

RESEARCH

Open Access



# Melon ripeness detection by an improved object detection algorithm for resource constrained environments

Xuebin Jing<sup>1,2</sup>, Yuanhao Wang<sup>1,2</sup>, Dongxi Li<sup>1\*</sup> and Weihua Pan<sup>2\*</sup>

## Abstract

**Background** Ripeness is a phenotype that significantly impacts the quality of fruits, constituting a crucial factor in the cultivation and harvesting processes. Manual detection methods and experimental analysis, however, are inefficient and costly.

**Results** In this study, we propose a lightweight and efficient melon ripeness detection method, MRD-YOLO, based on an improved object detection algorithm. The method combines a lightweight backbone network, MobileNetV3, a design paradigm Slim-neck, and a Coordinate Attention mechanism. Additionally, we have created a large-scale melon dataset sourced from a greenhouse based on ripeness. This dataset contains common complexities encountered in the field environment, such as occlusions, overlapping, and varying light intensities. MRD-YOLO achieves a mean Average Precision of 97.4% on this dataset, achieving accurate and reliable melon ripeness detection. Moreover, the method demands only 4.8 G FLOPs and 2.06 M parameters, representing 58.5% and 68.4% of the baseline YOLOv8n model, respectively. It comprehensively outperforms existing methods in terms of balanced accuracy and computational efficiency. Furthermore, it maintains real-time inference capability in GPU environments and demonstrates exceptional inference speed in CPU environments. The lightweight design of MRD-YOLO is anticipated to be deployed in various resource constrained mobile and edge devices, such as picking robots. Particularly noteworthy is its performance when tested on two melon datasets obtained from the Roboflow platform, achieving a mean Average Precision of 85.9%. This underscores its excellent generalization ability on untrained data.

**Conclusions** This study presents an efficient method for melon ripeness detection, and the dataset utilized in this study, alongside the detection method, will provide a valuable reference for ripeness detection across various types of fruits.

**Keywords** Ripeness detection, Object detection, Melon, Deep learning

\*Correspondence:

Dongxi Li  
lidongxi@tyut.edu.cn  
Weihua Pan  
panweihua@caas.cn

Full list of author information is available at the end of the article



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

## Background

The melon, a fruit of the Cucurbitaceae family with significant economic value, is renowned for its distinctive aroma and sweet taste, making it highly favored by consumers and presenting a broad market prospect [1]. Additionally, melon is recognized for its health benefits, as it is exceptionally rich in nutritional value [2]. Beyond common factors like soil quality, climate, moisture, pests, and diseases, fruit ripeness plays a crucial role in determining fruit quality [3]. As a phenotype that significantly influences the growth and harvesting process of melon, achieving appropriate harvesting maturity ensures optimal sweetness, flavor, and nutritional value.

Ripeness denotes the stage at which fruit reaches full development, and the timing of harvest maturity significantly influences both the quality and marketability of the fruit. Real-time detection of fruit ripeness during its growth not only ensures the quality of harvested fruit but also mitigates post-harvest losses [4]. Harvesting fruit prematurely, before it reaches full ripeness, can result in diminished flavor and nutrient content. Additionally, unripe fruit typically exhibits a firmer texture and is less palatable compared to fully ripe fruit [5]. Conversely, overripe fruit is more prone to mold, rot, and spoilage, thereby shortening its shelf life and diminishing overall quality. Furthermore, certain nutrients may degrade in overripe fruit, leading to a potential loss of nutritional value [6]. Therefore, fruit ripeness detection is crucial to maximizing the flavor, texture, nutrient content, and shelf life of various fruits, such as melon.

Numerous methods exist for gauging the ripeness of melon, each differing in cost, time investment, and accuracy. Traditionally, growers rely on their experience to assess ripeness, considering factors such as color, aroma, texture, and weight, or resorting to tasting samples directly. However, these traditional methods, though economical, are marred by time-consuming procedures, labor-intensive practices, and subjective evaluations, resulting in imprecise determinations. Furthermore, the diverse ripening traits exhibited by various melon types hinder the universal applicability of these approaches.

