agronomy

MDPI

Article

Intelligent Detection of Muskmelon Ripeness in Greenhouse Environment Based on YOLO-RFEW

Defang Xu [1,*], Rui Ren [2], Huamin Zhao [2,*] and Shujuan Zhang [2]

[1] Department of Mathematics and Artificial Intelligence, Lvliang University, Xueyuan Road, Lishi District, Lvliang 033001, China
[2] College of Agricultural Engineering, Shanxi Agricultural University, Jinzhong 030801, China; b20221003@stu.sxau.edu.cn (R.R.); zsj2021@sxau.edu.cn (S.Z.)
* Correspondence: 20211018@llu.edu.cn (D.X.); zhaohuamin@sxau.edu.cn (H.Z.)

Abstract: Accurate detection of muskmelon fruit ripeness is crucial to ensure fruit quality, optimize picking time, and enhance economic benefits. This study proposes an improved lightweight YOLO-RFEW model based on YOLOv8n, aiming to address the challenges of low efficiency in muskmelon fruit ripeness detection and the complexity of deploying a target detection model to a muskmelon picking robot. Firstly, the RFAConv replaces the Conv in the backbone part of YOLOv8n, allowing the network to focus more on regions with significant contributions in feature extraction. Secondly, the feature extraction and fusion capability are enhanced by improving the C2f module into a C2f-FE module based on FasterNet and an Efficient Multi-Scale attention (EMA) mechanism within the lightweight model. Finally, Weighted Intersection over Union (WIoU) is optimized as the loss function to improve target frame prediction capability and enhance target detection accuracy. The experimental results demonstrate that the YOLO-RFEW model achieves high accuracy, with precision, recall, F1 score, and mean Average Precision (mAP) values of 93.16%, 83.22%, 87.91%, and 90.82%, respectively. Moreover, it maintains a lightweight design and high efficiency with a model size of 4.75 MB and an inference time of 1.5 ms. Additionally, in the two types of maturity tests (M-u and M-r), APs of 87.70% and 93.94% are obtained, respectively, by the YOLO-RFEW model. Compared to YOLOv8n, significant improvements in detection accuracy have been achieved while reducing both model size and computational complexity using the proposed approach for muskmelon picking robots' real-time detection requirements. Furthermore, when compared to lightweight models such as YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, YOLOv8s, and YOLOv8n, the YOLO-RFEW model demonstrates superior performance with only 28.55%, 22.42%, 24.50%, 40.56%, 22.12%, and 79.83% of their respective model sizes, respectively, while achieving the highest F1 score and mAP values among these seven models. The feasibility and effectiveness of our improved scheme are verified through comparisons between thermograms generated by YOLOv8n and YOLO-RFEW as well as detection images. In summary, the YOLO-RFEW model not only improves the accuracy rate of muskmelon ripeness detection but also successfully realizes the lightweight and efficient performance, which has important theoretical support and application value in the field of muskmelon picking robot development.

Keywords: greenhouse environment; muskmelon fruit; YOLO-RFEW; detection

Citation: Xu, D.; Ren, R.; Zhao, H.; Zhang, S. Intelligent Detection of Muskmelon Ripeness in Greenhouse Environment Based on YOLO-RFEW. *Agronomy* **2024**, *14*, 1091. https://doi.org/10.3390/agronomy14061091

Academic Editor: Juncheng Ma

Received: 22 April 2024
Revised: 17 May 2024
Accepted: 20 May 2024
Published: 21 May 2024

1. Introduction

With the acceleration of agricultural modernization and the continuous improvement of consumers' demand for agricultural products, the accurate detection of muskmelon ripeness has become a key issue to be solved in the agricultural production process. The unevenness of flowering and ripening of muskmelon makes accurate detection of its ripeness essential to guarantee the quality of the fruit, optimize the picking time, and improve the economic benefits. However, the detection of muskmelon ripeness is more complex than

that of other fruits, as its appearance and color changes may not be obvious at different stages of ripening and are affected by a variety of factors such as the greenhouse environment, light conditions, and fruit overlap [1–3]. This makes the traditional detection methods relying on manual visual judgment and experience accumulation not only inefficient but also highly susceptible to the influence of subjective factors, which makes it difficult to ensure the consistency and accuracy of the detection results. Therefore, the development of an efficient and accurate automatic muskmelon ripeness detection model is of great significance to enhance the intelligence level of the muskmelon industry and market competitiveness.

In recent years, researchers both domestically and internationally have made remarkable strides in leveraging deep learning techniques coupled with computer vision technology for agricultural applications [4–6]. You Only Look Once (YOLO) has garnered considerable attention as an advanced real-time target detection algorithm owing to its exceptional efficiency, speed, and accuracy [7–9]. Solimani et al. [10] proposed a detection method improved by data balancing based on the YOLOv8 deep learning model to address the effect of unbalanced sample distribution by considering unbalanced categories of flowers, fruits, and nodes of tomato plants. The results show that using the attention mechanism and pre-trained weights obtained from the balanced dataset to evaluate the unbalanced data results in a maximum mAP50 of 65.77% for YOLOv8m, improving the overall detection performance. Chen et al. [11] proposed the MTD-YOLOv7 detection model for multi-task detection of cherry tomato bunches, achieving recognition of cherry tomato bunches as well as fruit and bunch maturity detection. The mAP of MTD-YOLOv7 improved by 0.5%, 1.4%, and 0.5% in the classification-only, fruit maturity-only, and bunch maturity-only tasks, respectively, resulting in an overall performance improvement of 0.8% compared to the original model. Edy et al. [12] employed a computer vision approach that integrated the YOLOv4-Tiny model with genetic algorithm to detect oil palm maturity using a curated dataset optimized based on genetic algorithm, leading to enhancements of 0.1% and 0.5% in the model's mAP. Kazama et al. [13] employed the enhanced YOLOv8 architecture and Receptive-Field Attention coordinate attention Conv module to accurately detect coffee fruits and classify their ripening stages. The model achieved an mAP of 74.20%, with individual AP of 73.40%, 67.10%, and 71.90% for unripe, semi-ripe, and overripe fruits, respectively. Xiong et al. [14] proposed the YOLOv5-Lite model, which demonstrates visual recognition of papaya ripeness on trees in natural environments and enables monitoring of ripeness changes during papaya growth. The model achieves an mAP of 92.4% for papaya detection, completes the detection process within a mere 7 ms, and occupies only 11.3 MB of memory space. Chen et al. [15] proposed the Des-YOLOv4 algorithm and an apple detection method for enabling a harvesting robot to accurately and quickly detect apples in complex environments. The results demonstrate that the Des-YOLOv4 network possesses desirable features, with an apple detection mAP of 97.13%, recall of 90%, and a detection speed of 51 f/s. The improved model can meet both precision and speed requirements for apple detection. Ren et al. [16] introduced a lightweight YOLO-GEW detection method to achieve fast and accurate identification of "Yuluxiang" pear fruits in unstructured environments. Experimental outcomes indicate that the F1 and AP of YOLO-GEW are 84.47% and 88.83%, respectively, while its model size is only 65.50% compared to that of YOLOv8s, indicating efficient real-time identification capability under unstructured environmental conditions. Li et al. [17] proposed an enhanced lightweight algorithm based on YOLOv5 to address the challenges of low detection efficiency in field flat jujube and the complexity of target detection algorithms that are difficult to deploy on inexpensive devices. The improved model L-YOLOv5s-RCA achieved an mAP of 97.2% and had a model size of 7.1 MB, providing valuable insights for implementing automated picking of flat dates in the field. Huang et al. [18] introduced Mobile-YOLOv5s, a lightweight algorithm for multi-stage identification and detection of strawberries based on an optimized version of YOLOv5s. It achieves a detection frame rate of 44 frames/s with a model size of 4.5 MB, demonstrating a detection accuracy of 99.5% for ripe strawberries and a mean average

accuracy of 99.4%. This improved model enables faster and more accurate identification at various stages, offering technical support for intelligent strawberry harvesting operations. Despite the significant advancements in deep learning technology for agricultural applications, the research on muskmelon ripeness detection still encounters several challenges. Existing deep learning-based detection methods often suffer from high model complexity and computational demands, which hinder real-time and accurate detection on low-cost devices. Therefore, it is crucial to develop an efficient, precise, and lightweight automatic model for muskmelon ripeness detection.

