A Lightweight and High-Precision Passion Fruit YOLO Detection Model for Deployment in Embedded Devices

Qiyan Sun *, Pengbo Li *, Chentao He , QiMing Song , Zhicong Luo

Posted Date: 29 May 2024

doi: 10.20944/preprints202405.1950.v1

Keywords: passion fruit detection; lightweight; deep learning; knowledge distillation; embedded devices

Article

A Lightweight and High-Precision Passion Fruit YOLO Detection Model for Deployment in Embedded Devices

Qiyan Sun [1,†], Pengbo Li [2,†], Chentao He [1], Qiming Song [1], and Zhicong Luo [1,3,*]

[1] College of Mechanical and Electrical Engineering, Fujian Agriculture and Forestry University, Fuzhou 310002, China
[2] College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 310002, China
[3] Fujian Key Laboratory of Agricultural Information Sensoring Technology, Fuzhou 310002, China
* Correspondence: zcl@fafu.edu.cn
† These authors contributed equally to this work.

Abstract: In order to shorten the detection time and improve the average precision on embedded devices, A lightweight and high accuracy model is proposed for passion fruit in complex environments (backlight, occlusion, overlap, sunny, cloudy, rainy). Firstly, replacing the backbone network of YOLOv5 with a lightweight GhostNet model reduces the number of parameters and computation while improving detection speed. Secondly, a new feature branch is added to the GhostNet network, and the feature fusion layer in the neck network is reconstructed to effectively combine the lower-level and higher-level features, which not only improves the accuracy of the model but also maintains its lightweight. Finally, the knowledge distillation methods are used to transfer the knowledge from the more capable teacher model to the less capable student model, which significantly improving the detection accuracy. The improved model is denoted as G-YOLO-NK. The average accuracy of the G-YOLO-NK network is 96.00%, which is 1.00% higher than the original YOLOv5s model. Furthermore, the improved model size is 7.14MB, reduced to half of the original model, and the real-time detection frame rate is 11.25 FPS on the Jetson Nano. Compared to the state-of-the-art model, the proposed model outperforms them in terms of average precision and detection performance. The present work provides an effective model for real-time detection of passion fruits in complex orchard scenes, which can provide valuable technical support for the development of orchard picking robots and greatly improve the intelligence level of orchards.

Keywords: passion fruit detection; lightweight; deep learning; knowledge distillation; embedded devices

1. Introduction

Passion fruit and its by-products are highly nutritious and have significant commercial value that can be exploited [1] Passion fruit cultivation is mainly distributed in regions such as Guangdong, Yunnan, Fujian, and others in China. The planting area is expanding and the number of varieties is increasing. Currently, passion fruit picking is still mainly done by hand, which undoubtedly consumes a great deal of labor. The development of agricultural robotic picking is of great significance in terms of liberating labor and leading the fruit industry towards a precision model [2] In recent years, using image technology to detect fruits has garnered research interest and emerged as a prominent topic, which determines the accuracy and integrity of agricultural robotic picking efforts.

Traditional machine learning approaches are based primarily on manually designed combinations of features and classifiers [3] For example, the basic texture, color, shape features of fruits are studied. Tu et al. [4] established an RGB color space model to detect the maturity of passion fruit. Li et al. [5]used a region classifier to classify ripe and unripe tomatoes and used the Hough transform circle detection method to achieve detection of unripe tomatoes, which takes a lot of time and does not have

 doi:10.20944/preprints202405.1950.v1

high detection accuracy. Yang et al. [6] attempted to use various machine learning methods to classify apricots based on shape features. The above image recognition methods suffer from poor robustness and difficulty in handling a lot of data. Object detection technology mainly involves identifying and classifying the positions to be detected in images or videos. There are several algorithms for target detection, they can be generally classified as Faster-RCNN [7] algorithm based on two-stage detection, and SSD [8] and YOLO [9] algorithms based on one-stage detection. The one-stage detection algorithm has a higher detection speed, which is beneficial for mobile deployment.

The YOLO algorithm with simple structure and short inference time is one of the best choices for detection models [10] Lawal et al. [11] combined DenseNet with the YOLOv3 network and used the Mish activation function to detect tomato. Roy et al. [12] added DenseNet, SPP blocks, and improved PANet to YOLOv4 network to enhance network detection capability. Lin et.al. [13] improved the YOLOv4 network by incorporating attention mechanisms. The goal was to eliminate noise and enhance the feature extraction of small targets. Using the point-line distance loss function [14] and optimizing the upsampling algorithm [15] to improve the YOLOv5 model, thereby enhancing reliability and accuracy of model. Although the above study improves the detection accuracy of the model, it increases the parameters and the detection time of the model. Researchers have conducted various studies in order to compress and accelerate the model from various aspects [16] For example, lightweight network design, pruning, and knowledge distillation. The lightweight network design method is used to design small models and quickly recognize networks by adjusting the internal structure of the network, such as MobileNet [17], GhostNet [18], shuffleNet [19] etc. The purpose of pruning and lightweight network design is the same, which involves removing redundant parameters from the network, such as channel pruning [20], kernel pruning [21,22], and weight pruning [23]. The knowledge distillation method proposes to transfer information from one model to another, which efficiently extracts features and substantially improves detection accuracy [24,25].

Deploying deep learning models on mobile devices for agricultural detection is more meaningful for practical applications [26]. Researchers have effectively improved different models to improve detection accuracy and recognition speed. Xu et al. [27] introduced GhostNet to replace the YOLOv4 backbone network and an effective channel attention mechanism in the neck to detect fruit. The improved model size is 43.5MB, and the detection time for a single image is 48.2 ms. Jiang et al. [28] proposed Generalized-FPN (GFPN) as cross-scale connection style, integrating the features of the previous layer and current layer. Subsequently, Xu et al. [29] improved GFPN and applied it to the YOLO network, increasing accuracy by 1.4%. Guo et al. [30] combined knowledge distillation strategy in the YOLOv5s model and achieved an accuracy of 94.67% on a self-made dataset, which is 4.83% higher than the original model. Yang et al. [31] constructed a lightweight model method based on backbone replacement, sparse training and knowledge distillation techniques, which method reduces parameters and volume, but AP also decreases by 2.7%. Although the above methods have made some progress in model lightweighting or accuracy, compared to the original model, they have not achieved a balance between accuracy and lightweighting. Therefore, it is important to study a detection algorithm with high generalization ability on embedded devices.

In this paper, we constructed a passion fruit dataset in a complex environment and addressed the issuses of parameter redundancy and poor real-time performance of the model in embedded devices. This paper presents G-YOLO-NK model, which is a lightweight and high-precision model based on an improved YOLOv5. The first contribution of this study is that we used a lightweight GhostNet to replace the YOLOv5s backbone for reduce the number of parameters and computation of the network and compared it with other methods that use a lightweight network as the backbone network. Secondly, we reconstruct the neck of the network by combining the new branches of the feature extraction layer with the feature fusion layer. Finally, we used the knowledge distillation method to enable student models to learn useful dark knowledge from teacher models, verifying the effectiveness of the distillation method in a one-stage detector. The experimental results show that the

 doi:10.20944/preprints202405.1950.v1

improved algorithm reduces the number of model parameters while improving the detection speed, and has better real-time detection performance in complex environments on embedded devices.