More precise methods for detecting melon ripeness have emerged with advancements in inspection techniques. Mitsuru et al. investigated the temporal variation of the elasticity index of melons as a means to predict ripening progression [7]. They also utilized a non-destructive acoustic vibration method to monitor the elasticity index, thereby estimating the time required for melons to achieve optimal ripeness [8]. Sun et al. utilized hyperspectral imaging for the non-invasive evaluation of melon texture, enabling the prediction of various quality attributes [1]. Yang et al. conducted optical measurements to analyze fruit quality and investigate the propagation of

light through fruits, providing insights into the correlation between optical properties and fruit quality [9]. Calixto et al. implemented a non-destructive computer vision-based approach, utilizing textural differences in the yellow color of melon skin to determine ripeness [10]. These methods offer enhanced accuracy over traditional approaches for assessing melon quality and ripeness from diverse perspectives. However, these experiment-based detection methods inevitably entail higher costs and lower efficiency.

With the rapid progress of technologies like deep learning and computer vision, fruit ripeness detection methods are no longer confined to traditional manual judgment or experimental analysis [11]. Halstead et al. introduced a vision system that accurately estimates the ripeness of sweet peppers using the Faster R-CNN framework, validated through field data [12]. Wan et al. devised a tomato ripeness detection method leveraging computer vision technology to assess the ripeness level of tomato samples based on their color features [13]. Tu et al. developed a machine vision method to detect passion fruit and determine its ripeness using a linear support vector machine classifier [14]. Chen et al. proposed a method that integrates visual saliency with object detection algorithms for citrus ripeness recognition [15]. Wang et al. utilized the category balancing method along with deep learning techniques to detect and segment tomatoes at various maturity levels [16].

Compared to traditional methods and experimental analyses for determining fruit ripeness, computer vision and deep learning-based methods offer higher detection efficiency without the need for costly and labor-intensive experimental processes. However, as far as we are aware, most existing methods, while capable of accurately detecting melon ripeness, rely on experimental techniques that measure specific attributes of the fruit. Moreover, computer vision-based methods for assessing melon ripeness are typically limited to analyzing individual fruit images and have not yet been deployed in real field environments.

One of the main factors contributing to the scarcity of research focusing on melon ripeness in large field environments is the absence of high-quality large datasets. Existing melon datasets suffer from various issues, including low image resolution, insufficient volume of data, and absence of systematic ripeness classification in image annotations [17, 18]. Furthermore, detecting melon ripeness in field environments imposes stricter demands on methodologies. These requirements extend beyond achieving high detection accuracy to include considerations of computational efficiency. Given that edge device such as picking robots, smartphones, and agricultural drones, commonly deployed in such

environments, have limited computational capabilities, it becomes imperative to develop lightweight detection methods suitable for resource constrained settings. In light of the aforementioned issues in existing studies, the overarching objective of our research is to create a comprehensive melon dataset of high quality and to propose an accurate and lightweight method for detecting melon ripeness. The contributions of this study are summarized as follows:

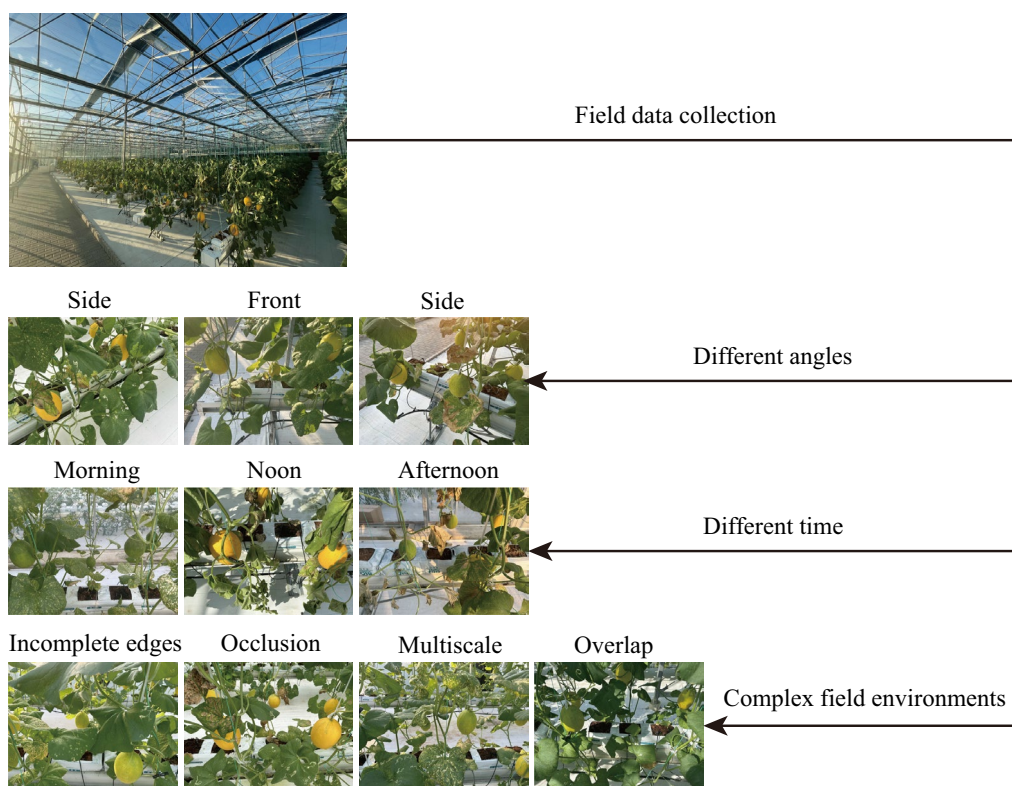
- We created a large, high-quality melon dataset collected in a real field environment, categorized according to ripeness.
- We proposed a lightweight, improved object detection algorithm, MRD-YOLO, tailored for resource constrained environments. The method builds upon the latest YOLOv8 network, opting for MobileNetV3 as the backbone network to reduce model size. Additionally, a slim-neck design paradigm was employed in the neck segment to further diminish the number of parameters and computational complexity while maintaining ample accuracy. We introduced coordinate attention to improve the model's capacity to handle feature maps, thereby boosting its performance in detecting targets within complex scenes.

- A series of ablation experiments, comparison studies, and other analyses underscored the outstanding performance of our proposed lightweight MRD-YOLO algorithm in melon ripeness detection tasks. Furthermore, we assessed the generalization capability of the MRD-YOLO model using two additional melon datasets sourced from Roboflow. The experimental results reaffirmed the effectiveness of our method, not only within the datasets we constructed ourselves, but across diverse datasets as well.

## Materials and methods

### Field data collection

The melon dataset utilized in this study was procured from a greenhouse situated at the Shenzhen Experimental Base of the Chinese Academy of Agricultural Sciences, located on Pengfei Road, Dapeng New District, Shenzhen, Guangdong Province, China. The images were captured during the months of October and November 2023. The process of field data collection is illustrated in Fig. 1. This dataset comprises images captured from various angles to ensure the model's efficacy in extracting comprehensive melon features. The temporal diversity of the images presents the model with the challenge of recognizing targets amidst shadows and bright light



**Fig. 1** Field data collection process in a greenhouse

conditions. Despite the controlled environment within the greenhouse, conducive for optimal growth and sufficient plant spacing, factors such as target overlap, leaf shading, and variations in melon sizes and scales across different growth stages posed significant challenges to our detection. Ultimately, a total of 3806 high-quality melon images with a resolution of  $4032 \times 3024$  were acquired to construct the dataset for this study.

### Image annotation and preprocessing

In Fig. 2, it can be observed that the rind of the melon in its early growth stage exhibits a green hue, resembling that of the leaves, and the fruits are relatively small in size. As the melon progresses towards maturity, the color of the rind gradually transitions to yellow. Upon reaching full ripeness, the rind assumes a golden yellow hue, appearing smooth or occasionally displaying unstable, thin reticulation patterns. We classified fully ripe melons as 'ripe' and those not yet ripe as 'unripe', utilizing the YOLO format within the LabelImg software. Subsequently, the images, along with their corresponding labels, were randomly divided into training, validation, and test sets in an 8:1:1 ratio. The resulting training set comprised 3044 images, with 381 images allocated to both the validation and test sets, respectively. Additionally, we utilized the melon dataset obtained from Roboflow as an independent test set [17, 18], ensuring that none of the data from this set were involved in the training. The efficacy of our model's generalization was assessed through tests conducted on this independent dataset.