In this study, aiming to overcome the limitations of existing muskmelon detection methods, a YOLO-RFEW-based model for intelligent detection of muskmelon ripeness in greenhouse environments is proposed. The model is based on the advanced YOLOv8n and optimizes the feature extraction capability of the main part of the model by introducing RFAConv to better capture the subtle feature changes of muskmelons; designs the lightweight module C2f-FE to enhance the feature fusion effect and improve the model's ability to recognize muskmelons at different ripening stages; and employs the Weighted Intersection over Union (WIoU) loss function to improve the performance of bounding box regression to ensure more accurate localization. These improvements enable YOLO-RFEW to significantly enhance the detection accuracy of muskmelon ripeness while maintaining a low computational complexity. Compared to previous studies on other fruits, our research delves into unique challenges specific to muskmelon ripeness detection and analyzes various considerations in model design comprehensively. By providing effective technical support for practical deployment of melon-picking robots in greenhouse environments, this study is expected to drive intelligent development within the muskmelon industry.

2. Materials and Methods

2.1. Dataset Information

The dataset of muskmelon fruits with varying maturity levels was collected from a greenhouse dedicated to muskmelon cultivation in Taigu District, Shanxi Province, China, between 8:00 and 18:00. High-resolution images were captured using an EVR-AL00 camera with a resolution of 3648 $\times$ 2736 pixels. Sample images were taken under diverse lighting conditions including smooth light, backlighting, near distance, and far distance to enhance the diversity of the dataset. A total of 910 test sample images were collected and resized to clear image samples measuring 640 $\times$ 640 pixels. Following the established practice in the field of deep learning, this study employs a random division strategy to allocate the dataset into three subsets: a training set (637 frames), a validation set (182 frames), and a test set (91 frames) with a ratio of 7:2:1. This partitioning scheme ensures sufficient training for the model while providing ample data for rigorous evaluation of its performance and generalization ability. The training set was utilized for model training, while the validation set served for hyperparameter tuning and initial assessment of model capability. Finally, the test set was employed to evaluate detection accuracy and generalization capability.

The images of different muskmelon maturity levels in the dataset were accurately labeled using LabelImg software (1.8.6). A maximum horizontal rectangular box was used to frame the muskmelon fruit region in the images, and an XML file in VOC format was used for annotation. Each image contains at least one muskmelon fruit for model training. Additionally, the XML file in VOC format is converted into a TXT file in YOLO format to optimize the model training process further. To prevent overfitting and poor generalization caused by a small dataset, noise was added and brightness and darkness were adjusted on the original images to enhance dataset quality. Finally, a training set consisting of 2548 images, a validation set of 728 images, and a test set of 364 images were obtained. In this study, unripe fruits are green and white while ripe fruits are yellow. The muskmelons labeled as targets for detection are divided into two categories: pickable ripe muskmelons (muskmelon_ripe) abbreviated as M-r and unripe muskmelons (muskmelon_unripe) abbreviated as M-u. The dataset containing muskmelons with different ripeness levels is shown in Figure 1.

Figure 1. Dataset of muskmelon fruits at different maturity levels.

2.2. Constructing the YOLO-RFEW Muskmelon Maturity Detection Model

The network architecture of YOLOv8 is based on a deep convolutional neural network (DCNN), which consists of three components: backbone, neck, and head. Among them, the backbone uses CSPDarknet53 to extract features from the input image, and CSPDarknet53 employs CSP (Cross-Stage Partial) connectivity, which introduces cross-stage partial connectivity to realize information exchange among network modules. This design not only improves the stability and generalization ability of the model but also enhances the nonlinear representation of the network, thus effectively improving the accuracy. The neck fuses the features extracted by the backbone to improve multi-scale target detection performance, and the head is responsible for generating final detection results, including target location, category, and confidence level. To meet different usage scenarios and performance requirements, YOLOv8 provides five backbone network models with different scales: n, s, m, l, and x. These models differ in terms of parameters count, computation complexity, detection speed, and accuracy [19–21]. Among them, YOLOv8n has the smallest number of parameters and fastest detection speed. Therefore, in order to ensure real-time performance and control model size, this study chose to use the YOLOv8n version.

The YOLO-RFEW model is proposed in this study, as illustrated in Figure 2, to further enhance the detection accuracy of the YOLOv8n model for different ripeness levels of muskmelon while reducing its complexity. Firstly, RFAConv is employed in the backbone section to replace the underlying Conv and strengthen the feature extraction capability of the model. Secondly, a lightweight C2f-F structure is devised to substitute both C2f in the backbone section and neck part, aiming at improving detection performance. Moreover, an EMA attention mechanism called C2f-FE is introduced on top of C2f-F to further enhance feature extraction and fusion capabilities by incorporating weighting during the feature extraction process for obtaining more informative data and achieving fusion between shallow and deep feature maps. Consequently, richer semantic information is obtained, leading to improved model performance. Finally, the WIoU loss function is incorporated into the IoU calculation by integrating categorization information for enhancing bounding box regression performance and detection accuracy.

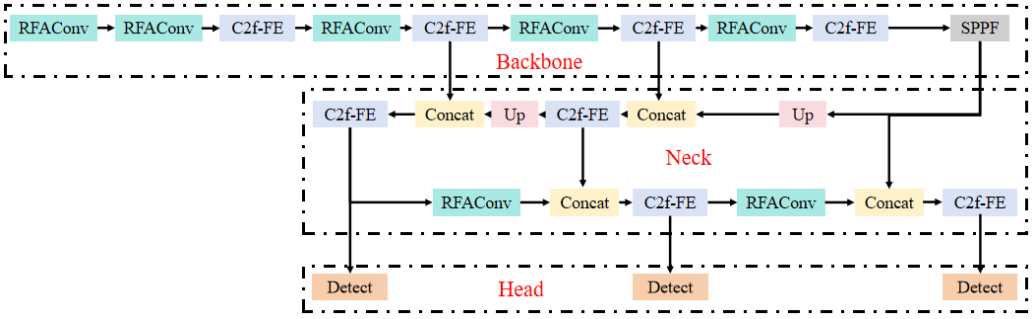

Figure 2. Structure of the YOLO-RFEW model.