The rest of the article is organized as follows. In Section 2, materials and methods related to preprocessing image datasets and detection algorithms are presented; in Section 3, training methods and evaluation metrics are described; Section 4 gives the results of the study and comparison experiments; and Section 5 contains the conclusions and outlook.

2. Materials and Methods

2.1. Image Acquisition and Preprocessing

In order to enhance the single passion fruit dataset, data collection was conducted at the Junzhiyu Passion Fruit Base in Minhou County, Fuzhou City, Fujian Province. The image acquisition device was a Nikon digital camera, and the distance of the camera from the passion fruit was 80-100 cm during the acquisition, and the image size was 1920×1080 pixels and saved as JPEG format. The weather at the time of data collection included sunny, rain and cloudy. Images of passion fruit were captured under different lighting conditions and compositions to enhance diversity. This included down light, back light, leaf shading, and fruit overlap scenarios. There were 3269 images in total, including 837 unshaded fruits, 1124 shaded by leaves, and 1308 overlapping fruit, of which 1732 were with light and 1537 were with back light. The images were collected as shown in Figure 1.

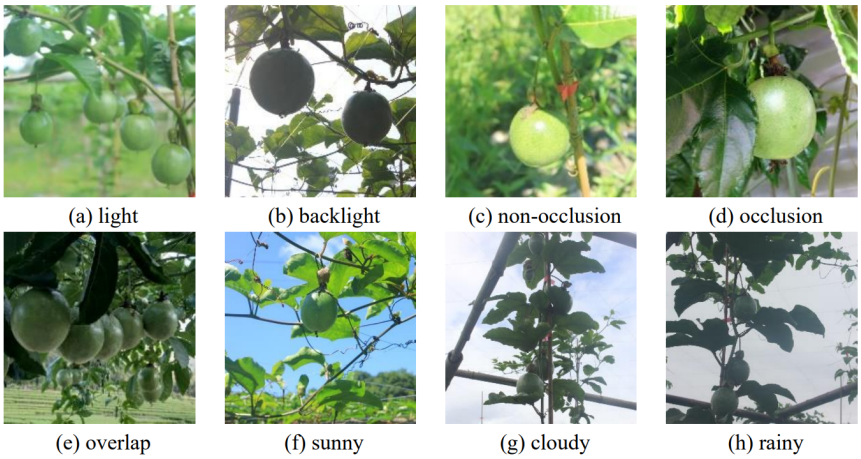


Figure 1. Pictures of passion fruit in a complex environment.

Enhancing images can emphasize the overall or local features of passion fruit images, enhance the differences between different object features, and suppress the extraction of irrelevant features by deep learning networks [32]. Expanding the image training set is advantageous in improving the learning capacity of deep neural networks and reducing overfitting caused by insufficient sample diversity [33]. This approach has the potential to greatly enhance the robustness and generalization capabilities of the trained model. Therefore, this section expands the dataset by image enhancement such as rotating the original data, adding Gaussian noise and contrast adjustment. Some of the enhanced images as shown in Figure 2. By rotating the original image by 90 degrees, adding Gaussian noise and adjusting the contrast is used to increase the recognition difficulty of the model, as shown in Figure 2 (b), (c) and (d). After the aforementioned offline data augmentation, 5140 images of passion fruit were finally obtained. The above images were manually marked and bounding boxes were drawn using the software LabelImg, which eventually generated xml format files. The completed dataset was randomly divided into 8:2 raining and test sets[34], resulting in 4,112 images allocated for training and 1,028 images for testing.

 doi:10.20944/preprints202405.1950.v1

(a) original (b) rotate (c) noise (d) contrast

Figure 2. Data augmentation.

2.2. YOLOv5 Algrithm

The YOLOv5 target detection algorithm was released by Ultralytics in 2020 and has high accuracy and fast inference, making it one of the best performing target detection models available today [35]. The YOLOv5 model can be separated into four parts: Input, Backbone, Neck, and Detect. The input side uses Mosaic data enhancement to randomly scale, cut, and stitch the passion fruit images into the network, which not only enriches the data set but also enhances the robustness of the network model. The backbone network adds Focus, C3 and SPP structure to the YOLOv3 network. The main role of the backbone network is to extract the features of the image and enhance the learning ability of the convolutional neural network. The path aggregation network (PANet) structure is applied in the neck network to effectively extract comprehensive location information from top to bottom, while simultaneously capturing semantic features from bottom to top. This integration enhances the localization of targets by leveraging both spatial and semantic information. The detection network produces the final output by combining the probability class of the target, the confidence score, and the location information of the target box. The structure of the YOLOv5 algorithm is shown in Figure 3.

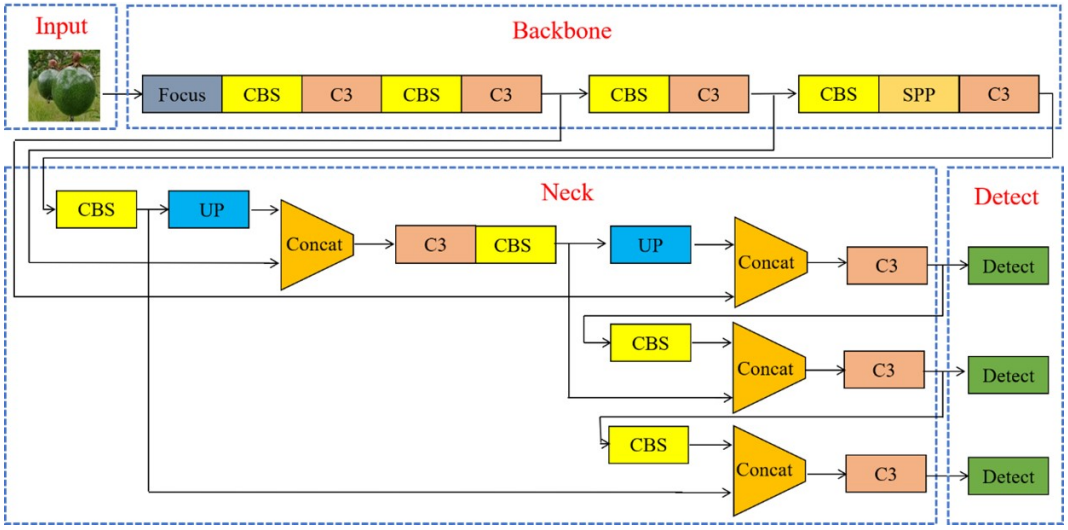


Figure 3. The architecture of the Yolov5 algorithm.

Researchers have developed four different YOLOv5 models based on the varying depth and width of the network, demonstrating the exceptional flexibility of this algorithm. This indicates that the algorithm is highly adaptable and can be customized to suit different requirements. In this study, the detection performance of the four models was tested using a homemade passion fruit dataset, and Table 1 shows the test results. In order to save the memory of embedded devices, the YOLOv5s model is chosen as the baseline in this paper. The YOLOv5s overall loss encompasses the classification, localization, and confidence losses. The cross-entropy loss function is employed for the classification and confidence losses, simplifying computation complexity. The localization loss uses CIoU Loss, which helps ensure that the model can accurately locate the target.

 doi:10.20944/preprints202405.1950.v1

Table 1. The performance comparison of different model of YOLOv5.