### The proposed MRD-YOLO architecture for melon ripeness detection

In this study, we introduce the MRD-YOLO model designed for melon ripeness detection, with its overall architecture depicted in Fig. 3. Enhancing the baseline YOLOv8n [19], we implemented several key improvements. Firstly, we opted for MobileNetV3 as a replacement for the backbone network in YOLOv8, effectively reducing the model's size [20]. Additionally, by adopting the Slim-neck design paradigm, we substituted the Conv and C2f structure in the baseline's neck segment with GSConv and VoV-GSCSP, respectively [21]. Leading to a

further reduction in both parameter count and computational complexity while maintaining sufficient accuracy. Lastly, we introduced Coordinate Attention after VoV-GSCSP within the neck segment [22], enhancing the model's processing capabilities for feature maps and subsequently improving detection performance, particularly in complex scenes. The subsequent section delineates the specifics of these model enhancements.

### MobileNetV3: a lightweight network architecture

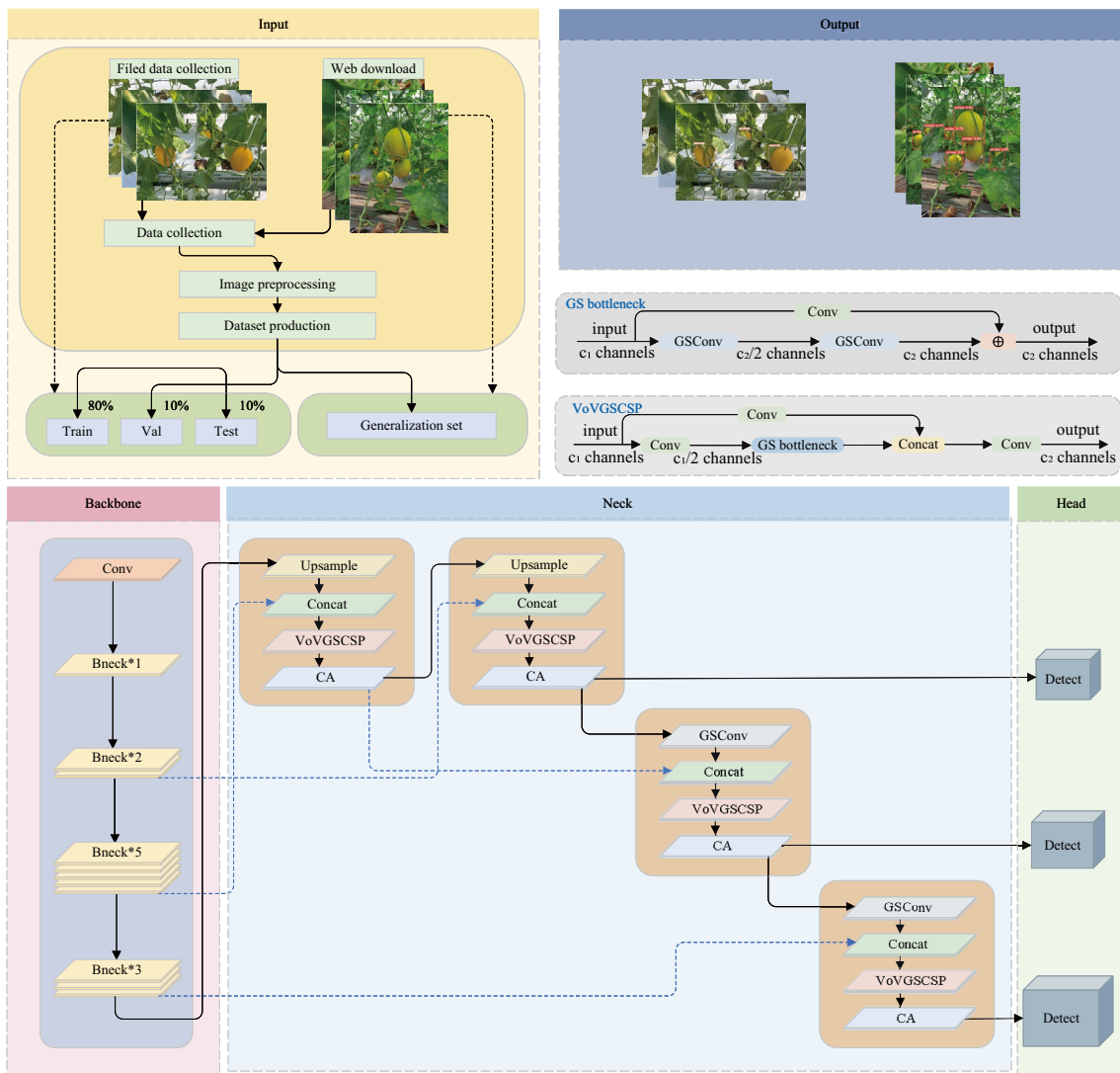
MobileNetV3 is a lightweight network architecture optimized for embedded and mobile devices, with the dual objectives of achieving high accuracy while maintaining low computational cost and memory footprint. The structure of the basic module bottleneck (bneck) within the MobileNetV3 network is illustrated in Fig. 4. Bneck denotes a layer structure comprising a sequence of operations, including depthwise convolution, pointwise convolution, as well as other operations such as activation functions and batch normalization. This module primarily integrates channel separable convolution, SE channel attention mechanism, and residual connection. By employing bneck, the network effectively reduces computational complexity while still capturing essential features. Therefore, we utilize it as the backbone of MRD-YOLO, leveraging its capabilities for melon ripeness detection task.