2.3. RFAConv Module

The RFAConv (Receptive-Field Attention Convolution) module presents a novel approach to convolutional operations, which effectively enhances the performance of CNNs while incurring minimal computational cost and parameter increase through the incorporation of the RFA attention mechanism [22]. The RFA (Receptive-Field Attention) attention

mechanism aims to optimize model performance by augmenting the adaptability of the receptive field within the convolutional neural network. Initially, spatial transformation is applied to the input feature map to acquire relative positional information for each pixel location across various directions, facilitating comprehensive capture of crucial features within the spatial structure. Subsequently, this positional information is utilized to compute attention weights for each pixel location and adjust network focus towards regions that contribute more significantly to feature extraction based on their importance. Ultimately, these calculated attention weights are multiplied with the original feature map, yielding a weighted feature map.

RFAConv is a convolutional algorithm designed based on the RFA attention mechanism, which utilizes the RFA-weighted feature maps as both outputs and inputs to the next layer of convolutional operations. This enables the network to focus more on regions that significantly contribute to feature extraction, thereby enhancing the model's accuracy and robustness.

2.4. Construction of the C2f-FE Module

2.4.1. Construction of the C2f-F Module

The FasterNet Block is specifically designed to address the challenges of high computational and memory requirements in conventional convolutional neural networks, while simultaneously enhancing network performance. Comprising two key components, namely Partial Convolution (PConv) and Pointwise Convolution (PWConv), the FasterNet Block optimizes computation by selectively applying the convolution kernel to a subset of input feature map channels, thereby reducing redundancy and improving efficiency [23]. Additionally, PWConv employs a point-by-point convolution operation that focuses on the central region of the feature map, augmenting the network's ability to capture local details.

The sequential application of PConv and PWConv in a FasterNet Block facilitates the comprehensive utilization of information across all channels while preserving sensitivity to intricate details. Within each FasterNet Block, a PConv operation is employed, succeeded by two PWConv operations. This architectural design effectively harnesses information from all channels, forming a T-shaped convolutional structure that enables the model to prioritize features at the central location. Normalization and activation layers are exclusively added after the intermediate layers to uphold feature diversity and minimize computational latency. The configuration of the FasterNet Block is depicted in Figure 3.

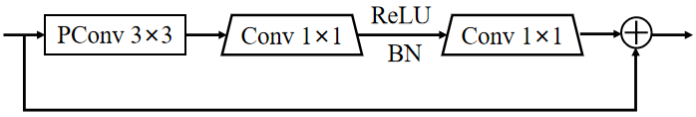

Figure 3. FasterNet Block structure.

The C2f-F module is introduced in YOLOv8n, replacing the Bottleneck in C2f with FasterNet Block, thereby achieving a reduction in model size and faster running speed while maintaining model performance. This innovation further enhances the feature extraction and features fusion capabilities of the model.

2.4.2. EMA Attention Mechanism

The attention mechanism helps the model to better focus on important features in the image and ignore irrelevant background information, thus improving the accuracy of detection. Through the attention mechanism, the model is able to explain its decision-making process, i.e., the image regions it focuses on when making predictions, thus enhancing the interpretability of the model [24–26]. EMA (Efficient Multi-Scale Attention) is an efficient multi-scale attention module, whose core idea is to achieve efficient learning across channels without increasing the model's complexity by reshaping some of the channels and grouping them together [27], i.e., complexity without increasing the model complexity

to achieve efficient learning across channels. The EMA attention mechanism is shown in Figure 4.

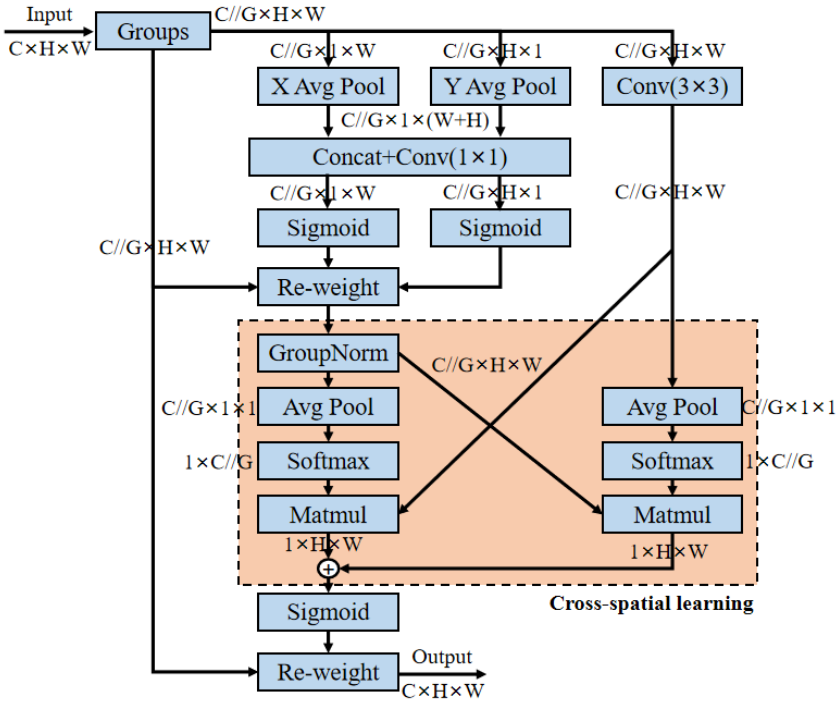

Figure 4. EMA module. "X Avg Pool" represents the 1D horizontal global pooling, and "Y Avg Pool" indicates the 1D vertical global pooling. C means the numbers of the input channels, H and W indicate the spatial dimensions of the input features, respectively. G represents the number of sub-features. Matmul stands for matrix multiplication. GroupNorm stands for normalization. Re-weight stands for re-weighting. Groups stands for grouped convolution. Sigmoid and Softmax are activation functions. The input tensor shape is defined as $C//G \times H \times W$.

The C2f-FE module, based on C2f-F, introduces the EMA attention mechanism to maintain high performance and real-time capability of the model. EMA assists the model in focusing better on important features in the image and enhances its feature extraction and fusion capabilities. The C2f-FE module is shown in Figure 5.

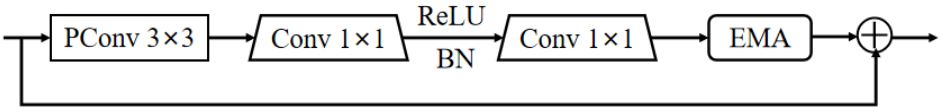

Figure 5. C2f-FE module.

2.5. WIoU Loss Function

While the traditional IoU loss function primarily focuses on the overlap area between the predicted and real frames in target detection tasks, it overlooks crucial information regarding the size and shape of the target frame. To address this limitation, this study introduces the WIoU loss function, which dynamically adjusts weighting coefficients to better accommodate targets with varying sizes and shapes. The calculation of these coefficients constitutes a pivotal aspect of the WIoU loss function. In practical applications, first compute shape and size features such as aspect ratio difference and area difference between the predicted and real frames, enabling us to effectively capture similarities or differences in their respective sizes and shapes [28].

The weight adjustment function dynamically generates a weight coefficient for each prediction frame based on the aforementioned shape and size features. This function is specifically designed to enhance the model's focus on target frames exhibiting significant

disparities in size or shape, thereby enabling targeted optimization during the training process. In calculating the WIoU loss function, this study incorporates both intersection area and concatenation area between predicted and real frames to compute IoU values. Subsequently, these IoU values are weighted using dynamically adjusted coefficients to derive final WIoU loss values. YOLOv8n leverages the optimized WIoU loss function to enhance its predictive capability for target frames as well as improve target detection accuracy.

