for fruit harvesting: Recent developments and future challenges. *Precis. Agric* **2022**, *23*, 1856–1907.
6. Nkalubo, L.B.; Nakibuule, R. A Review on Real-Time Object Detection Models Using Deep Neural Networks. *EasyChair* **2022**, *preprint*.
7. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
9. 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. *arXiv* **2022**, arXiv:2207.02696.
10. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer International Publishing: Cham, Germany, 2016; pp. 21–37.
12. Zheng, Y.-Y.; Kong, J.-L.; Jin, X.-B.; Su, T.-L.; Nie, M.-J.; Bai, Y.-T. Real-time vegetables recognition system based on deep learning network for agricultural robots. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 2223–2228.
13. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516.
14. Quan, L.; Li, H.; Li, H.; Jiang, W.; Lou, Z.; Chen, L. Two-Stream Dense Feature Fusion Network Based on RGB-D Data for the Real-Time Prediction of Weed Aboveground Fresh Weight in a Field Environment. *Remote Sens.* **2021**, *13*, 2288. https://doi.org/10.3390/rs13122288.
15. Lu, S.; Liu, X.; He, Z.; Zhang, X.; Liu, W.; Karkee, M. Swin-Transformer-YOLOv5 for Real-Time Wine Grape Bunch Detection. *Remote Sens.* **2022**, *14*, 5853.

16. Ridho, M.; Irwan, F. Strawberry Fruit Quality Assessment for Harvesting Robot using SSD Convolutional Neural Network. In Proceedings of the 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Semarang, Indonesia, 20–21 October 2021; pp. 157–162.
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Pan, Y.; Zhu, N.; Ding, L.; Li, X.; Goh, H.-H.; Han, C.; Zhang, M. Identification and Counting of Sugarcane Seedlings in the Field Using Improved Faster R-CNN. *Remote Sens.* **2022**, *14*, 5846. https://doi.org/10.3390/rs14225846.
20. Zhong, S.; Xu, W.; Zhang, T.; Chen, H. Identification and depth localization of clustered pod pepper based on improved Faster R-CNN. *IEEE Access* **2022**, *10*, 93615–93625.
21. Kumar, D.; Kukreja, V. Image-based wheat mosaic virus detection with Mask-RCNN model. In Proceedings of the 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 23–25 March 2022; pp. 178–182.
22. Zhang, C.; Kang, F.; Wang, Y. An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds. *Remote Sens.* **2022**, *14*, 4150. https://doi.org/10.3390/rs14174150.
23. Shang, Y.; Xu, X.; Jiao, Y.; Wang, Z.; Hua, Z.; Song, H. Using lightweight deep learning algorithm for real-time detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2023**, *207*, 107765. https://doi.org/10.1016/j.compag.2023.107765.
24. 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.
25. Luo, Z.; Li, P.; Song, F.; Sun, Q.; Ding, H. Lightweight Passion Fruit Detection Model Based on Embedded Device. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 262–269+322.
26. Wu, X.; Tang, R. Fast Detection of Passion Fruit with Multi-class Based on YOLOv3. In *Proceedings of 2020 Chinese Intelligent Systems Conference*; Springer: Singapore, 2021; Volume II, pp. 818–825.
27. Tu, S.; Pang, J.; Liu, H.; Zhuang, N.; Chen, Y.; Zheng, C.; Wan, H.; Xue, Y. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. *Precis. Agric.* **2020**, *21*, 1072–1091.
28. Tu, S.; Huang, J.; Lin, Y.; Li, J.; Liu, H.; Chen, Z. Automatic detection of passion fruit based on improved faster R-CNN. *Res. Explor. Lab.* **2021**, *40*, 32–37.
29. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
30. 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.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
32. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
33. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
34. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
36. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
37. Han, D.; Kim, J.; Kim, J. Deep pyramidal residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5927–5935.
38. Ding, X.; Zhang, X.; Han, J.; Ding, G. Diverse branch block: Building a convolution as an inception-like unit. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10886–10895.
39. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
40. Hu, M.; Feng, J.; Hua, J.; Lai, B.; Huang, J.; Gong, X.; Hua, X. Online convolutional reparameterization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 568–577.
41. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. An improved one millisecond mobile backbone. *arXiv* **2022**, arXiv:2206.04040.
42. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.

43. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. *arXiv* **2023**, arXiv:2303.03667.
44. Yang, B.; Bender, G.; Le, Q.V.; Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. In Proceedings of the Advances in Neural Information Processing Systems: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
45. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic convolution: Attention over convolution kernels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11030–11039.
46. Li, C.; Zhou, A.; Yao, A. Omni-dimensional dynamic convolution. *arXiv* **2022**, arXiv:2209.07947.
47. Glenn, J. YOLOv5 Release v6.1. 2022. Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.1 (accessed on 15 December 2022).
48. Meituan.YOLOV6 Release v4.0. 2023. Available online: https://github.com/meituan/YOLOv6 (accessed on 18 January 2023).
49. WongKinYiu. YOLOv7. 2023. Available online: https://github.com/WongKinYiu/yolov7 (accessed on 18 January 2023).
50. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.