agronomy

MDPI

Article

GFS-YOLO11: A Maturity Detection Model for Multi-Variety Tomato

Jinfan Wei [1], Lingyun Ni [1], Lan Luo [1], Mengchao Chen [1], Minghui You [1,2], Yu Sun [1,2,*] and Tianli Hu [1,2]

[1] College of Information Technology, Jilin Agricultural University, Changchun 130118, China; weijinfan@mails.jlau.edu.cn (J.W.)
[2] Jilin Province Intelligent Environmental Engineering Research Center, Changchun 130118, China
* Correspondence: sunyu@jlau.edu.cn

Abstract: In order to solve the problems that existing tomato maturity detection methods struggle to take into account both common tomato and cherry tomato varieties in complex field environments (such as light change, occlusion, and fruit overlap) and the model size being too large, this paper proposes a lightweight tomato maturity detection model based on improved YOLO11, named GFS-YOLO11. In order to achieve a lightweight network, we propose the C3k2_Ghost module to replace the C3K2 module in the original network, which can ensure a feature extraction capability and reduce model computation. In order to compensate for the potential feature loss caused by the light weight, this paper proposes a feature-refining module (FRM). After embedding each feature extraction module in the trunk network, it improves the feature expression ability of common tomato and cherry tomato in complex field environments by means of depth-separable convolution, multi-scale pooling, and channel attention and spatial attention mechanisms. In addition, in order to further improve the detection ability of the model for tomatoes of different sizes, the SPPFELAN module is also proposed in this paper. In combining the advantages of SPPF and ELAN, multiple parallel SPPF branches are used to extract features of different levels and perform splicing and fusion. To verify the validity of the method, this study constructed a dataset of 1061 images of common and cherry tomatoes, covering tomatoes in six ripened categories. The experimental results show that the performance of the GFS-YOLO11 model is significantly improved compared with the original model; the P, R, mAP50, and MAP50-95 increased by 5.8%, 4.9%, 6.2%, and 5.5%, respectively, and the number of parameters and calculation amount were reduced by 35.9% and 22.5%, respectively. The GFS-YOLO11 model is lightweight while maintaining high precision, can effectively cope with complex field environments, and more conveniently meet the needs of real-time maturity detection of common tomatoes and cherry tomatoes.

Keywords: maturity detection; complex field environment; lightweight model; YOLO11; feature refining; multi-scale feature fusion

check for updates

Citation: Wei, J.; Ni, L.; Luo, L.; Chen, M.; You, M.; Sun, Y.; Hu, T. GFS-YOLO11: A Maturity Detection Model for Multi-Variety Tomato. *Agronomy* **2024**, *14*, 2644. https://doi.org/10.3390/agronomy14112644

Academic Editor: Yuxing Han

Received: 23 October 2024
Revised: 4 November 2024
Accepted: 7 November 2024
Published: 9 November 2024

1. Introduction

Tomato is one of the most important cash crops in the world [1], and its maturity is directly related to the taste, nutritional value, and storage time of the fruit. The hardness, color, flavor, and nutritional content of tomatoes at different ripening stages are significantly different, which affects consumers' purchase intention and market price [2]. Globally, approximately 20–30% of fruits and vegetables suffer post-harvest losses each year due to inappropriate ripeness assessments and harvest timing, and this includes a significant number of tomatoes [3]. The huge production and high post-harvest loss rates highlight the importance of accurate ripening assessment in the tomato industry. In addition, traditional tomato ripeness detection mainly relies on manual experience for visual inspection, which has defects such as low efficiency, strong subjectivity, and easy interference by human factors. For example, experienced workers can only inspect a limited number of tomatoes per day, and assessment criteria vary between workers, resulting in less consistent inspection results. In addition, manual inspection is time-consuming and increases labor costs, especially in large-scale tomato production, and it is difficult to meet the needs of modern agriculture's fine and large-scale production. Therefore, there is an urgent need to explore efficient, accurate, and objective detection methods for tomato maturity to overcome the limitations of traditional approaches and achieve rapid, non-destructive, and consistent maturity assessments. This will improve tomato quality grading, optimize the picking timing, reduce post-harvest losses, lower labor costs, and ultimately promote the healthy development of the tomato industry and provide consumers with higher-quality tomato products.

In order to overcome the limitations of traditional methods, more and more researchers have begun to pay attention to automated fruit and vegetable maturity detection technology. For a long time in the past, researchers explored the use of digital image processing technology and machine learning algorithms to automatically identify the maturity of various fruits and vegetables. These methods first use image sensors to collect fruit and vegetable images; then extract color, shape, texture, and other features through image processing algorithms; and finally use machine learning algorithms to build maturity discrimination models. For example, Rahim Azadnia et al. [4] developed an automated algorithm based on machine learning to improve the automatic assessment of hawthorn maturity. The geometric attributes, color and texture features of hawthorn were extracted by image processing technology, and the feature dimension was reduced by quadratic discriminant analysis (QDA), and then classified by an artificial neural network (ANN) and support vector machine (SVM). The results showed that the ANN model based on high-efficiency features reached 99.57%, 99.16%, and 98.16% accuracy in the training, verification, and testing stages, respectively, which provided a rapid, accurate, and non-destructive detection method for the maturity assessment of hawthorn. Ferhat Kurtulmus et al. [5] developed a machine vision algorithm based on color images in order to detect unripe green citrus in natural outdoor conditions. The algorithm uses color threshold segmentation, a PCA-based 'eigenfruit' method, and circular Gabor texture analysis to identify green citrus. The result of the subwindow classifier is determined by moving a subwindow on three different scales to scan the whole image and implementing the majority voting method. On the verification set, the algorithm successfully detected 75.3% of the actual fruits, demonstrating the potential of using conventional color images for green citrus detection under natural conditions. Luiz Fernando Santos Pereira et al. [6] used digital imaging and random forest methods to predict the ripening of papaya fruit. Physical and chemical analysis were performed to determine the true ripening stage of the fruit. They extracted 21 manual color features using image analysis and then implemented a random decision forest to predict the maturity stage. In the experiment, 114 samples were used, a classification performance of 94.3% was obtained on the cross-validation set, and the accuracy of the prediction set was 94.7%. Although these methods have made some progress in the detection of fruit and vegetable maturity, their limitations are becoming increasingly prominent. These methods often rely on artificially designed color and texture features, which are difficult to adapt to the color diversity of fruits and vegetables and the complex field environment. For example, a fixed color threshold struggles to handle the color changes caused by changes in light and individual differences in fruits and vegetables. However, the texture feature extraction method based on an artificial design filter has the problems of low computational efficiency and insufficient generalization ability, and it is difficult to deal with a wide variety of fruit and vegetable textures.

In recent years, with the rapid development of deep learning technology, especially the emergence of convolutional neural networks, fruit and vegetable maturity detection methods based on deep learning have gradually become a research hotspot [7]. Among them, target detection technology is widely used in the maturity detection of fruits and vegetables. For example, to improve the real-time detection efficiency of cherry tomato ripening, Congyue Wang et al. [8] developed an improved YOLOv5n model. The model uses the K-Means ++ algorithm to optimize an anchor frame size, employs a coordinate attention mechanism to extend the perceptual domain, and adopts a boundary frame regression loss function (WIoU) with dynamic focusing. The results show that the accuracy and recall rate of the improved model improved by 1.4%, reaching 95.2% mAP; the average detection time was 5.3 ms, and the model size was only 4.4 MB, which is suitable for real-time and lightweight applications. Chenglin Wang et al. [9] designed the YOLO-BLBE model, which combined I-MSRCR enhanced color features, the GhostNet model, the CA mechanism module, a BIFPN structure, and the Alpha-EIOU loss function to improve the recognition efficiency of blueberry maturity. The experimental results showed that the recognition accuracy of the model was 99.58%, 96.77%, and 98.07%, respectively. The model size was 12.75 MB, and the average detection speed was 0.009 s. Defang Xu et al. [10] proposed the YOLO-RFEW model for the intelligent detection of melon maturity under an artificial greenhouse environment. Based on the improvement of YOLOv8n, RFAConv was used to enhance feature extraction, the C2f module was optimized, and the C2fFE and EMA attention mechanisms were combined. The improved WIoU loss function improves the prediction accuracy. The accuracy, recall rate, F1 score, and mAP of the YOLO-RFEW model reached 93.16%, 83.22%, 87.91%, and 90.82%, respectively. The model size was 4.75 MB, and the detection time was 1.5 ms. Xiangyang Sun et al. [11] artificially improved the detection accuracy of greenhouse tomatoes, developed the S-YOLO model, adopted lightweight a GSConv-SlimNeck structure, improved a-SimSPPF, and enhanced the b-SIoU algorithm and SE attention module. The detection accuracy of the S-YOLO model was 96.60%, the mAP was 92.46%, and the detection speed reached 74.05 FPS, which were, respectively, increased by 5.25%, 2.1%, and 3.49 FPS compared with the original model. The model parameter was 9.1m, which improved the problems of occlusion and tomato recognition, and supported the vision system of the tomato-picking robot. Ping Li et al. [12] proposed a new detection method based on MHSA-YOLOv8 for the automatic grading of tomato maturity and counting of tomatoes. Through introducing the MHSA attention mechanism, this method improves the performance of the model in complex scenarios and provides technical support for uncrewed operation robots in tomato picking. Despite the presence of occlusion and light interference, the model can still grade tomato maturity and count tomatoes online. Future research will be dedicated to alleviating these interferences and further enhancing the performance of the model. Renzhi Li et al. [13] proposed a tomato maturity recognition model based on the improved YOLOv5 to enhance the recognition accuracy and speed of greenhouse tomato maturity. This model optimizes the regression process of the prediction box by adopting Mosaic data enhancement, Focus and CSPNet network structures, and the EIoU loss function. Experimental results showed that the model achieved an accuracy of 95.58%, a recall rate of 90.07%, and an average precision of 97.42% on the test set, which were increased by 0.11% and 0.66%, respectively, compared to the original YOLOv5s model. The single-image detection speed of the model was 9.2 milliseconds, and the model size was 23.9 MB, meeting the accuracy and speed requirements for greenhouse tomato maturity recognition.