Model	P(%)	R(%)	AP(%)	Szie(MB)
YOLOv5s	94.90	90.60	95.40	14.40
YOLOv5m	94.80	90.90	95.70	40.10
YOLOv5l	94.80	91.20	96.00	88.40
YOLOv5x	94.90	92.20	96.10	164.00

2.3. Improvement of the yolov5s Model

By aiming to reduce the model size and computation, while improving its detection accuracy, this paper proposes improvements to the YOLOv5s model. The structure of the improved YOLOv5s algorithm is shown in Figure 4.

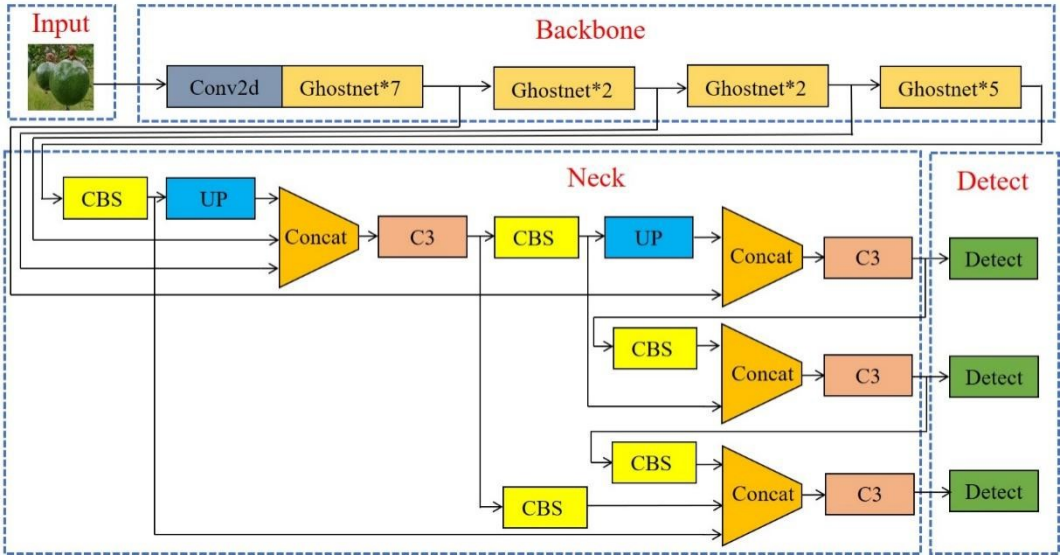


Figure 4. The structure of the G-YOLO-N algorithm.

2.3.1. Lightweight Improvements

Due to the limited storage space and computing resources of embedded devices, deploying deep learning models can be quite challenging [36], thus requiring further model compression [37]. GhostNet outperforms MobileNet and ShuffleNet in compu- ational performance in a compact network design[38]. A model with outstanding performance has sufficient complexity in the feature layer to understanding of the input information, which is an important factor in the success of a model [39]. In lightweight network design, it is not feasible to simply remove useful redundant features. Therefore, GhostNet was specifically designed to enable fast inference on mobile devices while maintaining important features. The Ghost module in GhostNet is the key structure for generating feature layers, which facilitates the extraction of effective feature layers. The Ghost module shown in Figure 5, The Ghost module uses a series of inexpensive linear operations to generate new feature layers, which may be 1 × 1 convolutions or 3 × 3 convolutions.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2024 doi:10.20944/preprints202405.1950.v1

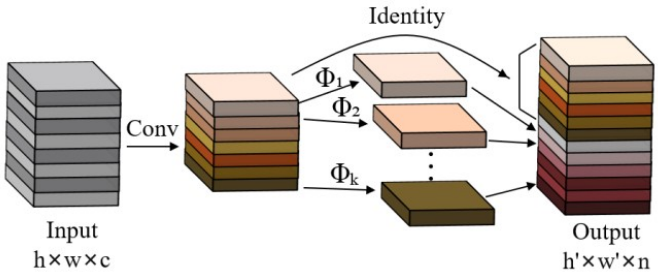


Figure 5. The structure diagram of Ghost module.

Suppose the input channel is denoted by c, the height and width of the feature map by h and w respectively, the height and width of the output feature map by h' and w', the number of convolution kernels by n, the size of the convolution kernel by k, the size of the linear transform convolution kernel by d, and the number of transforms by s. The parameter compression using Ghoset convolution instead of conventional convolution is shown in equations (1). The acceleration ratio is derived as shown in equations (2).

$$r_c = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{1}$$

$$r_c = \frac{n \times h' \times w' \times c \times k \times k}{\frac{n}{s} \times h' \times w' \times c \times k \times k + (s-1) \times h' \times w' \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s \tag{2}$$

It can be observed from the equation that the benefits of computational acceleration and parameter compression are influenced by the number of transformations. In the Ghost module, the total number of parameters and the computational complexity are reduced compared to a normal convolutional neural network, without changing the output feature layer size.

Ghost bottleneck was designed by combining the advantages of Ghost module and Restnet residual connection. As shown in Figure 6., when the step size is 1. The first Ghost module functions as an expansion layer that increases the number of channels, while the second Ghost module concentrates on reducing the number of channels in the resultant feature layer to align with the input feature layer. For a step size of 2, a deep convolution of step size 2 is introduced in between the two Ghost modules to construct the Ghost bottleneck structure.

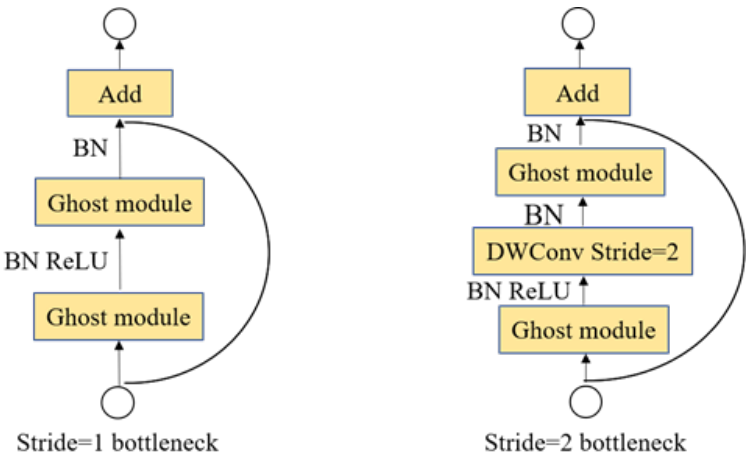


Figure 6. The structure diagram of Ghost bottleneck model.