### Slim-neck: a better design paradigm of neck architectures

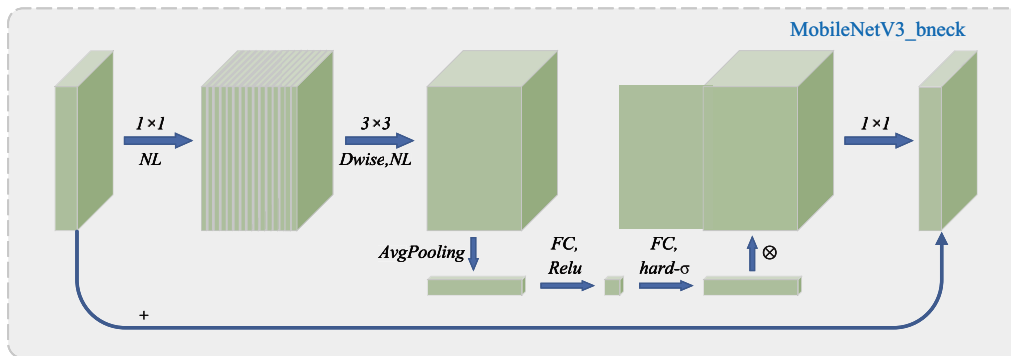
Slim-neck is a structural design paradigm aimed at reducing the computational complexity and inference time of detectors in Convolutional Neural Networks while maintaining accuracy. It encompasses the incorporation of several lightweight techniques, such as GSConv, GS bottleneck, and VoV-GSCSP (cross stage partial network), with GSConv structure depicted in Fig. 5. GSConv, as the basic component of slim-neck, is designed to align the outputs of Depthwise Separable Convolution (DSC) as closely as possible to those of Standard Convolution (SC). It combines elements from both DSC and SC along with a shuffling operation. Slim-neck achieves a harmonious equilibrium between model efficiency and accuracy through the strategic utilization of these techniques. This design is grounded in the