As mentioned above, the existing research on fruit and vegetable maturity based on target detection has made some progress, but there are still some problems to be solved. For example, most of the existing tomato maturity detection methods are aimed at a single tomato variety, ignoring the differences in color, shape, size, etc., among different varieties. When dealing with complex field environments (such as light changes, occlusion, fruit overlap, etc.) [14] and images of different tomato varieties (such as common tomatoes and cherry tomatoes), the recognition accuracy and generalization ability need to be improved. In addition, existing models often have high computational complexity and large numbers of parameters and are difficult to deploy on mobile or embedded devices with limited resources, which limits their application in practical scenarios. In order to solve the above problems, a lightweight tomato maturity detection model GFS-YOLO11 based on improved YOLO11 is proposed in this paper. The main contributions of this paper are as follows:

- An efficient lightweight model, GFS-YOLO11, is proposed. In order to meet the requirements of the real-time maturity detection of common tomato and cherry tomato, the model not only guarantees the recognition accuracy but also focuses on optimizing the model structure to reduce the number of parameters and calculations, making it easier to deploy on mobile devices.
- C3k2_Ghost module: This module generates redundant feature maps through inexpensive linear transformations, effectively reducing the computational burden of traditional convolution operations, thus achieving a lightweight model.
- FRM: Considering that lightweight operation may lead to information loss, we propose a feature-refining module (FRM) to enhance the feature expression ability of the model and improve the identification accuracy of tomatoes of different sizes and different ripening stages.
- SPPFELAN module: In combining the advantages of SPPF and ELAN, this module further improves the detection ability of common tomatoes and cherry tomatoes.
- A diverse dataset containing common tomatoes and cherry tomatoes was constructed to train and evaluate the model performance and provide data support for related studies.

2. Materials and Methods

2.1. Production of Datasets

2.1.1. Data Sample Collection

In order to comprehensively evaluate the performance of the GFS-YOLO11 model proposed in this paper on the maturity detection of common tomatoes and cherry tomatoes, we established an image dataset of common tomatoes and cherry tomatoes covering various scenes, named Tomato-Detect. The dataset was derived from the tomato laboratory greenhouse of Jilin Agricultural University and consisted of 1061 high-resolution color images after manually screening out low-quality images. The images cover tomatoes of different ripening stages, shooting angles, lighting conditions, occlusion degrees, and fruit sizes, striving to truly reflect the complexity of the field environment and improve the generalization ability of the model. Some images are shown in Figure 1. To ensure the effectiveness and authenticity of the model training and evaluation, we randomly divided the dataset into a training set and a validation set in an 8:2 ratio.

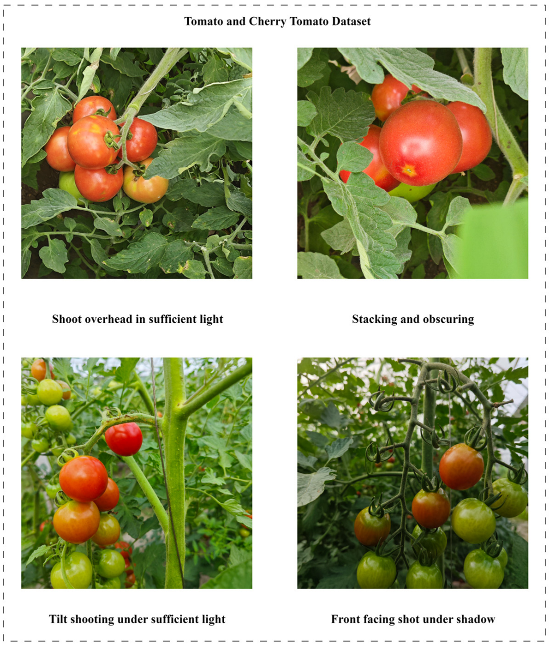
Tomato and Cherry Tomato Dataset
Shoot overhead in sufficient light
Stacking and obscuring
Tilt shooting under sufficient light
Front facing shot under shadow

Figure 1. In order to create a dataset of ordinary tomatoes and cherry tomatoes with a variety of light environments, shooting angles, and occlusion conditions, we paid special attention to the following

scenarios when capturing images: (1) shooting from above under bright light; (2) the shooting angle when the object is partially occluded or overlapped; (3) shooting from the side under sufficient lighting conditions; (4) shooting from the front in a low-light environment.

2.1.2. Dataset Enhancement

In order to enhance the generalization ability of the model and prevent the model from overfitting specific features in the training data (for example, specific lighting conditions, shooting angles, or tomato morphology), we performed data enhancement on the training set of the dataset [15]. Specifically, we used a series of random transformation strategies to extend the original images, including randomly rotating the image by 15 to 45 degrees, randomly flipping the image, adding random noise, and randomly shifting the image position. These operations effectively simulated the growth posture, lighting conditions, and shooting angle changes in real scenes of common tomatoes and cherry tomatoes, thereby increasing the diversity of the training samples. A visualization of the data enhancement is shown in Figure 2. Through data enhancement, we expanded the training set by 4 times, and the specific data distribution is shown in Table 1.

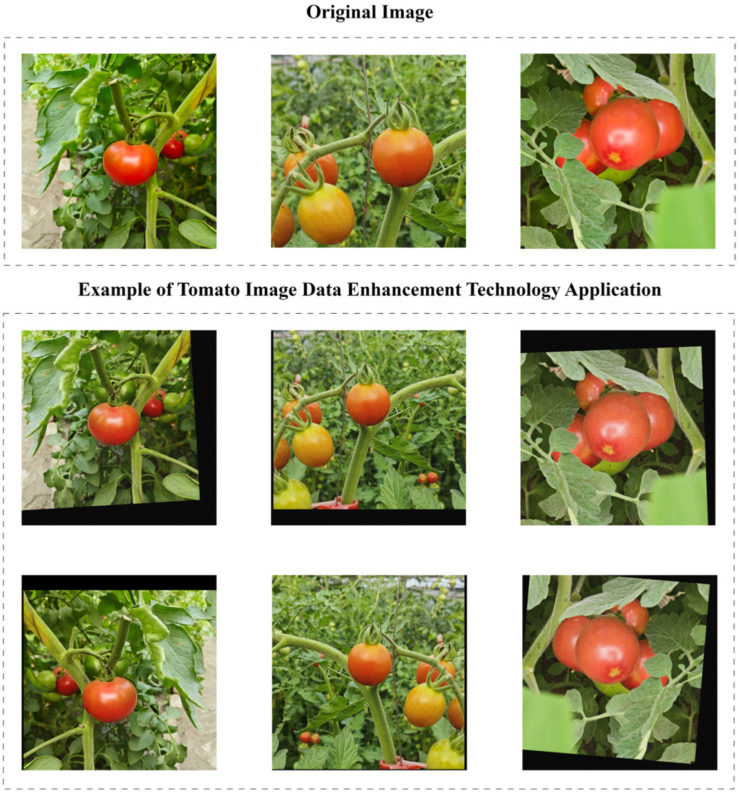
Original Image
Example of Tomato Image Data Enhancement Technology Application

Figure 2. Examples of data enhancement techniques: random processing of images, including rotation in the range of 15 to 45 degrees, horizontal flipping, introduction of random noise, and horizontal or vertical translation.

Table 1. This table shows the composition of the dataset, covering the total number of images in the original training set, the enhanced training set, and the validation set, and the number of samples in six different categories. Specifically, the six categories correspond to three stages in the ripening process for regular and cherry tomatoes: full ripeness, semi-ripeness, and under-ripeness.