2.3.2. Reconstructing the Neck Network

In deep learning networks, the robustness and generalization ability of an improved model depends on modifying the backbone network, but modifying the neck network can also have this effect. In order to identify objects at different scales, Adelson et al. [40] first proposed image pyramid to build

 doi:10.20944/preprints202405.1950.v1

a feature pyramid, which has been applied to image analysis, data compression, and image processing. However, this approach calculates features on each image scale slowly and inaccurately. To address this problem, A top-down connected Feature Pyramid Network (FPN) [41] and Path Aggregation Network (PANet) [42] were proposed by researchers for boosting information flow. Jiang et al. [43] proposed the Generalized-FPN for efficient object detection, which improves FPN with a novel queen-fusion. In order to achieve the goal of multi-scale information exchange, In this paper, we propose an adaptive feature pyramid network (AFPN) based on the Generalized-FPN idea to effectively detect passion fruit targets.

Due to the close shooting distance, most of the passion fruit in the complex environment are large targets. At the same time, the backbone network is replaced by a lightweight network, the information in the feature extraction layer will be reduced, and the information passed to the neck feature fusion stage will also be lost by a part, and the detection performance of the model will be further reduced. Adding an input feature layer at backbone network and combining the large target output features of the neck network to improve the detection performance of the model for medium and large targets, making up for some of the information lost in the lightweight network. With the modification of the above scheme, the neck network has strong semantic features at high level and localization features at low level. In the meantime, the network improves the sensitivity and detection capability of large targets. The neck network before and after the improvement is shown in Figure 7.

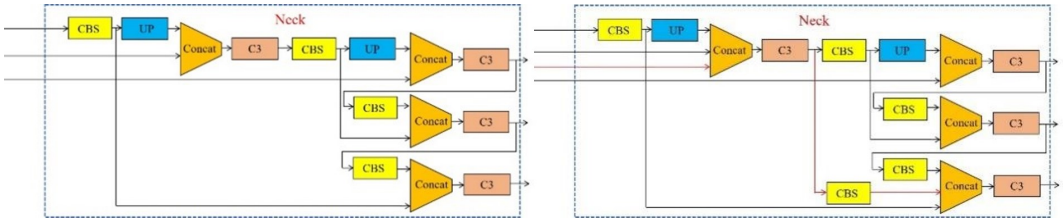


Figure 7. Improved neck network before and after.

2.3.3. Knowledge Distillation Enhancement

Knowledge distillation (KD) is an effective method to further improve the accuracy of model detection [44]. Distillation is not yet widely used in the YOLO series of network improvements, especially for small models with a single target. We did a special study for G-YOLOv5-N and finally used the distillation technique to achieve the effect improvement on the G-YOLOv5-N model. Firstly, the teacher network model was chosen rationally. We choose the YOLOv5 series of models in order to ensure that the student and teacher models have the same scale in the output layer. Next, the YOLOv5x model with hight accuracy was selected as the teacher model based on the results of training on the passion fruit dataset.

In general, the implementation of distillation has to train the teacher network first after parameter initialization, and then use the teacher network with rich knowledge learned to train the student network. The flow chart of this paper using this algorithm is shown in Figure 8.

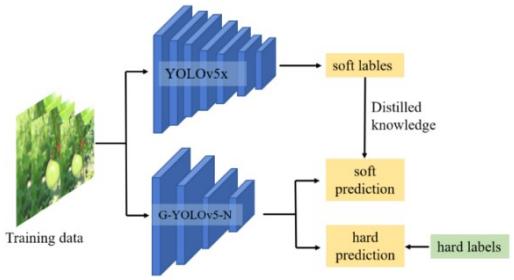


Figure 8. The flow chart of the knowledge distillation algorithm.

 doi:10.20944/preprints202405.1950.v1

Teacher networks can be used for predictive learning in student networks. To enhance the information exchange between them, the predictions of the teacher's network are used as soft labels. The teacher network then trains the student network using these soft labels, allowing the student network to learn from the teacher's knowledge. Additionally, the student network helps to prevent the teacher network from making mistakes by learning from hard labels, by incorporating soft labels, the student network can acquire more nuanced and hidden knowledge. This hidden knowledge is usually expressed as a categorical output y, as shown in equations (3).

$$y_i{}' = \frac{\exp y_i}{\sum \exp y_i} \tag{3}$$

It can be analyzed from Eq 3. that the model does not facilitate to learning the dark knowledge in the passion fruit image. Then the warming process is needed. As in equations (4).

$$y_i{}' = \frac{\exp(y_i/T)}{\sum \exp(y_i/T)} \tag{4}$$

The cumulative loss function utilized in the knowledge distillation algorithm introduced comprises the original network model's loss and the distillation loss. The distillation loss is composed of the classification loss, bounding box loss, and localization loss. To highlight the model learning passion fruit target, the background region is weakened, thus introducing a weighting factor K. Distillation loss is shown in equations (5), and the total loss equation is shown in equations (6).

$$L_{dloss} = K(L_{obj} + L_{cl} + L_{bb}) \tag{5}$$

$$L_{loss} = \alpha L_{dloss} + (1 - \alpha) L_{yolo} \tag{6}$$

3. Model Training and Evaluation

3.1. Experimental Environmet

To comprehensively evaluate the effectiveness of the enhanced algorithms proposed in the paper under different experimental scenarios, two distinct platforms were utilized. The first platform involved a PC development environment, while the second platform focused on an embedded development environment. This approach allowed for a comprehensive assessment of the proposed algorithms' performance across diverse computing environments. Windows 10 x64 operating system was selected for the PC development platform with Intel® Core™ i7-10700F CPU 2.90 GHz, NVIDIA GeForce RTX 3070 8 G GPU, and 32.0 GB RAM running memory.

The embedded experimental platforms used the Nvidia Jetson Nano device for model inference and testing. The experimental environment was Ubuntu 18.04 with Jetpack 4.5, CUDA 10.2, and cuDNN 8.0. The programming language used was Python 3.6, and the deep learning framework was Pytorch 1.8.1 and Torchvision 0.9.1. The real-time detection in Jetson nano is shown in Figure 9.

 doi:10.20944/preprints202405.1950.v1

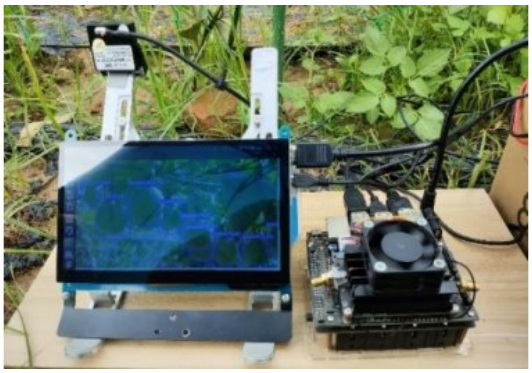


Figure 9. The real-time detection at Jetson nano.

In the PC platform, training specific parameters are as follows: The input image is 640 × 640 pixels, the batch size is 8, the initial learning rate is set to 0.001, and the optimizer is set to Adam. The number of training iterations is set to 70 to obtain better model, and the loss value change curve and AP value change curve after 70 training sessions are applied to the test set, as shown in Figure 10. During the first 15 cycles of network training, the loss value of the network decreases rapidly and the AP value increases rapidly, and then enters a stable convergence phase. After 60 epochs, the loss value decreases gently, the AP value increases gently, and the loss function curve and AP value curve converge, indicating that the model training effect is successful.