2.6. Grad-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique for visualizing the decision-making process of deep neural networks, especially for convolutional neural networks (CNNs). It analyzes the gradient information of the last convolutional layer in a CNN to reveal which image regions contribute the most when the model makes a prediction.

The core idea of Grad-CAM is to obtain importance weights for each feature map by calculating the gradient of a particular category relative to the feature map of the convolutional layer. These weights reflect how much the feature maps contribute to the final category prediction. Subsequently, the weights are weighted and overlaid with the feature maps to generate a Class Activation Map (CAM) that visualizes the key regions that the model focuses on during prediction.

The advantage of Grad-CAM is its versatility and flexibility. It can be applied to a variety of CNN structures and tasks without modifying the network structure or retraining the model. By combining Grad-CAM with YOLOv8n, it is possible to visualize the regions of interest of the model when detecting different targets, to better understand how the model locates and identifies targets in complex scenes, which not only helps to improve the performance of the model but also provides a strong support for subsequent model optimization and improvement.

2.7. Test Platforms

The research used the Windows 10 operating system, Intel Core i7-13790F CPU, 32 GB RAM, 2 T hard drive, and 24,564.0 MB NVIDIA GeForce RTX 4090 GPU. The running software mainly includes python 3.8.8, torch 1.13.1, and CUDA 11.6. All programming was conducted in pycharm 2023.

The network input image size was 640 × 640 × 3 pixels, Stochastic Gradient Descent Optimization (SGD) was used as the parameters were updated using a stochastic gradient descent optimizer with a momentum of 0.937, and the initial learning rate is set to 0.01. The batch size is 16, and 200 epochs are trained. All models use the same dataset and training parameters.

2.8. Evaluation Indicators

In order to accurately assess the performance of the model, the evaluation metrics used in this study are Precision, Recall, F1, Accuracy (AP), and Mean Average Precision (mAP). In calculating the mAP, the IoU threshold of 0.5 was chosen in this study. This is because in the study, for the task of detecting the ripeness of muskmelon in a greenhouse environment, the IoU threshold of 0.5 can better reflect the accuracy of the model in localizing the bounding box of the muskmelon. By calculating the mAP when the IoU threshold is 0.5, the performance of the model in detecting muskmelon ripeness can be effectively evaluated. Therefore, in this study, the mAP represented is the mAP of muskmelon fruits under the IoU threshold of 0.5. The specific formula is as follows:

$$AP = \int_0^1 P\text{recision}(\text{Recall})\text{d}(\text{Recall}) \tag{1}$$

$$F1 = \frac{2\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

$$\mathrm{m}AP = \frac{1}{k}\sum_{i=1}^{k} AP_i \tag{3}$$

In addition, the running time of the model is evaluated using the inference time, the computational complexity of the model is evaluated using the number of floating-point computations (FLOPs), the number of parameters, and the model size evaluates the size of the model. In this study, M-r and M-u need to be detected; therefore, the number of detection categories k is set to 2.

3. Results and Analysis

3.1. Effect of Different Convolutions on Model Feature Extraction

The backbone component of the YOLOv8n model was experimentally examined in this study to enhance its feature extraction capability. All convolutional layers were substituted with RFAConv, Depthwise Convolution (DWConv), KernelWarehouse (KWConv), and Group Shuffle Convolution (GSConv). The results are shown in Table 1.

Table 1. Results of different convolution tests.

Model	Precision (%)	Recall (%)	F1 (%)	mAP (%)	Model Size (MB)	Inference Time (ms)
Conv	86.44	80.64	83.44	88.31	5.95	1.7
RFAConv	93.53	78.27	85.22	89.23	6.06	1.7
DWConv	90.33	82.68	86.34	87.76	5.22	1.5
KWConv	94.34	80.68	86.98	88.94	6.00	2.0
GSConv	91.59	76.24	83.21	87.04	5.59	1.7

The experimental results presented in Table 1 demonstrate that RFAConv, DWConv, KWConv, and GSConv exhibit precision improvements of 7.09%, 3.89%, 7.9%, and 5.15%, respectively, when compared to the original convolutional model. However, with respect to recall performance, DWConv achieves the most significant enhancement of 82.68%, while RFAConv and GSConv experience decreases of 2.37% and 4.4%, respectively, relative to the baseline model's performance. In terms of F1, GSConv exhibits a marginal decrease of only 0.23% compared to the original model; conversely, RFAConv, DWConv, and KWconv all demonstrate notable improvements of 1.78%, 2.9%, and 3.54%, respectively. Compared to Conv, DWConv and GSConv resulted in a reduction in model size by 0.73 MB and 0.36 MB, respectively. However, the mAP metrics experienced a decrease of 0.55% and 1.27%, respectively. Conversely, RFAConv and KWConv led to an increase in model size by 0.11 MB and 0.05 MB compared to Conv but demonstrated improvements in the mAP metrics by 0.92% and 0.63%, respectively. In terms of inference time, compared to the original Conv, RFAConv and GSConv remain essentially the same, with DWConv speeding up by 0.2 ms, and KWConv slowing down the inference time of the model by 0.3 ms due to the complexity of the internal convolution. Taken together, among the five different types of convolutions, RFAConv exhibited the highest mAP value of 89.23%, showcasing significant enhancements in both accuracy and F1 value over the original model. Therefore, for this study we selected RFAConv as the backbone component of the YOLOv8n model named YOLO-R, which effectively enhances feature extraction with minimal computational cost.

3.2. Effect of C2f-F Combined with the Attention Mechanism on Model Performance

The C2f-F structure is employed for the backbone and neck components of the model to achieve lightweightness and enhance its detection performance for muskmelons with varying maturity levels. Additionally, an attention mechanism is incorporated into the C2f-F structure to optimize feature extraction and fusion effects. To validate the efficacy of EMA on model detection performance, this study compares Efficient Local Attention (ELA) [29], Simple Parameter-Free Attention Module (SimAM) [30], Squeeze and Excitation (SE) [31], and EMA—four attention modules under identical experimental conditions

in terms of their impact on the performance of the lightweighted model. The effects of different attention mechanisms on the performance of the lightweighted model are shown in Table 2.

Table 2. Effect of different attention mechanisms on the performance of lightweight models.

Model	Precision (%)	Recall (%)	F1 (%)	mAP (%)	Model Size (MB)	Parameters (M)	Inference Time (ms)
YOLOv8n	86.44	80.64	83.44	88.31	5.95	3.011	1.7
C2f-F	93.26	80.33	86.31	88.89	4.61	2.306	1.4
EMA	90.78	82.31	86.34	89.96	4.66	2.315	1.4
ELA	93.32	79.91	86.10	88.55	4.73	2.360	1.5
SimAM	92.81	81.04	86.53	89.10	4.61	2.306	1.4
SE	88.69	85.05	86.83	88.95	4.63	2.313	1.4

As presented in Table 2, the C2f-F structure significantly enhances the detection accuracy of the YOLOv8n-based model. Specifically, there is a notable improvement of 6.82% in precision, 2.87% in F1, and 0.58% in mAP; however, recall experiences a slight decrease of 0.31%. Moreover, C2f-F achieves effective model lightweighting by reducing the model size by 1.34 MB and parameters by 0.705 M compared to the original model, resulting in an accelerated inference time of only 0.3 ms. This demonstrates that the incorporation of C2f-F structure positively impacts both lightweighting and detection performance across various ripeness levels for muskmelon models. Amongst the four attention mechanisms evaluated, ELA exhibits superior precision at 93.32%, while its recall, F1, and mAP are relatively lower at values of 79.91%, 86.10%, and 88.55%, respectively. Furthermore, the model size and parameters of ELA are also larger than C2f-F by 1.22 MB and 0.054 M, respectively, while the inference time becomes slower by 0.1 ms. Therefore, ELA exhibits poor performance in terms of model detection effect. As SimAM employs a parameter-free attention mechanism, its model size, parameters, and inference time are essentially consistent with those of C2f-F. In terms of accuracy, recall, F1, and mAP, there is an improvement of 0.4%, 0.22%, and 0.21%, respectively, compared to C2f-F's performance. On the other hand, SE shows a smaller enhancement in model performance compared to C2f-F as its mAP only improves by 0.06% while increasing the model size by 0.02 MB.