	Train		**Val**
Images **Instances**	**Original (848)**	**Enhance (2544)**	**Original (213)**
Large, fully mature	584	1752	128
Large, semi-mature	633	1899	139
Large, immature	1500	4500	354
Small, fully mature	1025	3075	246
Small, semi-mature	854	2562	262
Small, immature	3592	10,776	1006
All	8188	24,564	2135

2.2. Model Improvement

In this study, we chose the YOLO11 [16] model as our infrastructure, and this choice was based on multiple considerations. Firstly, YOLO11 is an efficient object detection algorithm newly developed by the Ultralytics team. It inherits the fast and accurate excellent characteristics of the YOLO series of algorithms. The YOLO series is widely praised for its outstanding performance in the field of real-time object detection, and YOLO11, as the latest member of this series, has demonstrated excellent performance in various complex visual tasks through its advanced neural network architecture and optimized training strategies. Secondly, the introduction of YOLO11 represents the latest progress in current object detection technology, providing a strong starting point for our research direction. We believe that standing on the shoulders of giants can further the depth and cutting-edge qualities of our research. Moreover, the advancement of YOLO11 also means that it has better adaptability and scalability, which is crucial for us to address the challenges in specific visual tasks. However, we are also aware of some shortcomings of the YOLO11 model in practical applications. For example, the model has a large volume, which makes it difficult to deploy on resource-constrained platforms such as mobile devices. This challenge is particularly important for our research because our goal was to develop an efficient and practical object detection system. At the same time, when dealing with targets with large size differences such as regular tomatoes and cherry tomatoes, there is still room for improvement in the detection accuracy of YOLO11. This indicates that although YOLO11 performs well in many aspects, in specific fields, such as the visual recognition of agricultural products, further optimization and adjustment are still needed. In order to solve these problems, the GFS-YOLO11 model was improved in the following three aspects: The C3k2_Ghost module was proposed to replace the C3k2 module in the original network to reduce the computational complexity and memory consumption of the model, and improve the reasoning speed of the model. Considering that the lightweight design may lose part of the feature information, we embedded an FRM after each feature extraction module (C3k2) in the backbone network to improve the feature expression ability of common tomatoes and cherry tomatoes in complex field environments and enhance the detection accuracy of the model for different mature stages and varieties of tomatoes. The SPPFELAN module was proposed to replace the SPPF module in the original network, which further improves the detection ability for different sizes of common tomatoes and cherry tomatoes. The improved algorithm model structure is shown in Figure 3.

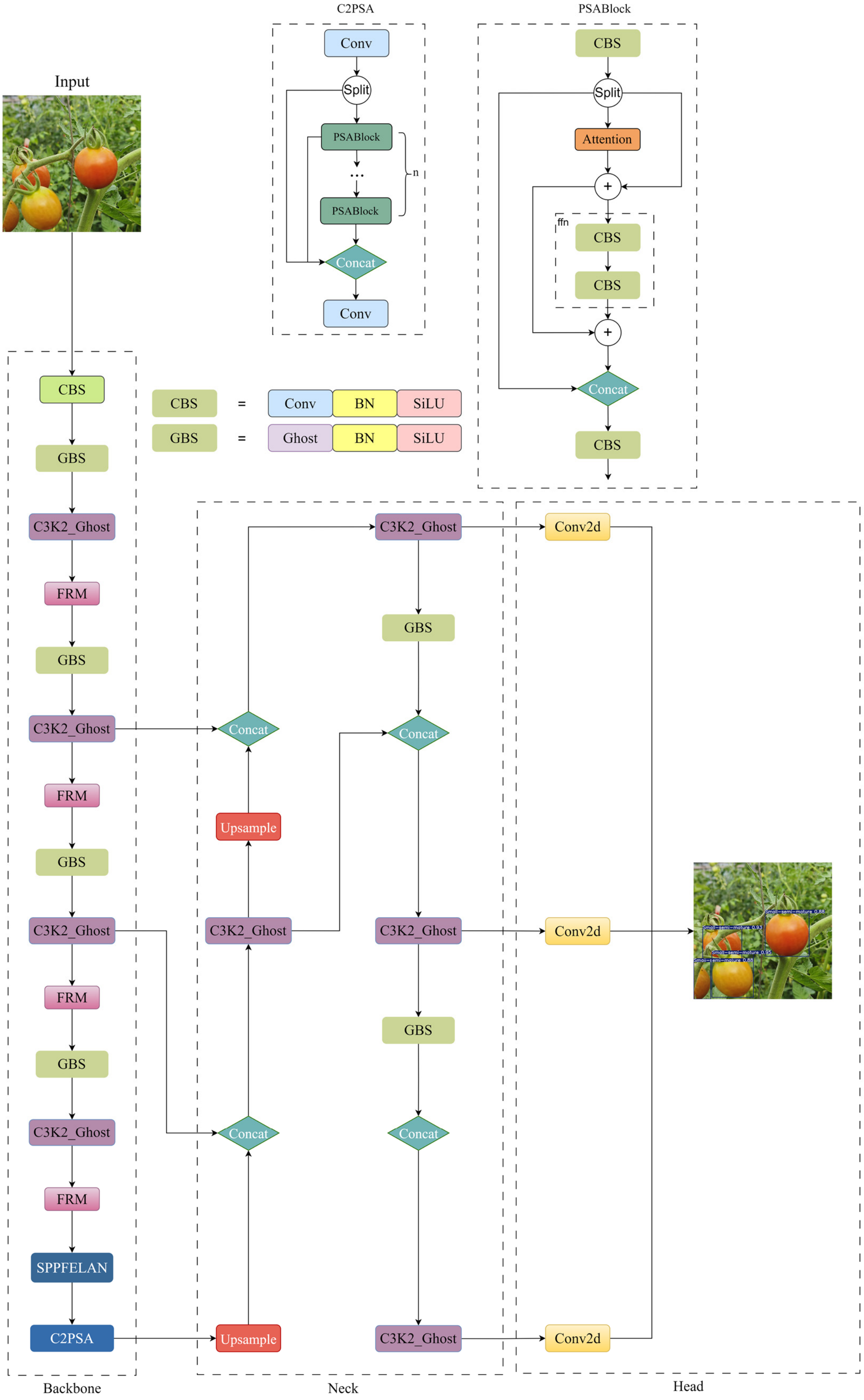

Figure 3. Model structure diagram of GFS-YOLO11.

2.2.1. C3K2_Ghost

In the optimization of deep neural network models, being lightweight is a critical goal, especially when deploying models on resource-constrained devices [17]. In this study, a lightweight improvement method for the C3k2 feature extraction module was designed. With the introduction of the GhostBottleneck [18] structure to replace the traditional bottleneck structure, the model's computing cost and memory usage are significantly reduced without ensuring the model's performance. The module structure is shown in Figure 4.

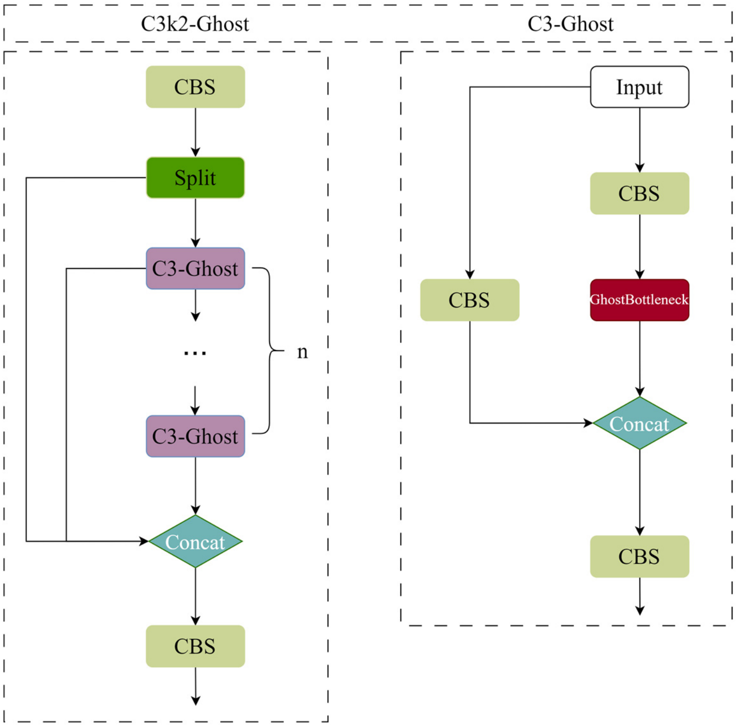

Figure 4. Network structure of the C3K2_Ghost module.

The C3K2 module is a feature extraction module of the latest YOLO11 model. Based on the CSPNet structure [19], it handles the input feature graph by dividing it into two parts and using a bottleneck module for multi-scale feature extraction. However, traditional bottleneck structures usually contain multiple convolution layers and require a lot of computation. The introduction of the GhostBottleneck structure solves this problem effectively. Compared with the traditional bottleneck structure, GhostBottleneck proposes a more efficient way for feature extraction, which uses GhostConv operations as its core. GhostConv generates part of the feature map with fewer convolution cores and then expands it with inexpensive linear transformation operations to generate more diverse feature representations, thus significantly reducing the computational effort while maintaining performance. Finally, the two parts of the feature map are spliced together to obtain the final output feature map. This design philosophy causes GhostBottleneck to maintain or even improve the model's performance while significantly reducing the number of parameters and computation. By replacing the bottleneck structure in the C3k2 module with the GhostBottleneck structure, we can greatly reduce the computational cost at the feature extraction stage of the model, thus achieving a lightweight model.

The immediate benefits of this improvement include faster inference speeds and smaller model sizes. The faster inference speed allows the model to better meet the needs of real-time applications.