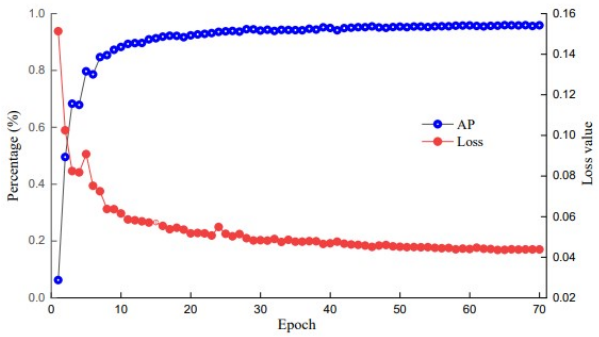


Figure 10. AP and Loss curves.

3.2. Evaluation Metrics

In order to assess the performance of the improved YOLOv5s based on its detection results, several evaluation metrics are employed by the investigators including precision (P), recall (R), average precision (AP), parameters amount, floating point operations per second (FLOPs), model size, and frames per second (FPS). Taking passion fruit samples as an example, precision refers to the proportion of correctly predicted passion fruit samples to all predicted passion fruit samples by the model classifier. Recall represents the proportion of correctly predicted passion fruit samples to the actual positive passion fruit samples. These are shown in equations (7) and (8), respectively. However, Precision and Recall do not allow for a direct assessment of detection accuracy. The performance of the detection network is assessed by introducing average precision (AP), which represents the average accuracy in detection. As shown in equations (9), where TP is actual passion fruit and predicted not to be passion fruit, FP is not actual to be passion fruit and predicted to be passion fruit, and FN is actual passion fruit and predicted not to be passion fruit. The number of floating-point operations per second reflects the time complexity of the model, measuring the computation involved in operations such as convolution and pooling. The number of parameters, on the other hand, describes the size of the model and its

 doi:10.20944/preprints202405.1950.v1

spatial complexity in the algorithm. Lastly, the frames per second is used to measure the real-time performance of the model on the hardware platform.

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{9}$$

4. Experiment Results and Analysis

4.1. Impact of Different Backbone Networks on the Algorithm

In order to select a network with better lightweight performance as the backbone network for the YOLOv5s model. The first experiment compares the results of the performance impact of lightweight network layers and configurations on the backbone network, selecting three types of backbone networks, namely MobileNetv3, ShuffluNetv3, and GhostNet. To avoid reasonable bias, the FPN network and the detection head are kept constant. The experiment was conducted as a state-of-the-art comparison on the embedded platform Jetson Nano, and the results are presented in Table 2.

Table 2. Comparison of different backbone networks.

Model	P(%)	R(%)	AP(%)	GFLOPs	Param(M)	Size(MB)
YOLOv5s	94.90	90.60	95.40	15.80	7.10	14.40
M-YOLOv5	92.50	86.60	92.50	6.30	3.54	7.08
S-YOLOv5	92.50	87.30	92.90	7.40	3.55	7.12
G-YOLOv5	92.90	87.40	93.10	6.50	3.20	7.10

Compared to the original YOLOv5s model, the improved model (G-YOLOv5) has an average precision (AP) reduction of 2.30%, The reason for the above is that the reduced number of model parameters and convolutional layers of the G-YOLOv5 model leads to a reduction in the network's ability to extract features. Compared to the M-YOLOv5 and S-YOLOv5 models with AP improvements of 0.6% and 0.2%. Meanwhile, the network model volume is 7.10MB, reducing the original network by 50.69%. The FLOPs and the params of the improved model are significantly reduced, compared to the YOLOv5s model, the FLOPs has been reduced by 58.86% and the params by 54.93%. Interestingly, there is a discrepancy between the results of the proposed three networks after replacing the backbone and the results of the original network, which indicates that the effectiveness of the network is influenced by the total number of its parameters and the particular network structure. Ultimately, the lightweighting of the model was achieved by replacing the backbone network.

4.2. Ablation Experiments

The ablation experiment focuses on analysing the value of the existence of each improved method. An ablation experiments were conducted on the self-made dataset constructed in this study. The improved model was tested and the experimental results are shown in Table 3.

 doi:10.20944/preprints202405.1950.v1

Table 3. Results of the ablation experiment.

Baseline	Light	Neck	KD	P(%)	R(%)	AP(%)	Size(MB)
YOLOv5s				94.90	90.60	95.40	14.40
	✓			92.90	87.40	93.10	7.10
		✓		95.10	90.50	95.50	14.46
			✓	93.70	92.90	96.10	14.40
	✓	✓		93.70	87.40	93.60	7.14
	✓		✓	93.50	90.70	95.80	7.10
	✓	✓	✓	93.00	92.10	96.40	7.14

By replacing the lightweight model, reconstructing the neck and knowledge distillation enhancement on the YOLOv5s baseline network, the AP of the improved network can meet the detection requirements. At the same time, the value of the precise, recall and AP of the model tests have all declined. To compensate for the loss of accuracy caused by replacing the backbone network, we reconstructed the neck network by passing useful information from the redundant feature layer to the neck network for fusion, the size of the passion fruit in the dataset is also taken into account. This approach fuses the low-level semantic information of passion fruit with the high-level location information to obtain more useful feature layers, and the AP increased from 93.10% to 93.60%, an increase of 0.5 percentage points. Finally, using the learning approach of knowledge distillation, the teacher model passes on to the student model rich information about the passion fruit features, and this approach substantially improves the average accuracy of the model. Compared to the YOLOv5s model, the improved model has a mean average precision improvement of 1.00% and a volume reduction of 50.42%.

4.3. Effect of Different Temperatures on the Algorithm

We have found in our distillation experiments that the temperature coefficient has a significant effect on the distillation effect. Therefore, the effect of different temperature coefficients on knowledge distillation results was explored on the basis of the student model G-YOLOv5-N and the teacher model YOLOv5x. The specific approach we employed was to select different temperature coefficients sequentially during the distillation experiments while maintaining a weighting factor of 0.5. This was done to achieve a balance between the knowledge distillation losses and the losses of the original network. The results of the knowledge distillation at different temperatures are shown in Figure 11.

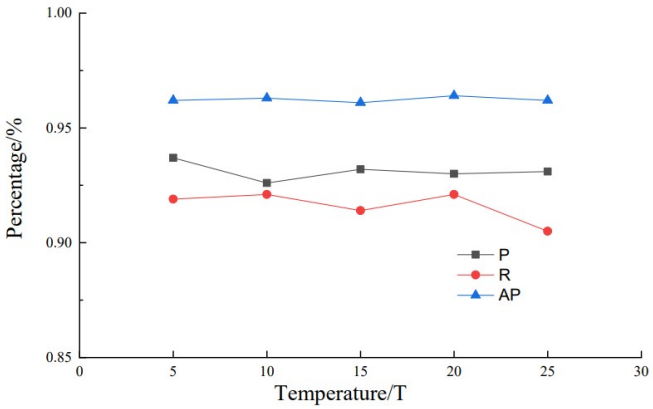


Figure 11. Indicators at different temperatures.