Compared to C2f-F, EMA demonstrates a 1.07% increase in mAP while maintaining F1 and inference time at essentially unchanged levels. Moreover, the augmentation in model size and parameters is merely 0.05 MB and 0.009 M, respectively. In contrast to ELA, SimAM, and SE, EMA maintains comparable model size and parameters but surpasses them in terms of mAP with enhancements of 1.41%, 0.86%, and 1.01%, correspondingly. In addition, the enhanced model also demonstrated notable achievements in lightweighting. Specifically, the size and parameters of the C2f-FE module were reduced by 1.29 MB and 0.696 M, respectively, while exhibiting an inference time that was 0.3 ms faster than that of YOLOv8n. Considering its exceptional lightweighting capabilities, feature extraction proficiency, and feature fusion aptitude, this study opted to employ the C2f-FE module in this study as a replacement for the original C2f module within YOLOv8n architecture, denoting it as YOLO-FE for extracting features at varying stages of muskmelon maturity. This model possesses the ability to prioritize valuable information during feature extraction and effectively fuse shallow and deep feature maps to acquire more comprehensive semantic information, thereby enhancing performance, thus laying a solid foundation for subsequent research endeavors.

3.3. Ablation Test

The ablation test serves to validate the efficacy of the enhancements made to the YOLOv8n model by comparing its performance and post-improvement, thereby facilitating a comprehensive understanding of its underlying mechanisms [32]. Furthermore, this test enhances interpretability by analyzing the impact of different improvements on model per-

formance, with corresponding results presented in Table 3. Specifically, YOLO-R combined with YOLO-FE is denoted as YOLO-RFE; models enhanced with the WIoU loss function are referred to as YOLO-W, YOLO-FEW, and YOLO-RFEW.

Table 3. Ablation test results.

Model	F1 (%)		AP (%)		mAP (%)	Model Size (MB)
	M-u	M-r	M-u	M-r		
YOLOv8n	80.20	86.26	82.92	93.71	88.31	5.95
YOLO-R	80.41	89.64	84.52	93.94	89.23	6.06
YOLO-FE	82.98	89.52	85.71	94.22	89.96	4.66
YOLO-W	80.27	88.78	82.97	94.43	88.70	5.95
YOLO-RFE	84.83	88.91	87.36	93.18	90.27	4.75
YOLO-FEW	84.36	92.06	86.17	94.09	90.13	4.66
YOLO-RFEW	84.71	91.06	87.70	93.94	90.82	4.75

The improved models YOLO-R, YOLO-FE, YOLO-W, YOLO-RFE, YOLO-FEW, and YOLO-FEW, which demonstrated enhanced mAP compared to the baseline model YOLOv8n, are shown in Table 3. Specifically, the improvements were observed as follows: 0.92%, 1.65%, 0.39%, 1.96%, 1.83%, and 2.51%, respectively, for each model variant. These results indicate that each of the proposed enhancements effectively contributes to accuracy improvement. Among these variants, the optimized C2f-FE module in the improved model based on YOLO-R, YOLO-RFE, successfully reduced the model size by approximately 1.31 MB while achieving significant performance enhancement in M-u detection with an increase in F1 value and AP by 4.42% and 2.84%, respectively; however, a slight decrease was observed in M-r detection with a decline in F1 value and AP by 0.73% and 0.76%, respectively. Overall, mAP increased by approximately 1.04%. In contrast, the enhanced YOLO-RFEW model, built upon YOLO-FE, exhibits a mere 0.09 MB increase in model size while achieving a notable 0.31% improvement in mAP by incorporating the RFAConv layer. Notably, this layer significantly enhances M-u detection but demonstrates suboptimal performance in M-r detection. These findings suggest that the fusion of C2f-FE with RFAConv can effectively enhance the detection of M-u targets without compromising accuracy for M-r targets, thereby resulting in an overall increase in mAP precision. Furthermore, these advancements also yield reduced model sizes compared to YOLOv8n.

By enhancing the YOLOv8n, YOLO-FE, and YOLO-RFE models while employing WIoU as the loss function, a novel improved model is obtained. The findings demonstrate that WioU can augment model performance without inflating its dimensions. Specifically, YOLO-W based on YOLOv8n enhances both F1 and AP metrics in both M-u and M-r detections. In comparison to YOLO-FE, YOLO-FEW boosts F1 by 1.38% and 2.54% in M-u and M-r, respectively, while also improving AP by 0.46% in M-u; only the AP in M-r experiences a marginal decrease of 0.13%. Overall, mAP has witnessed an improvement of 0.17%. Moreover, in YOLO-RFEW, while the F1 of M-u decreased by 0.12%, there was a significant increase of 0.76% in AP compared to YOLO-RFE. Additionally, when compared with YOLO-RFE, there was an improvement of 2.15% in M-r for F1 and a 0.76% increase in AP. The incorporation of the WioU loss function introduces categorical information into the IoU computation and leads to substantial enhancements in both bounding box regression performance and detection accuracy, resulting in a notable increase of 0.55% in mAP.

Building upon YOLO-FEW, YOLO-RFEW introduces RFAConv as a backbone enhancement. In comparison to the original model, which resulted in a marginal increase of 0.09 MB in model size, there was an improvement of 0.69% in mAP. Specifically, F1 and AP values for M-u were enhanced by 0.35% and 1.53%, respectively, while F1 and AP values for M-r experienced a decrease of 1.00% and 0.15%, respectively. These findings indicate that RFAConv effectively balances the detection performance between M-u and M-r components. Compared to YOLOv8n, YOLO-RFEW demonstrates significant advancements in both model detection accuracy and reduction in model size by 1.2 MB; it also enhances the

F1s of M-u and M-r by 4.51% and 4.78%, respectively, while increasing the AP scores of M-u and M-r by 5.2% and 0.23%, respectively, resulting in an overall mAP improvement over the base model by 2%. To summarize, this improved lightweight YOLO-RFEW model not only enhances detection performance but also validates each proposed enhancement through ablation tests.

The size of parameters and FLOPs for each model in the ablation test is shown in Figure 6. It can be observed that YOLO-R, YOLO-RFE, and YOLO-RFEW introduce five RFAConvs into the backbone, resulting in an increase of 0.026 M parameters and 0.3 G FLOPs compared to the original model. This indicates that RFAConv increases the computational cost of the model. On the other hand, YOLO-FE and YOLO-RFE improve C2f from the original model to C2f-FE, reducing parameters by 0.696 M and FLOPs by 1.6 G, respectively, suggesting that C2f-FE is more effective in lightweighting models. Additionally, YOLO-W, YOLO-FEW, and YOLO-RFEW introduce WioU without changing parameters or FLOPs, demonstrating that WioU has no effect on computational cost. Furthermore, YOLO-RFEW reduces parameters by 0.67 M and FLOPs by 1.3 G compared to YOLOv8n, which further confirms that this improved algorithm effectively lightens the model.