2.2.2. FRM

Although a lightweight method can effectively reduce the computational cost, it may also cause the sensitivity of the model to some features, such as the sensitivity of the surface texture of the tomato, the color gradient, and other subtle features [20]. To solve this problem, this paper presents an innovative structure named feature refinement module (FRM) and embedded it after the C3K2 module of the YOLO11 backbone network. This module aims to enhance the capturing ability of feature representations by integrating local features and global context information so as to improve the performance of the model in the maturity detection of common tomato and cherry tomato. The model structure is shown in Figure 5.

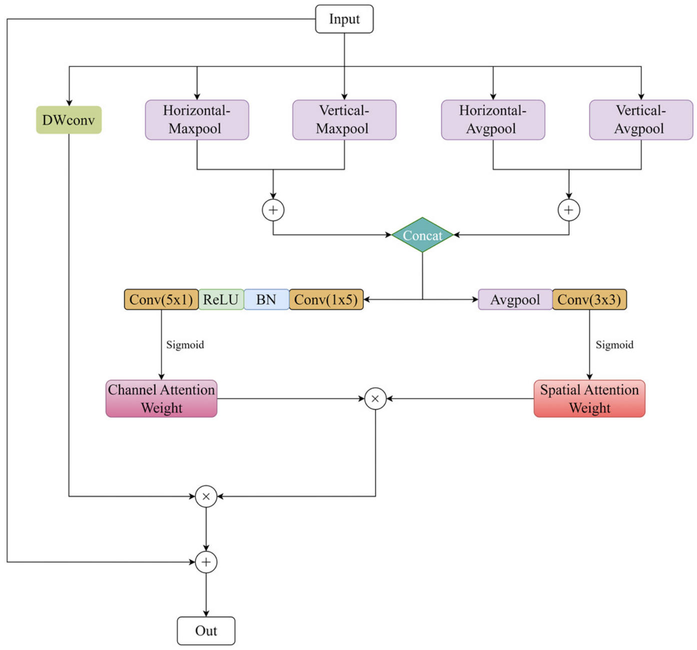

Figure 5. Model structure diagram of FRM.

The core idea of the FRM is to focus on both local details and global context information. For local features, the FRM uses efficient depth-separable convolution [21] to extract them. This method significantly reduces the computational cost while maintaining the sensitivity to local spatial information, which is conducive to capturing subtle features such as texture and color changes on the tomato surface, which are crucial for distinguishing the tomato species and maturity. The acquisition of global context information depends on multi-scale pooling operations. The FRM uses average pooling and maximum pooling in horizontal and vertical directions to effectively capture feature information at different scales. The FRM fuses these multi-scale features in a process that can be described as

$$\mathrm{X-MSFF} = Concat\left[AvgPool_{h(x)} + AvgPool_{v(x)}, MaxPool_{h(x)} + MaxPool_{v(x)}\right] \quad (1)$$

Here, $AvgPool_h$ and $AvgPool_v$ represent average pooling operations in the horizontal and vertical directions, respectively; $MaxPool_h$ and $MaxPool_v$ represent maximum pooling operations in the horizontal and vertical directions, respectively; Concat represents concatenation operations; and X-MSFF represents multi-scale features after fusion. Average pooling can reflect the overall statistical characteristics of feature maps, while maximum pooling highlights the most significant local features. The combination of horizontal and vertical pooling results can describe the spatial distribution of features more comprehensively, which helps to understand the overall shape and color distribution of tomatoes and then judge the tomato species more accurately. In order to further enhance the ability of feature expression, the FRM introduces channel attention and spatial attention mechanisms.

The channel attention mechanism focuses on screening important feature channels, and its calculation process is as follows:

$$CA - W = Sigmoid(Conv_2(ReLU(BN(Conv_1(X - MSFF))))) \tag{2}$$

Here, $Conv_1$ and $Conv_2$ represent two-layer packet convolution operations,BN represents batch normalization operations, $ReLU$ represents activation functions, and CA-W represents the channel attention weight. The design of group convolution can effectively reduce the computational complexity while maintaining the ability to model the relationship between channels. Subsequently, batch normalization and ReLU activation functions are used to enhance the nonlinear representation of the network. Finally, the sigmoid gating function is used to generate channel attention weights to highlight important feature channels and suppress irrelevant channel information. The spatial attention mechanism focuses on the importance of different positions in the feature map. Firstly, the FRM performs global average pooling on the fused multi-scale features to obtain a global feature vector, which represents the whole features of the image. Then, through a convolution operation with a kernel size of 3, the global feature vector is transformed to generate the spatial attention weight. This process can be expressed by the formula:

$$SA - W = Sigmoid(Conv(AvgPool(X - MSFF))) \tag{3}$$

Here, $AvgPool$ represents the global average pooling operation, $Conv$ represents the convolution operation, and $SA - W$ represents the spatial attention weight. Finally, the sigmoid activation function is used to normalize the value of the attention diagram to between 0 and 1, indicating the importance of each location.

The channel attention diagram and space attention diagram are multiplied by elements to obtain the final attention mask. The mask takes into account the importance of features in both the channel dimension and spatial dimension to more precisely guide the network to focus on important areas and features, such as color changes or texture features on the tomato surface. The application of the attention mask to extracted local features can effectively enhance key features and suppress background noise, thus improving the discriminability of feature representation.

Finally, the FRM adopts the residual connection structure. The residual connection adds the input of the module directly to the output, effectively alleviates the problem of gradient disappearance, promotes the training of the network, and retains the original feature information. This design enables the FRM to be better integrated into the backbone network of YOLO11 and improve the overall detection performance.

2.2.3. SPPFELAN

In target detection tasks, the efficient extraction and fusion of multi-scale features is essential for the recognition of objects of different sizes and shapes [22]. SPP and its efficient version SPPF use fixed-size pooling kernels to check the input feature map and extract multi-scale features, which effectively improves the detection ability of the model for objects of different sizes. However, SPP and SPPF usually use simple concatenation operations to integrate features extracted from different pooling layers, which may not make full use of the complementary information between multi-scale features, limiting the performance of the model. ELANs (Efficient Layer Aggregation Networks) [23] have shown unique advantages in feature fusion. Through parallel multiple branches, they extract different levels of feature information and finally perform efficient aggregation, which improves performance while maintaining low computing costs.

In order to combine the advantages of SPPF and ELANs and further improve the model's ability to detect the maturity of tomatoes of different sizes, a new module named SPPFELAN was proposed in this paper. Its structure is shown in Figure 6.

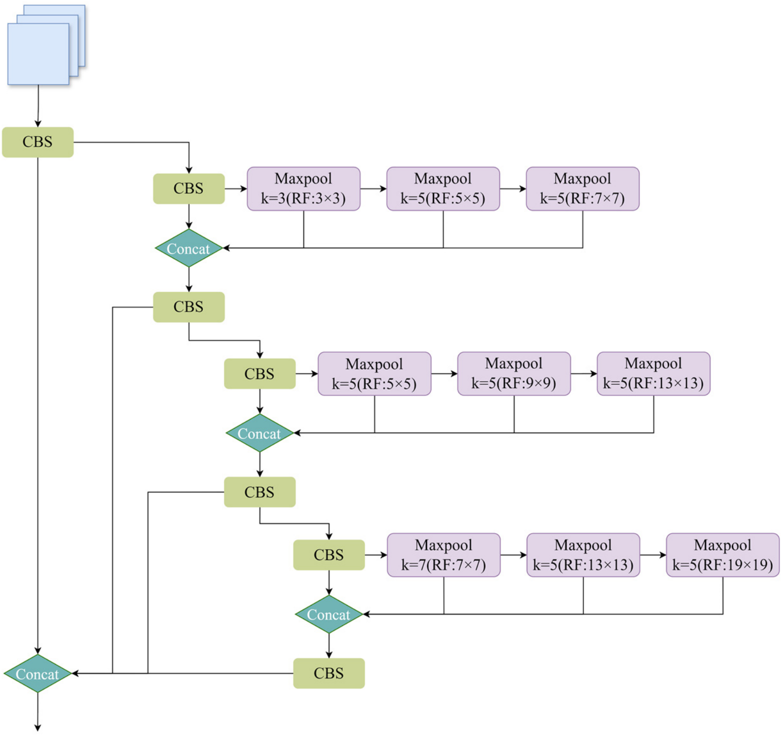

Figure 6. SPPFELAN model structure diagram.

The module borrows from the ELAN's idea of paralleling multiple branches to extract multi-scale features and cleverly uses the SPPF module as the concrete implementation of the branches. Each SPPF module has pooling kernels of different sizes (3, 5, 7), which can extract different levels of feature information: large-size pooling kernels can capture global features such as the overall shape and color distribution of the tomato, while small-size pooling kernels can focus on detailed features such as peel texture and local defects. Finally, the SPPFELAN module splices and fuses the output of all branches, which is simple and efficient and can integrate the feature information from different levels to form a more comprehensive feature expression, thus further improving the model's ability to detect the maturity of tomatoes of different sizes.

2.3. Evaluation Indicators

In order to comprehensively evaluate the performance of the model in the detection tasks of common tomato and cherry tomato maturity, this paper adopted commonly used evaluation indicators in the field of target detection, including precision (P), recall (R), mAP50, and MAP50-95.