When the temperature coefficient was 20, the distilled model G-YOLO-NK had a high recall and average precision of 92.10% and 96.40% respectively. With different distillation temperature coefficients, the accuracy and recall curves fluctuate up and down, indicating that different temperature coefficients cause the model to focus on different information about passion fruit characteristics. And

 doi:10.20944/preprints202405.1950.v1

the average precision mean always tends to be higher, indicating that the distilled model performs well in identifying passion fruit in complex environments.

4.4. Comparison with State-of-the-Art Models

In order to compare the performance of the improved model with the current mainstream target detection models, the current mainstream models SSD, Faster-RCNN, RetinaNet, YOLOv5s, YOLOv5x, YOLOv6, YOLv7-tiny and YOLOv8s were tested on the Jetson Nano. Comparing indicators such as Floating-point operations per second (FLOPs), parameters, Frames per second (FPS), Precision, Recall, AP and model size on the same self-made dataset. The comparison results are shown in Table 4.

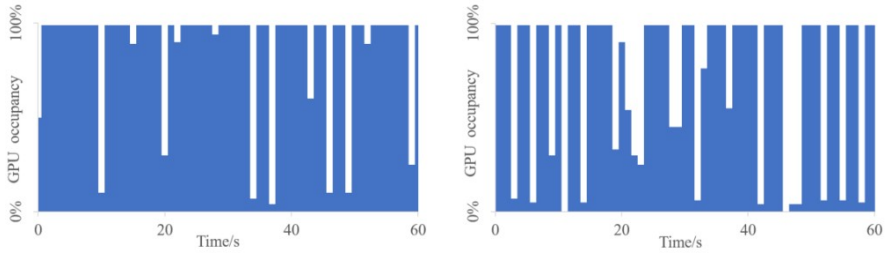
Table 4. Performance comparison of different models

Model	Input	GFLOPs	Param	FPS	PC-FPS	P(%)	R(%)	R(%)	Size
SSD	512*512	61.20	100.10	0.79	5.60	85.48	80.26	80.99	90.60
Faster-Rcnn	600*600	273.40	118.20	0.28	2.96	90.54	87.80	89.90	521.00
RetinaNet	512*512	145.51	36.39	0.58	4.94	75.49	96.00	94.89	138.00
YOLOv5s	640*640	16.30	7.10	6.19	78.74	94.90	90.60	95.40	14.40
YOLOv5x	640*640	203.80	86.17	0.63	50.00	94.90	92.20	96.10	173.21
YOLOv6s	640*640	45.17	18.50	3.10	76.00	75.40	81.20	89.27	38.70
YOLOv7-tiny	640*640	13.00	6.01	7.93	90.14	92.30	89.90	90.20	11.60
YOLOv8s	640*640	28.40	11.13	3.98	83.33	95.70	92.30	95.40	21.40
G-YOLOv5-NK	640*640	6.60	3.51	11.23	125.00	93.00	92.10	96.40	7.14

Due to the instability of the frame rate of real-time detection in Jetson Nano platform, the frame rate of this experiment is the average of 100 detected frame rates. The improved Average Precision (AP) for the G-YOLO-NK model was 96.40%, which is 15.41%, 6.50%, 1.51%, 1.00%, 0.30%, 7.13%, 6.20%, and 1.00% higher compared to the SSD, Faster-Rcnn, RetinaNet, YOLOv5s, YOLOv5x, YOLOv6s, YOLOv7-tiny, and YOLOv8s models, respectively. Obviously, the G-YOLO-NK model has better Average Precision (AP), indicating that the model is capable of detecting passion fruit in complex environments. In terms of real-time detection speed, the G-YOLO-NK model has better real-time detection rate on both the PC and the Jetson Nano. 125.00f/s and 11.23f/s, respectively, compared to the YOLOv5, YOLOv6, YOLOv7-tiny and YOLOv8s models with average frame rate improvements of 10.44f/s, 10.95f/s, 10.65f/s, 5.04f/s, 10.60f/s, 8.13f/s, 3.30f/s, 7.25f/s on the Jetson Nano. The improved model size, FLOPs and the params are 7.14MB, 6.60G and 3.51M, respectively, which is 50.42%, 59.51% and 50.56% reduction of the YOLOv5s model, Further proof of the effectiveness and superiority of the improved network. In summary, the G-YOLO-NK model outperforms the extant models for detecting passion fruit in all metrics and has good overall performance, making it the most promising model for high-performance real-time passion fruit detection.

AI requires an efficient computing power to process a large amount of data. In this process, the GPU has a significant number of cores and high-speed memory, and uses parallel computing processing technology, which can greatly alleviate the bottleneck at the computing level and make deep learning a practical algorithm. GPU utilization is an indicator of how busy various resources on the GPU are. If the GPU usage is too high, jetson nano will experience freezes and crashes in real-time target detection, Long-term GPU usage will affect its performance and lifespan. Therefore, it is necessary to visualize the GPU occupancy. When running YOLOv5s and G-YOLO-NK models on Jetson Nano to detect passion fruit in real time, observe the change of GPU usage over time. The visualization results of the two models are shown in Figure 12.

 doi:10.20944/preprints202405.1950.v1



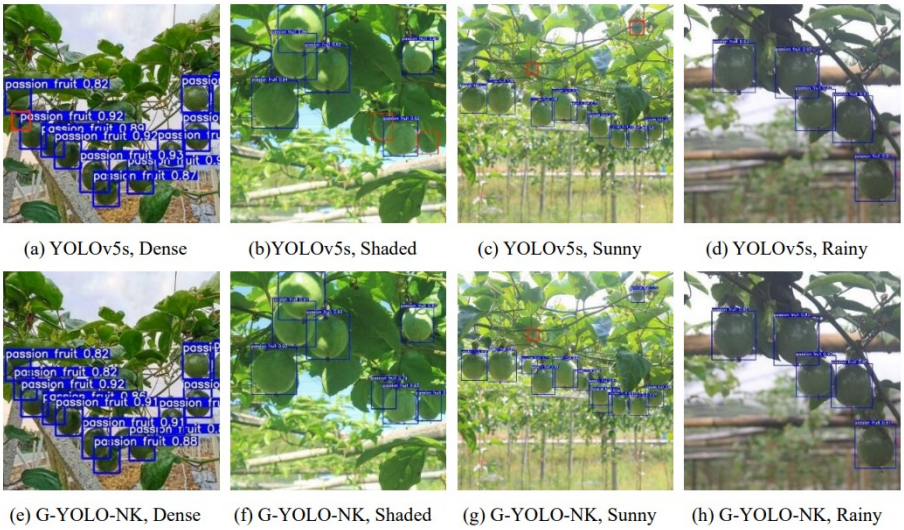
(a) YOLOv5s (b) G-YOLO-NK

Figure 12. Visualisation of GPU utilisation.

As can be seen from the figure 12, the GPU occupancy rates of the two models run by Jetson Nano are different. The more complex the network, the tighter the blue bar graph shown above. It also means that the more GPU resources are required for the embedded device. The model of G-YOLO-NK requires the least amount of computation for real-time detection on the Jetson Nano, which certainly illustrates the importance of model lightweighting and improved effectiveness.