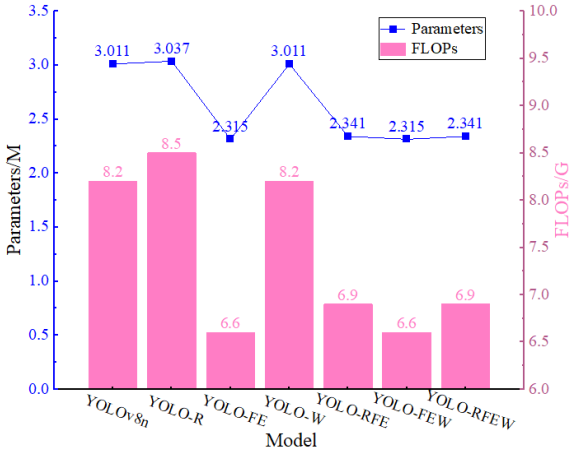

Figure 6. Plot of different model parametric quantities and GFLOPs for ablation test.

To visualize the efficacy of the YOLO-RFEW model in this study, a target detection heat map is employed as a gradient-based visualization technique. This approach generates a heat map by computing the gradient of the feature map to depict the model's attention level across different regions. In this study, Grad-CAM is utilized to produce a heat map, which enhances interpretability by back-propagating the network's gradient onto the input image to determine each pixel's significance for final classification. This information is then leveraged to generate a heat map that illustrates the network's focus on regions of interest within the input image. The heat map of partial test set dataset detection before and after the improvement is shown in Figure 7. From Figure 7a,c, it can be observed that YOLOv8n is susceptible to interference from the background, leaves, etc., and fails to detect smaller fruits located at a distance, whereas YOLO-RFEW demonstrates improved focus on muskmelon fruits. The red part indicates the model's attention towards the fruits. From Figure 7b,d, it can be seen that YOLO-RFEW exhibits higher attentiveness towards the fruits, suggesting that the enhanced model effectively mitigates the impact of background, lighting conditions, and other factors while accurately extracting features of muskmelon fruits.

The detection effects of the target detection models in the test set before and after the improvement are shown in Figure 8. In Figure 8a, YOLO-RFEW demonstrates better overall detection performance than YOLOv8n under frontlighting conditions. In Figure 8b, YOLO-RFEW successfully detects fruits that were missed by YOLOv8n under backlighting conditions. In Figure 8c, YOLO-RFEW improves recognition accuracy for occluded fruits,

while in Figure 8d, it addresses repeated detections observed in YOLOv8n, indicating that adopting the WIoU loss function can enhance target frame prediction ability and consequently improve target detection accuracy. Collectively, the YOLO-RFEW model proposed in this study achieves a detection accuracy of 90.82% while maintaining a compact model size of only 4.75 MB, thereby striking an optimal balance between these two performance indicators. Consequently, the enhanced model effectively facilitates the advancement of intelligent detection technology for muskmelon fruits across various maturity levels.

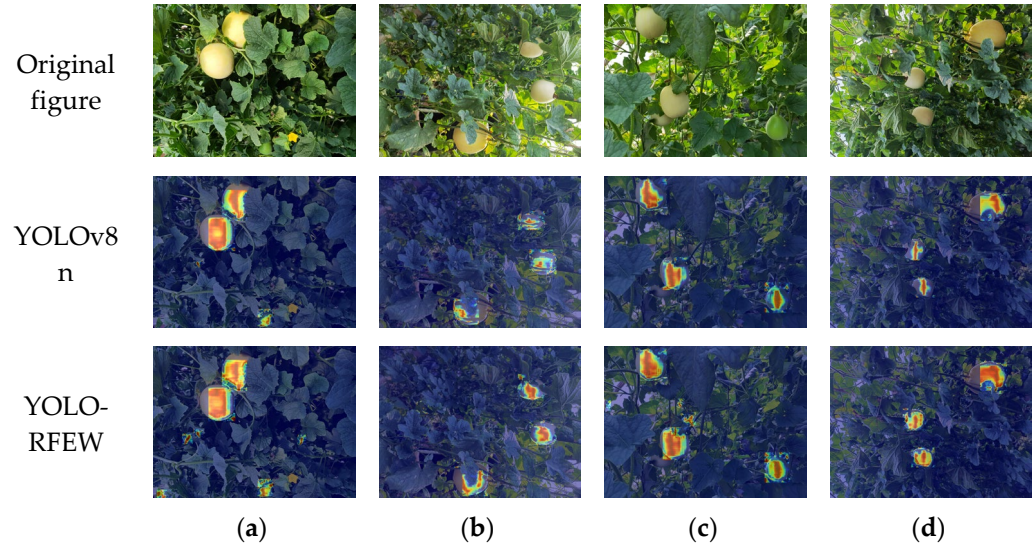

Figure 7. Heat map of dataset detection for some test sets.

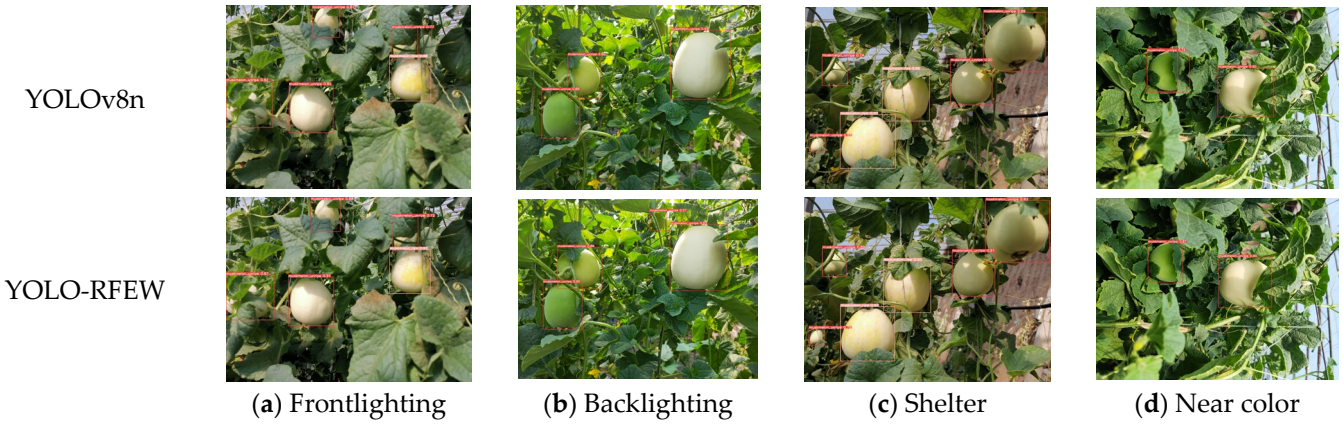

Figure 8. Detection effect graph of some test set datasets.