Precision (P) focuses on measuring the accuracy of the model identification results, that is, the proportion of samples that the model judges as ripe tomatoes that are really ripe. In tomato ripeness detection, high accuracy means that the model is able to identify ripe tomatoes more accurately, reducing the number of cases where immature tomatoes or other objects are misjudged as ripe tomatoes. Its calculation formula is as follows:

$$p = \frac{TP}{TP + FP} \tag{4}$$

Here, TP (True Positive) refers to situations in which the model correctly classifies actual ripe tomatoes as ripe, and FP (False Positive) refers to situations in which the model incorrectly classifies actual unripe tomatoes or other objects as ripe.

The recall rate (R) represents the proportion of all images that actually contained ripe tomatoes that were correctly detected by the model. The recall rate measures the comprehensiveness of the model, i.e., whether ripe tomato targets are missed. In ripened tomato detection, a high recall rate means that the model is able to identify as many ripe tomatoes in the image as possible, avoiding omissions and allowing for more efficient subsequent processing, such as picking or grading. Its calculation formula is as follows:

$$R = \frac{TP}{TP + FN} \tag{5}$$

Here, FN (False Negative) refers to a situation in which the model incorrectly classifies an actual ripe tomato as immature or another object.

The average accuracy mean (mAP) is an index that combines the model accuracy and recall rate. The average precision (AP) value ranges from 0 to 1, with higher values indicating better model performance. The mAP is the average of AP values for all categories. In tomato maturity detection, we usually pay attention to mAP50 (the mAP when the IOU threshold is 0.5, paying special attention to the accuracy of matching the detection box with the real box) and mAP50-95 (the average mAP value when the IOU threshold changes from 0.5 to 0.95, with 0.05 as the step, providing the performance under different IOU thresholds and more comprehensively reflecting the average performance of the model under different IOU thresholds). The calculation formulas of these two indicators are as follows:

$$AP = \int_0^1 p(r)dr \tag{6}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{7}$$

In practical applications, we hope that the model will be able to not only identify ripe tomatoes but also accurately locate the location of the tomato for precise manipulation. The mAP, as a comprehensive index, can more comprehensively evaluate the comprehensive performance of the model in different tomato maturity detection tasks. In addition, the efficiency evaluation indexes of the model include memory occupation, parameters, computational complexity (GFLOPs), and inference time, which are used to comprehensively evaluate the model's demand for hardware resources and inference speed.

3. Results

3.1. Experimental Environment and Parameter Setting

This experiment was built on a PyTorch deep learning framework and executed in the Anaconda environment. Table 2 shows the main experimental equipment environment configuration, and Table 3 shows the main hyperparameter settings.

Table 2. Experimental environment configuration.

Environment Configuration	**Parameter**
Operating system	Linux
CPU	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
GPU	2 × A100 (80 GB)
Development environment	PyCharm 2023.2.5
Language	Python 3.8.10
frame	PyTorch 2.0.1
Operating platform	CUDA 11.8

Table 3. Hyperparameter settings.

Hyperparameter	**Parameter**
Epochs	200
Batch	64
AdamW learning rate	0.000714
Momentum	0.9
Weight decay	0.0005
Input image size	640

3.2. Experimental Results of GFS-YOLO11 Model

Figure 7 shows the performance of the GFS-YOLO11 model on the Tomato-Detect dataset. It can be clearly seen from the figure that during the training process, various loss indicators of the model change with the number of iterations, as well as the precision, recall rate, and mAP indicators under the boundary box, all of which jointly reflect the overall performance of the model in the detection task.

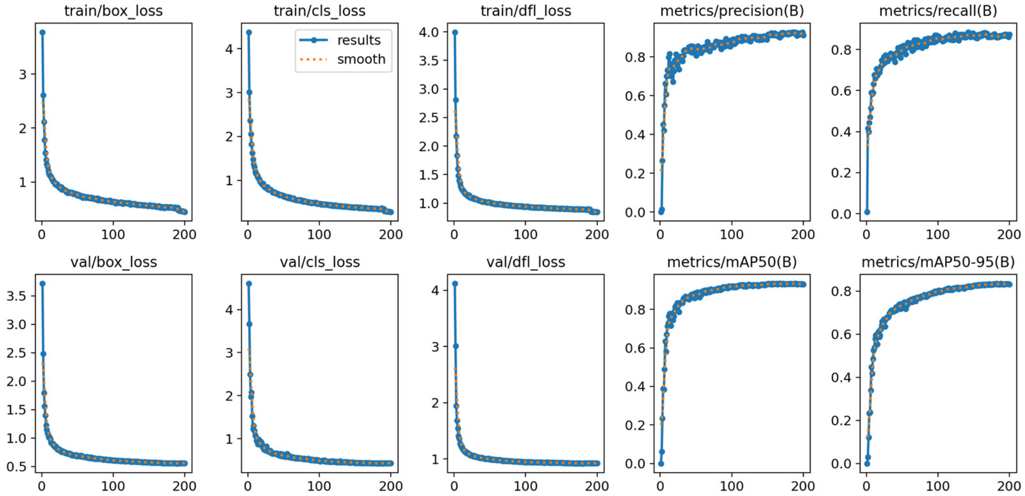

Figure 7. Experimental results of GFS-YOLO11 model.

To more fully assess the model's ability to detect maturity for both common and cherry tomatoes under different confidence thresholds, Figure 8 shows the F1 score curve for each category. The F1 score, as a comprehensive index used to consider the accuracy and recall rate of the model, can effectively reflect the overall performance of the model.

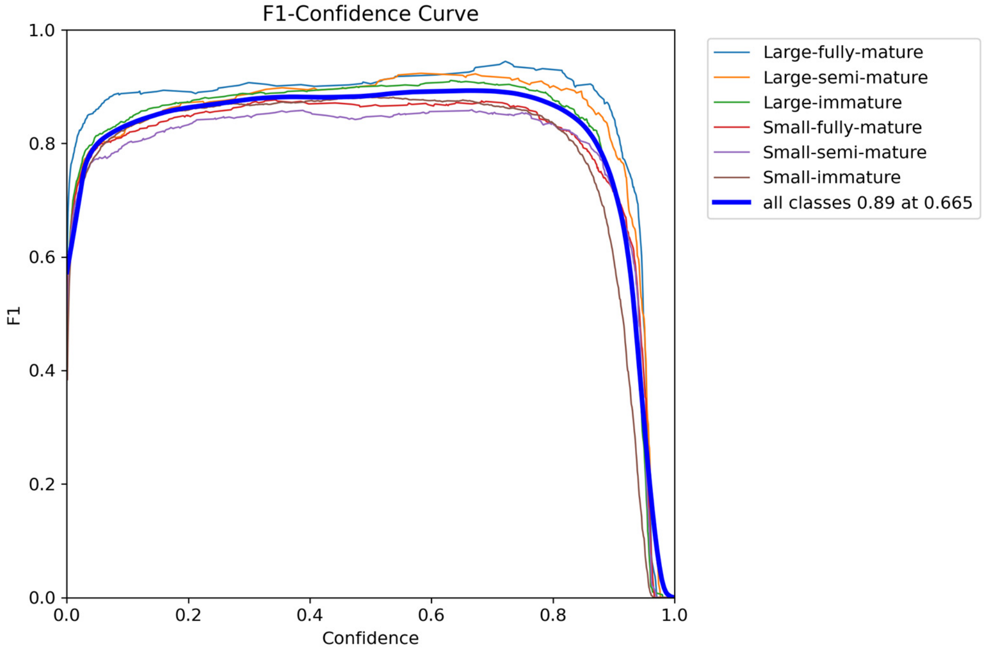

Figure 8. F1 fraction curve of GFS-YOLO11 model.

Further, Figure 9 depicts the P-R curve of the model, which intuitively shows the balance between the accuracy and recall rate of the model under different confidence thresholds. The trend in the curve shows that the model as a whole presents a relatively smooth P-R curve, and can maintain a good level of accuracy while maintaining a high recall rate. This shows that the model can effectively control the false detection rate and achieve a satisfactory recognition effect while ensuring the comprehensive detection of tomatoes at each maturity level.

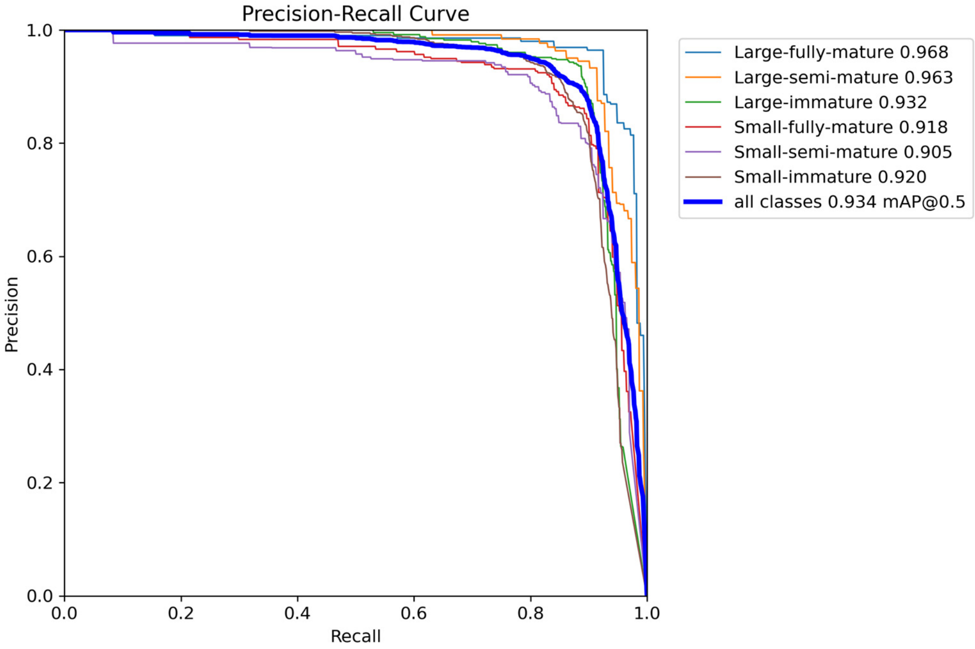

Figure 9. Precision–Recall curve of GFS-YOLO11 model.