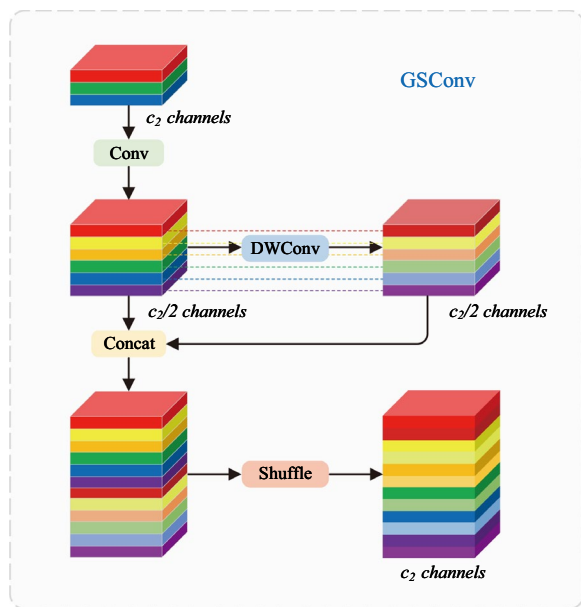
**Fig. 2** Different levels of melon ripeness



**Fig. 3** Overall architecture and process of the proposed MRD-YOLO method



**Fig. 4** Bneck structure of the MobileNetV3

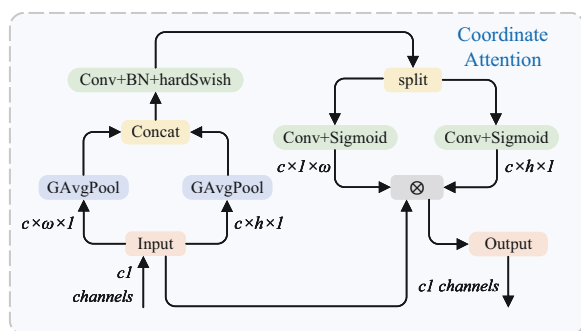


**Fig. 5** Structure of the GSConv module

principles of existing methodologies like DenseNet [23], VoVNet [24], and CSPNet [25], which have been adapted to fashion a streamlined and efficient architecture suitable for real-time detection tasks, particularly in scenarios where computational resources may be constrained. Moreover, it aligns seamlessly with our objective of proposing a lightweight model.

**Coordinate attention: an attention mechanism for efficient mobile network design**

Coordinate Attention is an attention mechanism designed to enhance a model’s capability to focus on a particular spatial location within an image by incorporating positional information into channel attention. This enables the mobile network to gather information about a broader area without introducing significant overhead. Its structure is illustrated in the Fig. 6.



**Fig. 6** Structure of the Coordinate Attention

Coordinate Attention serves several crucial roles in our melon ripeness detection. Firstly, it enables the model to concentrate on specific spatial locations within an image, facilitating the gathering of contextual information around these areas. This contextual understanding proves invaluable for accurately detecting melon targets amidst complex backgrounds or cluttered scenes. Moreover, by attending to coordinates at various scales, the model adeptly detects objects of different sizes, thus effectively handling scale variations of melons within the image, which may occur at different growth stages. Importantly, the integration of coordinate attention into our proposed MRD-YOLO framework does not introduce additional computational overhead, further underscoring its significance.

**Results**

**Experimental environment and parameter settings**

In this study, all experiments were conducted on a server equipped with an Intel® Xeon® Gold 6230R CPU@2.10GHz, with 100GB of memory, and a Tesla V100S-PCIE graphics card. All detection models involved in this study were developed on the same Linux platform, utilizing Python 3.8.16, torch 1.10.1, and CUDA 11.1. Additionally, the parameter settings are presented in Table 1 and were consistently maintained throughout the experiments.

**Evaluation metrics**

To meticulously evaluate the performance of the proposed MRD-YOLO model for melon ripeness detection, we adopt six evaluation metrics: P (Precision), R (Recall), FLOPs (Floating-point operations), FPS (Frames Per Second), mAP (mean Average Precision), and the number of parameters. Precision, recall, and mAP indicators are evaluated using the following equations:

**Table 1** Initialization parameters settings

Parameter	Setting
Input image size	640x640
Training epochs	100
Batch size	16
Optimizer	AdamW
Momentum	0.937
Weight decay	0.0005
Warmup epochs	3
Warmup momentum	0.8
Initial learning rate	0.001667

$$R = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{1}$$

$$P = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

$$\text{mAP} = \frac{1}{|C|} \sum_{c \in C} AP(c) \tag{3}$$

Precision measures the accuracy of positive predictions made by a model, while recall measures the ability of a model to identify all relevant instances. In Eqs. 3,  $C$  represents the set of object classes and  $AP$  stands for average precision.