4. Discussion

Inconsistent stages of fruit ripening in muskmelons result in uneven flowering, with unripe fruits displaying a green and white coloration, while ripe fruits exhibit a yellow hue. Within the greenhouse environment, green muskmelons share a similar color with the leaves and stems, whereas white muskmelons resemble mature ones in terms of their coloration. Furthermore, challenges arise from muskmelon leaves and stems casting shadows on the fruit, as well as the influence of the greenhouse environment and light on model performance. These factors contribute to difficulties encountered when determining whether a harvested muskmelon fruit has reached maturity in field conditions. Henceforth, it is crucial to intelligently and efficiently identify ripe muskmelon fruits within greenhouse environments while minimizing model parameters. This approach facilitates model deployment and establishes the groundwork for real-time harvesting.

The results of different lightweighting models for muskmelon fruit detection are shown in Table 4, where the F1 and mAP of YOLO-RFEW reached 87.91% and 90.82%,

respectively, where the mAP was improved by 2.68% in comparison to YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, YOLOv8s, and YOLOv8n, respectively, 0.57%, 1.36%, 2.58%, 1.31%, and 2.51%, respectively. In addition, the model size of YOLO-RFEW is merely 4.75 MB, representing a reduction of 28.55%, 22.42%, 24.50%, 40.56%, 22.12%, and 79.83% compared to YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, YOLOv8s, and YOLOv8n, respectively. Despite achieving the highest precision of 93.45%, YOLOv4-Tiny exhibits the longest inference time at 2.5 ms; conversely, while YOLOv8s has the highest precision of 94.70%, the minimum recall is 79.60%. YOLOv3-Tiny experiences an inference delay of only an additional 0.3 ms compared to that of YOLO-RFEW.

Table 4. Detection results of muskmelon fruits by different lightweighting models.

Model	Precision (%)	Recall (%)	F1 (%)	mAP (%)	Model Size (MB)	Inference Time (ms)
YOLOv3-Tiny	88.26	84.10	86.13	88.14	16.64	1.8
YOLOv4-Tiny	93.45	81.06	86.82	90.25	21.19	2.5
YOLOv5s	88.91	81.67	85.14	89.46	13.77	1.6
YOLOv7-Tiny	91.23	80.32	85.43	88.24	11.71	1.6
YOLOv8s	94.70	79.60	86.50	89.51	21.47	2.0
YOLOv8n	86.44	80.64	83.44	88.31	5.95	1.7
YOLO-RFEW	93.16	83.22	87.91	90.82	4.75	1.5

To objectively evaluate the performance improvement of the models proposed in this study, a comparative test was conducted by comparing them with lightweight models such as YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, and YOLOv8s. The P-R curves of different models are presented in Figure 9 to demonstrate changes in precision rate and recall rate across various scenarios for performance evaluation. Figure 9 visualizes the trade-off between precision and recall for each model. Although the P-R curve of YOLO-RFEW intersects with other models, it can be inferred that the area under the P-R curve of YOLO-RFEW is larger than that of all other models, indicating that our improved method leads to optimal performance.

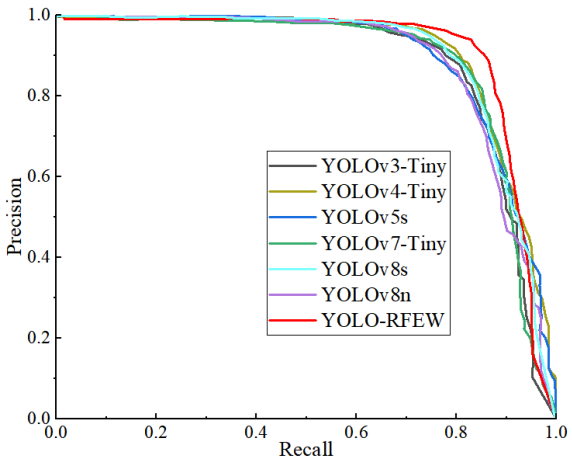

Figure 9. P-R curve.

The present study constructed a dataset comprising muskmelon fruits at various maturity levels for the purpose of detecting muskmelon fruits in greenhouse conditions. To ensure accurate detection, the proposed lightweight detection model YOLO-RFEW effectively reduces the number of model parameters, thereby enhancing stability, generalization, and robustness. The model achieves impressive performance with an mAP of 90.82%, an AP of 87.70% and 93.94% for M-u and M-r, respectively, a compact model size of 4.75 MB, and a fast inference time of 1.5 ms. Moreover, future research should prioritize improving

camera image acquisition efficiency to further enhance the accuracy of the model and provide better support for muskmelon picking robotics technology.

5. Conclusions

The present study aims to achieve real-time and accurate detection of muskmelon fruit ripeness in a greenhouse environment. To this end, this study proposes an enhanced version of the YOLOv8n model by incorporating RFAConv as a replacement for the original Conv in the backbone part. Additionally, this study improves the C2f module to C2f-FE and modifies the loss function to WIoU, thereby enhancing both accuracy and model lightweighting for muskmelon fruit detection. The key findings are summarized as follows:

The Conv layers in the backbone section of the YOLOv8n model were substituted with RFAConv to enhance the model's feature extraction capability. Through a comparative analysis of RFAConv, DWConv, KWConv, and GSConv against the original Conv model, it was observed that RFAConv achieved the highest mAP value of 89.23% among five different convolution types while only increasing the model size by 0.11 MB compared to the original Conv model. To reduce weight and further improve performance, this study enhanced all C2f layers in the original model to C2f-FE. Consequently, precision, recall, F1, and mAP witnessed improvements of 4.34%, 1.67%, 2.9%, and 1.65%, respectively, in the improved YOLO-FE model. Additionally, there was a reduction in model size by 1.29 MB along with a decrease in parameters by 0.696 M compared to YOLOv8n while also achieving an inference time improvement of 0.3 ms.

The effects of different improvements on the model performance were analyzed through ablation tests. Compared to YOLOv8n, the mAP of the six improved models (YOLO-R, YOLO-FE, YOLO-W, YOLO-RFE, YOLO-FEW, and YOLO-FEW) increased by 0.92%, 1.65%, 0.39%, 1.96%, 1.83%, and 2.51%, respectively. WIoU improves bounding box regression performance and detection accuracy while maintaining a constant model size. Additionally, compared to YOLOv8n, the YOLO-RFEW reduces model size by 1.2 MB and parameters by 0.67 M as well as FLOPs by 1.3 G; it also improves F1 by 4.51% and 4.78% for M-u and M-r, respectively; furthermore, it enhances AP by 5.2% for M-u and increases AP slightly for M-r in comparison with YOLOv8n. The improved YOLO-RFEW model improves the detection accuracy on the basis of lightweighting, and the ablation test demonstrates the feasibility and effectiveness of each improvement scheme.

In order to objectively evaluate the performance of YOLO-RFEW, this study conducted a comparative test between the improved model and lightweight models including YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, and YOLOv8s. The results demonstrate that YOLO-RFEW achieves optimal performance on the P-R curve with an mAP that is 2.68%, 0.57%, 1.36%, 2.58%, 1.31%, and 2.51% higher than that of YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, YOLOv8s, and YOLOv8n, respectively. Additionally, compared to other models, the model size can be significantly reduced by using the lightweight model YOLO-RFEW proposed in this study. Specifically, its model size is only YOLOv3-Tiny, YOLOv4-Tiny, YOLOv5s, YOLOv7-Tiny, YOLOv8s, and YOLOv8n of 28.55%, 22.42%, 24.50%, 40.56%, 22.12%, and 79.83%, respectively. Considering both accuracy and size factors, in terms of real-time accurate detection of muskmelon fruit ripeness in greenhouse environment, the method YOLO-RFEW proposed in this study is of great significance for timely and effective picking and precise management of muskmelon.