3.3. Comparative Experiments of Different Models

In order to verify the effectiveness of the proposed method, we selected several mainstream target detection models for comparative experiments, including the YOLO series (v3-tiny [24], v5s [25], v6 [26], v7 [23], v8s [16], v9c [27], v9e [27], v10s [28], v11 [16]), the RT-Detr series [29], and the GFS-YOLO11 model proposed in this paper. Table 4 shows the comparison results of different target detection models.

Table 4. Performance comparison of 11 mainstream models.

Model	P	R	mAP50	mAP50-95	Memory (MB)	Parameters (m)	GFLOPs	Time (ms)
RT-Detr-l [29]	88.8	88.7	92.6	82.2	63.8	32.970476	108.3	20.9
RT-Detr-resnet50 [29]	91.1	85.4	91.2	83.8	83.7	42.925132	130.8	26.5
YOLOv3-tiny [24]	78.2	71.8	77	60	23.808	9.565872	14.5	4.2
YOLOv5s [25]	83.9	81.6	84.7	71.6	18.092	7.856496	19.1	4.4
YOLOv6s [26]	84.5	78.3	86.2	73.2	32.077	16.019424	43.1	8.7
YOLOv7 [23]	88.6	81.1	88.1	75.6	74.8	37.223526	105.2	20.9
YOLOv8s [16]	87.3	82.2	87.2	77.1	21.996	9.869.904	23.7	4.5
YOLOv9c [27]	89.6	85.5	91.6	81.5	50.395	21.419120	84.4	11.6
YOLOv9e [27]	91.2	87.1	94.0	83.9	114.526	54.034800	173.4	35.5
YOLOv10s [28]	86.0	79.2	86.6	77.3	16.145	8.128256	25.1	4.1
YOLO11 [16]	86.2	81.9	87.2	78.1	18.738	9.458736	21.7	4.2
our	92	86.8	93.4	83.6	12.413	6.162686	16.8	3.8

As can be seen from Table 4, the GFS-YOLO11 model proposed in this paper outperforms most of the comparison models in multiple indicators and reaches the lowest level in terms of the number of parameters, FLOPs, and inference speed. Specifically, the improved model achieved 92%, 86.6%, 93.4%, and 83.6%, respectively, in the P, R, mAP50, and MAP50-95 indexes, which were 5.8%, 4.9%, 6.2%, and 5.5% higher than those of the original YOLO11 model. At the same time, the improved model only occupies 12.413 MB, the number of parameters is 6.16 million, the computation is 16.8 GFLOPs, and the average inference time is 3.8 ms. Figure 10 directly shows the comprehensive performance comparison of different models. Although some models, such as the RT-Detr series and YOLOv9 series models, also achieved excellent results in mAP50, the GFS-YOLO11 model was superior in terms of comprehensive performance. Especially in terms of model size and computational efficiency, the advantages of the proposed method are more prominent, which is crucial for model deployment and real-time performance in practical applications.

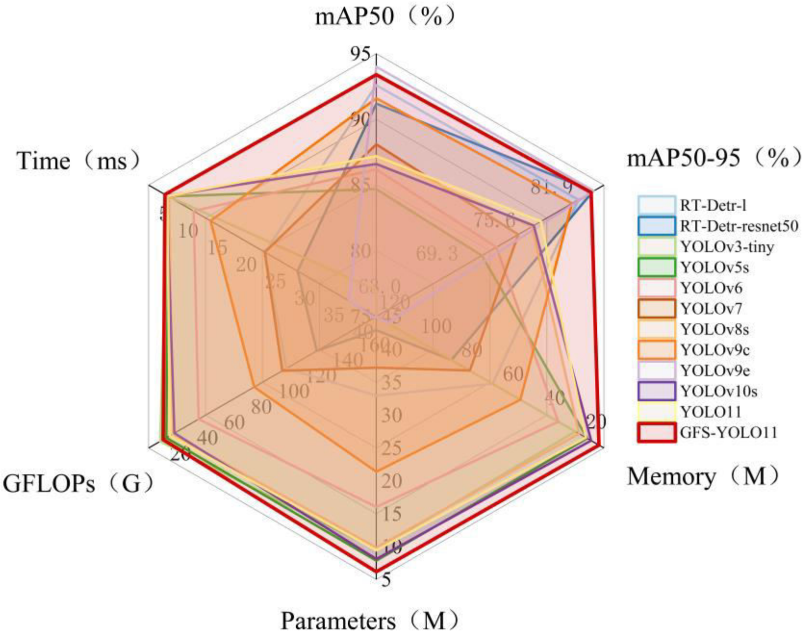

Figure 10. This figure shows a performance comparison of 12 models on multiple evaluation indicators, including the mAP50, MAP50-95, model volume, number of parameters, computational complexity, and average inference time. In the radar map, each curve represents a model, and the closer the intersection of the curve and the axis is to the edge, the better the model performs on the corresponding indicator. The larger the area enclosed by the curve, the stronger the overall performance of the model.

3.4. Visual Comparison of Test Results

In order to more intuitively demonstrate the superiority of the GFS-YOLO11 model in tomato maturity detection, we visualized and compared the results of the assay for regular and cherry tomatoes, as shown in Figures 11 and 12:

Figure 11 shows the detection results of the model on common tomatoes. It can be seen from the figure that the original YOLO11 model has some missing and false detection situations when dealing with tomatoes of different maturities. For example, the original model failed to recognize an occluded ordinary tomato and misjudged a partially ripe tomato as semi-ripe. Figure 12 shows the detection results of the model on cherry tomatoes. It can be seen that the improved model is more accurate in detecting cherry tomato maturity, and no missing detection occurred. The GFS-YOLO11 model shows higher detection accuracy and stronger robustness. The improved model can effectively identify blocked or overlapping tomatoes and can more accurately distinguish between common and cherry tomatoes. In addition, the GFS-YOLO11 model also performed well in identifying tomatoes with different maturities and was able to predict the maturity of tomatoes more accurately. These visual results show that the C3k2_Ghost module, FRM, and SPPFELAN module proposed in this paper can effectively improve the detection accuracy of the model, enhance the robustness of the model in complex scenarios, and enable it to better meet the actual demands of tomato maturity detection.

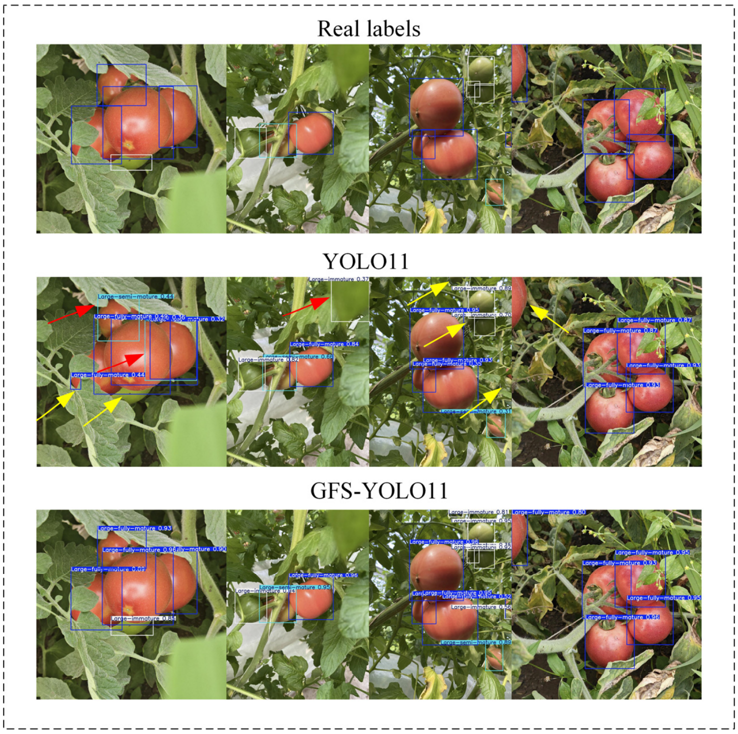

Figure 11. This figure shows the detection results of the original model and GFS-YOLO11 on common tomatoes. The first image shows the real labels, the second image shows the detection results of the original model, and the third image shows the detection results of GFS-YOLO11. The red arrows indicate false detections, and the yellow arrows indicate missed detections.