4.5. Comparison of Recognition Effect before and after Improvement

To verify the detection performance of the G-YOLO-NK model, passion fruit images captured in complex environments, including dense, shaded, sunny and rainy conditions, were selected for comparison testing against the original model. A confidence threshold of 0.7 and an IoU threshold of 0.5 were chosen. The detection results of the YOLOv5s and G-YOLO-NK models on embedded devices are shown in Figure 13.



(a) YOLOv5s, Dense (b)YOLOv5s, Shaded (c) YOLOv5s, Sunny (d) YOLOv5s, Rainy

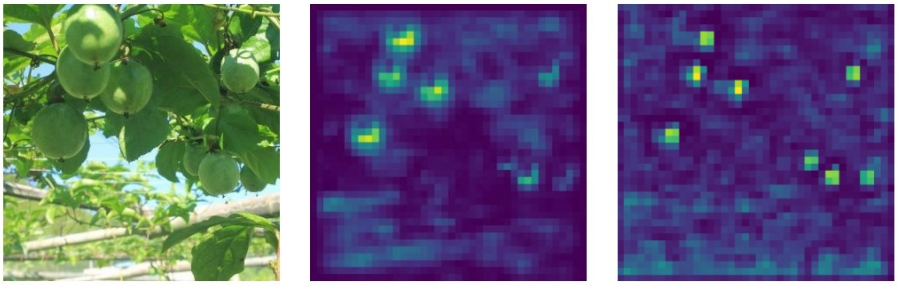
(e) G-YOLO-NK, Dense (f) G-YOLO-NK, Shaded (g) G-YOLO-NK, Sunny (h) G-YOLO-NK, Rainy

Figure 13. Comparison of the detection results of the two models.

Where the blue rectangular box is the predicted target box and the red rectangular box is the missed target. Both types of models correctly detected the passion fruit target in backlight, overcast and rainy weather. The YOLOv5s model produced missed passion fruit detection in both dense and shaded situations, while the G-YOLO-NK model correctly plotted the predicted boxes. The YOLOv5s missed two passion fruits and G-YOLO-NK missed one passion fruit on sunny, because the target was too small. In terms of confidence, G-YOLO-NK has a higher confidence level than the YOLOv5s model, indicating the good detection effect of the improved model and the effectiveness of the improved method. In summary, it is concluded from the recognition that the G-YOLO-NK model improves detection in a jamming environment, with good robustness and generalisation.

 doi:10.20944/preprints202405.1950.v1

The visualisation of the feature layer shows the performance of the model feature extraction and the distribution of contributions to the predicted output, which is more representative in the analysis. Using the masked passion fruit image as an example, the overall feature map of the depth convolution layer of the model before and after the improvement is compared, as shown in Figure 14.



(a) original (b) YOLOv5s (c) G-YOLO-NK

Figure 14. Comparison of detection feature layers.

As can be seen from the image above, there are six highlighted regions before improvement and eight highlighted regions after improvement. Each highlighted display corresponds to the passion fruit in the original image. This indicates that the G-YOLO-NK model has the superior feature extraction ability and accurate prediction ability.

5. Conclusions

This paper proposes a lightweight and high-precision passion fruit target detection algorithm, G-YOLO-NK, based on the improved YOLOv5 algorithm. The proposed algorithm addresses the issues of large parameter scale in general target detection models and poor detection accuracy of passion fruit targets in complex environments. By replacing the YOLOv5 backbone network, reconstructing the neck feature fusion network, and enhancing knowledge distillation, the proposed network achieves a lighter weight while improving detection accuracy. The improved model shows a 1% increase in accuracy and reduces the number and volume of parameters by 50.56% and 51.34%, respectively. The inference speed of the Jeson Nano on the embedded platform was also increased by 5.04f/s to 11.23f/s. The improved algorithm effectively reduces the power consumption of the algorithm and increases the detection speed, enabling passion fruit detection to run on a removable embedded platform with excellent detection performance. Our research is beneficial to the development of smart agriculture and can provide theoretical and technical support for similar work. In addition, to verify that the improved model can be applied to real-world scenarios, experiments in complex environments show that the precision, recall, and average precision of G-YOLO-NK are 93.00%,92.10% and 96.40%, respectively. The model has the highest average precision and the best overall performance compared to SSD, Faster-RCNN, RetinaNet, YOLOv5s, YOLOv5x, YOLOv6, YOLv7-tiny and YOLOv8s models. In the future, we will continue to refine the model to further optimise its detection and improve its performance on small targets. At the same time, this experiment only targets single passion fruit dataset for detection. In the future, we plan to collect passion fruit datasets with varying ripening stages and different colors to enable multi-classification detection of passion fruit. Finally, this study can also be applied to the detection and counting of other fruits, providing assistance for field fruit experiments.

Author Contributions: Conceptualization, P.L.; Methodology, P.L. and Q.S.; Project administration, Q.S. and Z.L.; Software, P.L.and C.H.; Supervision, Q.S.; Validation, P.L.; Visualization, J.Z.; Writing—original draft, P.L. and C.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

 doi:10.20944/preprints202405.1950.v1

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article. You can send an email to the first author and corresponding author to request the data and code.

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

References

1. Fonseca, A.M.; 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 Research International* **2022**, *160*, 111665. CrossRef.
2. Shi, Y.; Jin, S.; Zhao, Y.; Huo, Y.; Liu, L.; Cui, Y. Lightweight force-sensing tomato picking robotic arm with a "global-local" visual servo. *Computers and Electronics in Agriculture* **2023**, *204*, 107549. CrossRef.
3. Ren, H.B.; Feng, B.L.; Wang, H.Y.; Zhang, J.J.; Bai, X.S.; Gao, F.; Yang, Y.; Zhang, Q.; Wang, Y.H.; Wang, L.L.; et al. An electronic sense-based machine learning model to predict formulas and processes for vegetable-fruit beverages. *Computers and Electronics in Agriculture* **2023**, *210*, 107883. CrossRef.
4. Tu, S.; Xue, Y.; Zheng, C.; Qi, Y.; Wan, H.; Mao, L. Detection of passion fruits and maturity classification using Red-Green-Blue Depth images. *Biosystems Engineering* **2018**, *175*, 156–167. CrossRef.
5. Han, L.; Man, Z.; Yu, G.; Minzan, L.; Yuhan, J. Green ripe tomato detection method based on machine vision in greenhouse. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)* **2017**, *33*, 328–334. CrossRef.
6. Yang, X.; Zhang, R.; Zhai, Z.; Pang, Y.; Jin, Z. Machine learning for cultivar classification of apricots (Prunus armeniaca L.) based on shape features. *Scientia Horticulturae* **2019**, *256*, 108524. CrossRef.
7. Li, Z.; Li, Y.; Yang, Y.; Guo, R.; Yang, J.; Yue, J.; Wang, Y. A high-precision detection method of hydroponic lettuce seedlings status based on improved Faster RCNN. *Computers and Electronics in Agriculture* **2021**, *182*, 106054. CrossRef.
8. Sun, H.; Xu, H.; Liu, B.; He, D.; He, J.; Zhang, H.; Geng, N. MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. *Computers and Electronics in Agriculture* **2021**, *189*, 106379. CrossRef.
9. Roy, A.M.; Bose, R.; Bhaduri, J. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Computing and Applications* **2022**, *34*, 3895–3921. CrossRef.
10. Qi, J.; Liu, X.; Liu, K.; Xu, F.; Guo, H.; Tian, X.; Li, M.; Bao, Z.; Li, Y. An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. *Computers and Electronics in Agriculture* **2022**, *194*, 106780. CrossRef.
11. Lawal, M.O. Tomato detection based on modified YOLOv3 framework. *Scientific Reports* **2021**, *11*, 1–11. CrossRef.
12. Roy, A.M.; Bhaduri, J. Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4. *Computers and Electronics in Agriculture* **2022**, *193*, 106694. CrossRef.
13. Lin, Y.; Cai, R.; Lin, P.; Cheng, S. A detection approach for bundled log ends using K-median clustering and improved YOLOv4-Tiny network. *Computers and Electronics in Agriculture* **2022**, *194*, 106700. CrossRef.
14. Li, K.; Wang, J.; Jalil, H.; Wang, H. A fast and lightweight detection algorithm for passion fruit pests based on improved YOLOv5. *Computers and Electronics in Agriculture* **2023**, *204*, 107534. CrossRef.
15. Chen, S.; Zou, X.; Zhou, X.; Xiang, Y.; Wu, M. Study on fusion clustering and improved YOLOv5 algorithm based on multiple occlusion of Camellia oleifera fruit. *Computers and Electronics in Agriculture* **2023**, *206*, 107706. CrossRef.
16. Agarwal, M.; Gupta, S.K.; Biswas, K. Genetic algorithm based approach to compress and accelerate the trained Convolution Neural Network model. *International Journal of Machine Learning and Cybernetics* **2023**, *14*, 2367–2383. CrossRef.
17. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314–1324.
18. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1580–1589.

19. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.
20. Shen, L.; Su, J.; He, R.; Song, L.; Huang, R.; Fang, Y.; Song, Y.; Su, B. Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s. *Computers and Electronics in Agriculture* **2023**, *206*, 107662. CrossRef.
21. He, Y.; Liu, P.; Wang, Z.; Hu, Z.; Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4340–4349.
22. Zhu, J.; Pei, J. Progressive kernel pruning CNN compression method with an adjustable input channel. *Applied Intelligence* **2022**, pp. 1–22. CrossRef.
23. Niu, W.; Ma, X.; Lin, S.; Wang, S.; Qian, X.; Lin, X.; Wang, Y.; Ren, B. PatDNN: Achieving Real-Time DNN Execution on Mobile Devices with Pattern-based Weight Pruning, New York, NY, USA, 2020; pp. 907–922.
24. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2020, Vol. 34, pp. 5191–5198. CrossRef.
25. Wang, L.; Yoon, K.J. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2022**, *44*, 3048–3068. CrossRef.
26. Arablouei, R.; Wang, L.; Currie, L.; Yates, J.; Alvarenga, F.A.; Bishop-Hurley, G.J. Animal behavior classification via deep learning on embedded systems. *Computers and Electronics in Agriculture* **2023**, *207*, 107707. CrossRef.
27. Xu, L.; Wang, Y.; Shi, X.; Tang, Z.; Chen, X.; Wang, Y.; Zou, Z.; Huang, P.; Liu, B.; Yang, N.; et al. Real-time and accurate detection of citrus in complex scenes based on HPL-YOLOv4. *Computers and Electronics in Agriculture* **2023**, *205*, 107590. CrossRef.
28. Tan, Z.; Wang, J.; Sun, X.; Lin, M.; Li, H.; et al. Giraffedet: A heavy-neck paradigm for object detection. In Proceedings of the International conference on learning representations, 2021.
29. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. DAMO-YOLO : A Report on Real-Time Object Detection Design, 2023, [arXiv:cs.CV/2211.15444]. CrossRef.
30. Guo, Y.; Yu, Z.; Hou, Z.; Zhang, W.; Qi, G. Sheep face image dataset and DT-YOLOv5s for sheep breed recognition. *Computers and Electronics in Agriculture* **2023**, *211*, 108027. CrossRef.
31. YANG Jiahao, ZUO Haoxuan, H.Q.S.Q.L.S.L.L. Lightweight Method for Crop Leaf Disease Detection Model Based on YOLO v5s. *Transactions of the Chinese Society for Agricultural Machinery* **2023**, *54*, 222. CrossRef.
32. Zhou, W.H.; Zhu, D.M.; Shi, M.; Li, Z.X.; Duan, M.; Wang, Z.Q.; Zhao, G.L.; Zheng, C.D. Deep images enhancement for turbid underwater images based on unsupervised learning. *Computers and Electronics in Agriculture* **2022**, *202*, 107372. CrossRef.
33. Bhargavi, T.; Sumathi, D. Significance of Data Augmentation in Identifying Plant Diseases using Deep Learning. In Proceedings of the 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2023, pp. 1099–1103. CrossRef.
34. Qiao, Y.; Guo, Y.; He, D. Cattle body detection based on YOLOv5-ASFF for precision livestock farming. *Computers and Electronics in Agriculture* **2023**, *204*, 107579. CrossRef.
35. Ding, J.; Cao, H.; Ding, X.; An, C. High Accuracy Real-Time Insulator String Defect Detection Method Based on Improved YOLOv5. *Frontiers in Energy Research* **2022**, *10*. CrossRef.
36. Sumathi, D.; Alluri, K., Deploying Deep Learning Models for Various Real-Time Applications Using Keras. In *Advanced Deep Learning for Engineers and Scientists: A Practical Approach*; Prakash, K.B.; Kannan, R.; Alexander, S.; Kanagachidambaresan, G.R., Eds.; Springer International Publishing: Cham, 2021; pp. 113–143. CrossRef.
37. Gui, Z.; Chen, J.; Li, Y.; Chen, Z.; Wu, C.; Dong, C. A lightweight tea bud detection model based on Yolov5. *Computers and Electronics in Agriculture* **2023**, *205*, 107636. CrossRef.
38. Yuan, X.; Li, D.; Sun, P.; Wang, G.; Ma, Y. Real-Time Counting and Height Measurement of Nursery Seedlings Based on Ghostnet–YoloV4 Network and Binocular Vision Technology. *Forests* **2022**, *13*. CrossRef.
39. Zhouyi, X.; Weijun, H. Research on multi-target recognition of flowers in landscape garden based on ghostnet and game theory. *Development of Science, Technologies, Education in The XXI Century* **2022**, pp. 46–56.

 doi:10.20944/preprints202405.1950.v1

40. Adelson, E.H.; Anderson, C.H.; Bergen, J.R.; Burt, P.J.; Ogden, J.M. Pyramid methods in image processing. *RCA engineer* **1984**, *29*, 33–41.
41. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944. CrossRef.
42. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768. CrossRef.
43. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision* **2021**, *129*, 1789–1819. CrossRef.
44. Wu, X.; Tang, R. Fast Detection of Passion Fruit with Multi-class Based on YOLOv3. In Proceedings of the Proceedings of 2020 Chinese Intelligent Systems Conference; Jia, Y.; Zhang, W.; Fu, Y., Eds., Singapore, 2021; pp. 818–825. CrossRef.