Author Contributions: Conceptualization, D.X. and H.Z.; methodology, H.Z. and R.R.; validation, D.X., H.Z. and S.Z.; formal analysis, H.Z.; resources, D.X., H.Z. and R.R.; data curation, D.X., H.Z. and R.R.; writing—original draft preparation, R.R.; writing—review and editing, D.X. and H.Z.; project administration, D.X.; funding acquisition, D.X. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key R&D project of introducing high-level scientific and technological talents in Lvliang City, grant number 2021RC-2-24, and the Basic Research Project of Shanxi Province, grant number 202103021223145.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Gothi, H.R.; Patel, P.S.; Raj, V.P.; Rabari, P.H. Diversity and abundance of insect pollinators on muskmelon. *J. Entomol. Res.* **2022**, *46*, 1102–1107. [CrossRef]
2. Xue, Q.; Li, H.; Chen, J.; Du, T. Fruit cracking in muskmelon: Fruit growth and biomechanical properties in different irrigation levels. *Agric. Water Manag.* **2024**, *293*, 108672. [CrossRef]
3. Mayobre, C.; Domingo, M.S.; Özkan, E.N.; Borbolla, A.F.; Lasierra, J.R.; Mas, J.G.; Pujol, M. Genetic regulation of volatile production in two melon introgression line collections with contrasting ripening behavior. *Hortic. Res.* **2024**, *11*, uhae020. [CrossRef] [PubMed]
4. Xu, D.; Zhao, H.; Lawal, O.M.; Lu, X.; Ren, R.; Zhang, S. An Automatic Jujube Fruit Detection and Ripeness Inspection Method in the Natural Environment. *Agronomy* **2023**, *13*, 451. [CrossRef]
5. Zhao, H.; Xu, D.; Lawal, O.; Zhang, S. Muskmelon Maturity Stage Classification Model Based on CNN. *J. Robot.* **2021**, *2021*, 8828340. [CrossRef]
6. Kuznetsova, A.; Maleva, T.; Soloviev, V. Using YOLOv3 Algorithm with Pre- and Post-Processing for Apple Detection in Fruit-Harvesting Robot. *Agronomy* **2020**, *10*, 1016. [CrossRef]
7. Ju, J.; Chen, G.; Lv, Z.; Zhao, M.; Sun, L.; Wang, Z.; Wang, J. Design and experiment of an adaptive cruise weeding robot for paddy fields based on improved YOLOv5. *Comput. Electron. Agric.* **2024**, *219*, 108824. [CrossRef]
8. Mathias, A.; Dhanalakshmi, S.; Kumar, R. Occlusion aware underwater object tracking using hybrid adaptive deep SORT-YOLOv3 approach. *Multimed. Tools Appl.* **2022**, *81*, 44109–44121. [CrossRef]
9. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
10. Solimani, F.; Cardellicchio, A.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [CrossRef]
11. Chen, W.; Liu, M.; Zhao, C.; Li, X.; Wang, Y. MTD-YOLO: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Comput. Electron. Agric.* **2024**, *216*, 108533. [CrossRef]
12. Edy, S.; Suharjito. Hyperparameter optimization of YOLOv4 tiny for palm oil fresh fruit bunches maturity detection using genetics algorithms. *Smart Agric. Technol.* **2023**, *6*, 100364. [CrossRef]
13. Kazama, E.H.; Tedesco, D.; Carreira, V.d.S.; Júnior, M.R.B.; Oliveira, M.F.d.; Ferreira, F.M.; Junior, W.M.; Silva, R.P.d. Monitoring coffee fruit maturity using an enhanced convolutional neural network under different image acquisition settings. *Sci. Hortic.* **2024**, *328*, 112957. [CrossRef]
14. Juntao, X.; Yonglin, h.; Xiao, W.; Zhexing, L.; Haoyan, C.; Qiyan, H. Method of Maturity Detection for Papaya Fruits in Natural Environment Based on YOLO v5-Lite. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 243–252.
15. Chen, W.; Zhang, J.; Guo, B.; Wei, Q.; Zhu, Z. An Apple Detection Method Based on Des-YOLO v4 Algorithm for Harvesting Robots in Complex Environment. *Math. Probl. Eng.* **2021**, *2021*, 7351470. [CrossRef]
16. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight "Yuluxiang" Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [CrossRef]
17. Li, S.; Zhang, S.; Xue, J.; Sun, H. Lightweight target detection for the field flat jujube based on improved YOLOv5. *Comput. Electron. Agric.* **2022**, *202*, 107391. [CrossRef]
18. Hang, J.; Zhao, X.; Gao, F.; Wen, X.; Jing, S.; Zhang, Y. Recognizing and detecting the strawberry at multi-stages usingimproved lightweight YOLOv5s. *Trans. CSAE* **2023**, *39*, 181–187. [CrossRef]
19. Guo, A.; Sun, K.; Zhang, Z. A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection. *J. Real-Time Image Process.* **2024**, *21*, 49. [CrossRef]
20. Kong, D.; Wang, J.; Zhang, Q.; Li, J.; Rong, J. Research on Fruit Spatial Coordinate Positioning by Combining Improved YOLOv8s and Adaptive Multi-Resolution Model. *Agronomy* **2023**, *13*, 2122. [CrossRef]
21. Zhichao, H.; Yi, W.; Junping, W.; Wanli, X.; Bilian, L. Improved Lightweight Rebar Detection Network Based on YOLOv8s Algorithm. *Adv. Comput. Signals Syst.* **2023**, *7*, 107–117. [CrossRef]
22. Zhang, X.; Liu, C.; Yang, D.; Song, T.; Ye, Y.; Li, K.; Song, Y. RFAConv: Innovating Spatial Attention and Standard Convolutional Operation. In Proceedings of the Computer Vision and Pattern Recognition, Xiamen, China, 26–28 April 2024. [CrossRef]
23. Chen, J.; Kao, S.H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023. [CrossRef]
24. Zhang, J.; Wang, W.; Li, X.; Han, Y. Recognizing facial expressions based on pyramid multi-head grid and spatial attention network. *Comput. Vis. Image Underst.* **2024**, *244*, 104010. [CrossRef]
25. Yasir, K.M.M.; Qingxian, W.; Bo, C.; Weidong, W. Cross-modality representation learning from transformer for hashtag prediction. *J. Big Data* **2023**, *10*, 148. [CrossRef]

26. Viet, B.D.; Masao, K.; Hiroshi, S. Attention-based neural network with Generalized Mean Pooling for cross-view geo-localization between UAV and satellite. *Artif. Life Robot.* **2023**, *28*, 560–570. [CrossRef]
27. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Liu, H. Expectation-Maximization Attention Networks for Semantic Segmentation. In Proceedings of the International Conference in Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
28. Zanjia, T.; Yuhang, C.; Zewei, X.; Rong, Y. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
29. Xu, W.; Wan, Y. ELA: Efficient Local Attention for Deep Convolutional Neural Networks. *arXiv* **2024**, arXiv:2403.01123.
30. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Jeju Island, Republic of Korea, 23–25 April 2021. [CrossRef]
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
32. Ren, R.; Sun, H.; Zhang, S.; Zhao, H.; Wang, L.; Su, M.; Sun, T. FPG-YOLO: A detection method for pollenable stamen in 'Yuluxiang' pear under non-structural environments. *Sci. Hortic.* **2024**, *328*, 112941. [CrossRef]