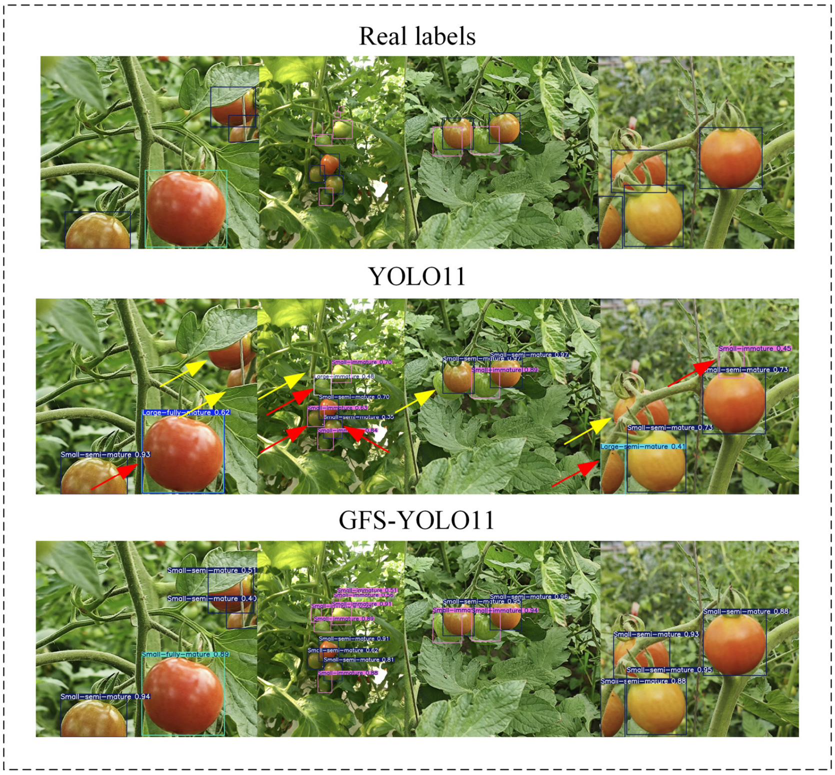

Figure 12. This figure shows the detection results of the original model and GFS-YOLO11 on cherry tomatoes. The first image shows the real labels, the second image shows the detection results of the original model, and the third image shows the detection results of GFS-YOLO11. The red arrows indicate false detections, and the yellow arrows indicate missed detections.

3.5. Ablation Experiment

The experiments in Sections 3.2–3.4 demonstrate the performance superiority of the GFS-YOLO11 model. In order to verify the effectiveness of each improvement module in the GFS-YOLO11 model, a series of ablation experiments were conducted on the Tomato-Detect dataset. In the experiment, we took the original YOLO11 model as the baseline, gradually added the C3k2_Ghost module, FRM, and SPPFELAN module and compared the performance indicators of different models. The ablation experiment aimed to study the influence of the innovative modules on the maturity detection of common tomato and cherry tomato. Table 5 shows the comparison between the model after adding the improved modules and the original model.

Table 5. Results of ablation experiments.

Number	YOLO11	C3k2_Ghost	FRM	SPPFELAN	P	R	mAP50	mAP50-95
1	√				86.2	81.9	87.2	78.1
2	√	√			84.3	78.5	85.9	77.2
3	√		√		89.1	84.2	90.9	81.4
4	√			√	90.1	85.1	90.1	80.7
5	√	√	√	√	92	86.8	93.4	83.6

As can be seen from Table 5, after the addition of the C3k2_Ghost module, all indicators of the model exhibit a slight decline, which is due to the loss of feature information caused by the lightweight processing of the model, but the C3k2_Ghost module greatly reduces the number of parameters of the model, as shown in Table 6:

Table 6. This table compares the number of parameters of the feature extraction networks (C3K2 and C3k2_Ghost) of the original model and the improved model.

	YOLO11 (C3K2)	GFS-YOLO11 (C3k2_Ghost)
Number of layers	**Params**	
2	26,080	25,088
4	103,360	99,328
6	346,112	175,840
8	1,380,352	695,744
13	443,776	263,168
16	127,680	82,432
19	345,472	164,864
22	1,511,424	826,816

This is due to the fact that the Ghost structure uses linear transformation operations to generate redundant feature maps, which significantly reduces the computational cost of the model while retaining most of the important feature information.

After the addition of the FRM, the P, R, mAP50, and MAP50-95 have improved by 2.9%, 2.3%, 3.7%, and 3.3%, respectively, compared to the benchmark model. To analyze the impact of the FRM in more depth, we visualized the feature maps for each layer of the baseline model and the improved model backbone network. Figure 13 shows the feature visualization results.

As can be seen from Figure 13, the FRM significantly enhances the clarity and richness of the feature representation because it integrates local features and global context information, extracts local detail features through depth-separable convolution, and captures feature information of different scales through multi-scale pooling. Finally, channel attention and spatial attention mechanisms are used to guide the model to focus on important feature areas. Compared to the benchmark model, the FRM model presents clearer contours, finer textures, and greater emphasis on tomato-specific features, especially when downsampled to 40 × 40.

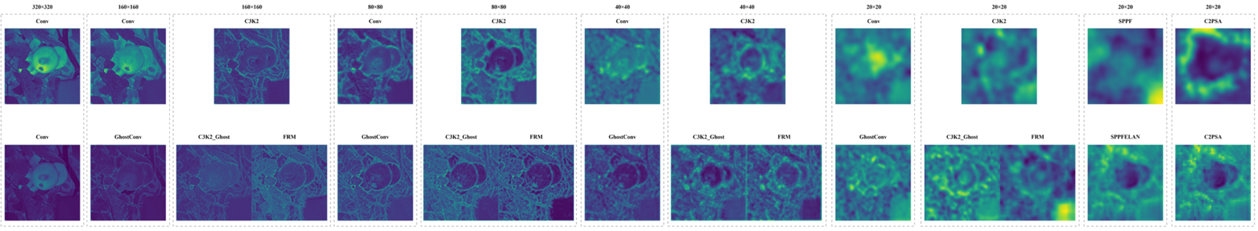

Figure 13. The first row is the feature visualizations of the YOLO11 backbone network, and the second row is the feature visualizations of the GFS-YOLO11 backbone network.

After the addition of the SPPFELAN module, the P, R, mAP50, and MAP50-95 improved by 3.9%, 3.2%, 2.9%, and 2.6%, respectively, compared to the benchmark model. In order to further analyze the influence of the SPPFELAN module, we generated a thermal map of the model on the SPPFELAN feature layer and compared it with that of the original model on the SPPF feature layer, as shown in Figure 14.

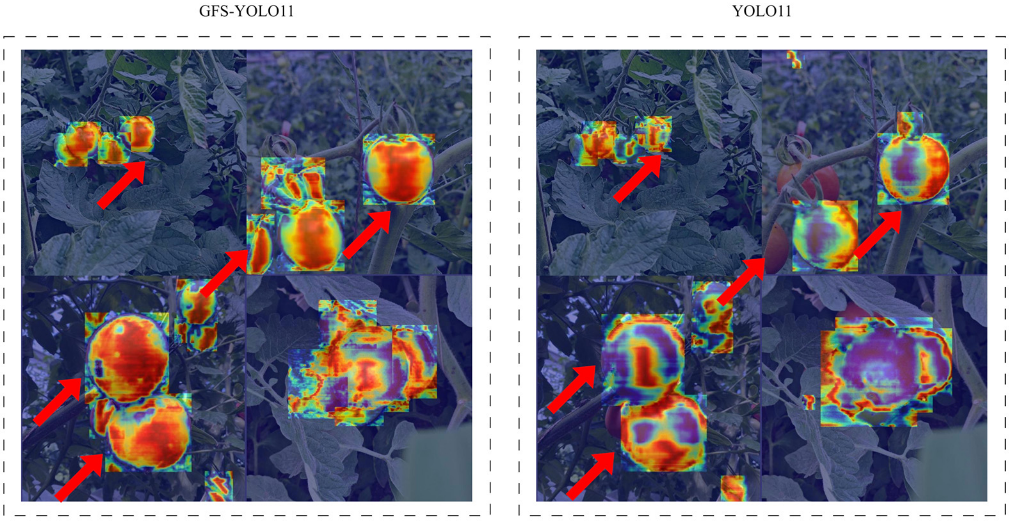

Figure 14. This figure shows the difference between the model with the SPPFELAN module and the original model in feature extraction.

Each pixel in the thermal map represents the activation degree of the corresponding position. The higher the activation value, the greater the probability of the target appearing in the position, which is reflected in the thermal map as brighter and more significant. It can be observed that after the application of the SPPFELAN module, the heat map corresponding to the generated feature map is significantly more focused on the region concerned with tomatoes, which proves the effectiveness of the SPPFELAN module in enhancing the model's feature extraction of tomatoes, enabling the model to focus on the target region more accurately, and thus improving the detection accuracy.

4. Discussion

In this study, a lightweight tomato maturity detection model GFS-YOLO11 based on improved YOLO11 was proposed, and good results were achieved on the self-built dataset. Through ablation experiments, we verified the effectiveness of the C3k2_Ghost module, FRM, and SPPFELAN module on improving the performance of the model. GFS-YOLO11 can not only accurately identify tomatoes with different maturity but also effectively distinguish between common tomatoes and cherry tomatoes, and it shows strong adaptability to complex field environments. This enables GFS-YOLO11 to be applied in various practical scenarios. For example, GFS-YOLOv8 can be integrated into field robots to achieve the automatic recognition of tomato maturity, guiding the robots for selective picking, improving the picking efficiency, and reducing labor costs.

The GFS-YOLO11 model achieves a good balance in terms of accuracy, speed, and model size. Compared with the original YOLO11 model, GFS-YOLO11 not only guarantees a high recognition accuracy but also realizes a lightweight model, reduces the computation and memory consumption, and improves the inference speed. This is due to the three improved modules proposed by us: The C3k2_Ghost module reduces the model complexity through efficient feature extraction. The FRM makes up for the loss of precision caused by its light weight by enhancing the feature expression ability. The SPPFELAN module improves the detection ability of different-size targets through multi-scale feature fusion.

Although the GFS-YOLO11 model has achieved good results, it still has some limitations: the training data of the model are mainly from specific regions and specific tomato varieties, and they may need to be fine-tuned to achieve optimal performance in the face of more different varieties and different growing environments of tomatoes. In the future, it is necessary to construct larger-scale and more diversified datasets to enhance the robustness of the model. We hope to expand the GFS-YOLO11 model to more tomato varieties and explore its application in the maturity detection of other fruits or vegetables. Additionally, we will also study how to improve the robustness of the model in complex environments such as extreme lighting and severe occlusion and explore its combination with other technologies by, for instance, combining it with robot technology to achieve automatic picking.

5. Conclusions

In this paper, a lightweight tomato ripened detection model, GFS-YOLO11, was proposed to solve the accuracy and efficiency problems faced by existing ripening detection methods when processing common tomatoes and cherry tomatoes in complex field environments. In order to achieve a lightweight model, we proposed the C3k2_Ghost module to replace the C3k2 module in the old network, which uses the GhostBottleneck structure to guarantee the ability of feature picking and reduce the amount of model computation. However, lightweight operations may lead to information loss. In order to compensate for this, we further proposed a feature-refining module (FRM) that enhances the model's ability to express the features of tomatoes of different sizes through depth-separable convolution, multi-scale pooling, and channel attention and spatial attention mechanisms, thereby improving the recognition accuracy of tomatoes at different ripening stages and varieties. Finally, to solve the problem of large size difference between ordinary tomatoes and cherry tomatoes, we proposed the SPPFELAN module. In combining the advantages of SPPF and ELANs, multiple parallel SPPF branches were used to extract features of different levels and perform splicing and fusion, further improving the detection ability of the model on ordinary tomatoes and cherry tomatoes. The experimental results on the Tomato-Detect dataset, which contains six categories, show that the proposed method achieves remarkable performance improvement. The accuracy, recall rate, mAP50, and MAP50-95 reached 92%, 86.8%, 93.4%, and 83.6%, respectively. The number of parameters, calculation amount, and reasoning speed were 12.2 MB, 6.16 M, and 3.8 ms, respectively. The experimental results show that GFS-YOLO11 can effectively reduce the parameter number and calculation amount of the model while maintaining a high recognition accuracy and can better meet the needs of real-time tomato maturity detection. In particular, the recognition accuracy of common tomatoes and cherry tomatoes is higher, the model size is smaller, and the reasoning speed is faster, showing its great potential in practical applications.

Author Contributions: Conceptualization, J.W.; methodology, J.W. and Y.S.; software, J.W. and L.L.; validation, M.C. and M.Y.; formal analysis, T.H. and Y.S.; investigation, L.N. and M.C.; resources, L.L.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, L.N. and L.L.; visualization, L.L.; supervision, Y.S. and M.Y.; funding acquisition, Y.S. and T.H.; project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Changchun Science and Technology Bureau, funding number [21ZGN27 http://kjj.changchun.gov.cn (accessed on 23 June 2024)]; the Science and Technology

Department of Jilin Province, funding number [20210302009NC http://kjt.jl.gov.cn (accessed on 23 June 2024)]; and the Department of Education of Jilin Province, funding number [JJKH20230386KJ http://jyt.jl.gov.cn (accessed on 23 June 2024)].

Data Availability Statement: All new research data were presented in this contribution.

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

References

1. Ma, M.; Taylor, P.W.J.; Chen, D.; Vaghefi, N.; He, J.-Z. Major Soilborne Pathogens of Field Processing Tomatoes and Management Strategies. *Microorganisms* **2023**, *11*, 263. [CrossRef] [PubMed]
2. El-Ramady, H.R.; Domokos-Szabolcsy, É.; Abdalla, N.A.; Taha, H.S.; Fári, M. Postharvest Management of Fruits and Vegetables Storage. In *Sustainable Agriculture Reviews: Volume 15*; Lichtfouse, E., Ed.; Springer International Publishing: Cham, Swizerland, 2015; pp. 65–152. ISBN 978-3-319-09132-7.
3. Mao, L. How to Break and Establish the High Loss of China's Fresh Agricultural Products with an Annual Loss of 300 Billion. Agricultural Industry Observation. 2022. Available online: https://www.163.com/dy/article/G1DQSJNA05118U1Q.html (accessed on 20 October 2024).
4. Azadnia, R.; Kheiralipour, K. Evaluation of Hawthorns Maturity Level by Developing an Automated Machine Learning-Based Algorithm. *Ecol. Inform.* **2022**, *71*, 101804. [CrossRef]
5. Kurtulmus, F.; Lee, W.S.; Vardar, A. Green Citrus Detection Using 'Eigenfruit', Color and Circular Gabor Texture Features under Natural Outdoor Conditions. *Comput. Electron. Agric.* **2011**, *78*, 140–149. [CrossRef]
6. Santos Pereira, L.F.; Barbon, S.; Valous, N.A.; Barbin, D.F. Predicting the Ripening of Papaya Fruit with Digital Imaging and Random Forests. *Comput. Electron. Agric.* **2018**, *145*, 76–82. [CrossRef]
7. Zhu, X.; Chen, F.; Zheng, Y.; Chen, C.; Peng, X. Detection of Camellia Oleifera Fruit Maturity in Orchards Based on Modified Lightweight YOLO. *Comput. Electron. Agric.* **2024**, *226*, 109471. [CrossRef]
8. Wang, C.; Wang, C.; Wang, L.; Wang, J.; Liao, J.; Li, Y.; Lan, Y. A Lightweight Cherry Tomato Maturity Real-Time Detection Algorithm Based on Improved YOLOV5n. *Agronomy* **2023**, *13*, 2106. [CrossRef]
9. Wang, C.; Han, Q.; Li, J.; Li, C.; Zou, X. YOLO-BLBE: A Novel Model for Identifying Blueberry Fruits with Different Maturities Using the I-MSRCR Method. *Agronomy* **2024**, *14*, 658. [CrossRef]
10. Xu, D.; Ren, R.; Zhao, H.; Zhang, S. Intelligent Detection of Muskmelon Ripeness in Greenhouse Environment Based on YOLO-RFEW. *Agronomy* **2024**, *14*, 1091. [CrossRef]
11. Sun, X. Enhanced Tomato Detection in Greenhouse Environments: A Lightweight Model Based on S-YOLO with High Accuracy. *Front. Plant Sci.* **2024**, *15*, 1451018. [CrossRef]
12. Li, P.; Zheng, J.; Li, P.; Long, H.; Li, M.; Gao, L. Tomato Maturity Detection and Counting Model Based on MHSA-YOLOv8. *Sensors* **2023**, *23*, 6701. [CrossRef]
13. Li, R.; Ji, Z.; Hu, S.; Huang, X.; Yang, J.; Li, W. Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy* **2023**, *13*, 603. [CrossRef]
14. Gongal, A.; Amatya, S.; Karkee, M.; Zhang, Q.; Lewis, K. Sensors and Systems for Fruit Detection and Localization: A Review. *Comput. Electron. Agric.* **2015**, *116*, 8–19. [CrossRef]
15. Arnal Barbedo, J.G. Plant Disease Identification from Individual Lesions and Spots Using Deep Learning. *Biosyst. Eng.* **2019**, *180*, 96–107. [CrossRef]
16. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 20 October 2024).
17. Du, Y.; Han, Y.; Su, Y.; Wang, J.H. A Lightweight Model Based on You Only Look Once for Pomegranate before Fruit Thinning in Complex Environment. *Eng. Appl. Artif. Intell.* **2024**, *137*, 109123. [CrossRef]
18. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
19. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 13–19 June 2020; pp. 1571–1580.
20. Liu, Z.; Xiong, J.; Cai, M.; Li, X.; Tan, X. V-YOLO: A Lightweight and Efficient Detection Model for Guava in Complex Orchard Environments. *Agronomy* **2024**, *14*, 1988. [CrossRef]
21. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
22. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef]
23. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
24. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.

25. Jocher, G. YOLOv5 by Ultralytics. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 October 2024).
26. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
27. Wang, C.-Y.; Yeh, I.-H.; Liao, H. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
28. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
29. Lv, W.; Xu, S.; Zhao, Y.; Wang, G.; Wei, J.; Cui, C.; Du, Y.; Dang, Q.; Liu, Y. DETRs Beat YOLOs on Real-Time Object Detection. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.

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.