These three metrics collectively offer insights into the trade-offs between the correctness and completeness of our proposed model’s predictions. A good model typically achieves a balance between precision and recall, leading to a high mAP score, indicating accurate and comprehensive performance.

Beyond these metrics, FPS, FLOPs, and the number of parameters are essential for assessing the computational efficiency and complexity of a model. FPS measures how many frames or images a model can process in one second. The following equation illustrates how FPS is calculated:

$$\text{FPS} = \frac{1}{\text{time per frame}} \tag{4}$$

FLOPs represent the number of floating-point operations required by a model during inference. It is calculated based on the operations performed in each layer of the network. Parameters refers to the total count of weights and biases used by the model during training and inference, which can be calculated by summing up the individual parameters in each layer, including weights and biases.

By focusing on these three metrics, we can effectively assess whether a model is lightweight or not. Models with lower FLOPs, and fewer parameters are generally considered lightweight and are better suited for deployment on resource constrained environments or for applications requiring efficient processing.

In summary, these six detection metrics provide a comprehensive framework for evaluating the performance and efficiency of our lightweight melon ripeness detection model, ensuring that it meets the stringent requirements of accuracy, speed, and resource efficiency demanded by complex agricultural environments.

### Ablation experiments

#### *Ablation experiments of the baseline and the proposed improvements*

To assess the efficacy of MobileNetV3, Slim-neck, and Coordinate Attention in enhancing the effectiveness of MRD-YOLO, we integrated these proposed improvements into the YOLOv8n model for individual ablation experiments. Precision, recall, FLOPs, FPS (GPU), mean Average Precision (mAP), and the number of parameters were the metrics employed in these ablation experiments, conducted within a GPU environment with consistent hardware settings and parameter configurations. Additionally, we evaluated the model’s inference speed within a CPU environment, with the experiment results detailed in Table 2.

MobileNetV3 aims to provide advanced computational efficiency while achieving high accuracy, as demonstrated by its excellent performance in our ablation experiments. Substituting the backbone network in the baseline model with MobileNetV3 resulted in a reduction in FLOPs from 8.2 G to 5.6 G, accompanied by a 25% decrease in the number of parameters. MobileNetV3 plays a crucial role in improving the efficiency of the YOLOv8, reducing both the number of parameters and computational cost. This is achieved while maintaining precision, recall, and

**Table 2** Performance comparison of the baseline and the proposed improvements

Methods	P(%)	R(%)	mAP(%)	Params(M)	FLOPs(G)	FPS	
						GPU	CPU
YOLOv8n	91.9	94.5	96.8	3.01	8.2	84.47	23.13
Slim-Neck	94.4	93.8	97.1	2.80	7.4	70.91	23.26
CA	93.1	94.3	97.3	3.03	8.2	84.96	22.82
MobileNetV3	95.0	92.6	96.8	2.25	5.6	73.79	28.76
MobileNetV3+CA	95.3	92.3	96.9	2.27	5.6	72.56	27.89
MobileNetV3+Slim-neck	93.3	93.4	96.8	2.04	4.8	69.90	22.30
Slim-neck+CA	93.4	93.9	97.3	2.81	7.4	68.91	23.63
MRD-YOLO	95.0	94.1	97.4	2.06	4.8	61.74	22.65

mAP across various configurations, such as Slim-neck and Coordinate Attention.

Slim-neck, a design paradigm targeting lightweight architectures, proves to be equally indispensable in our melon ripeness detection model. In our study, we implemented VoV-GSCSP, a one-shot aggregation module, to replace the c2f structure of the neck segment in YOLOv8n. Additionally, we integrated GSConv, a novel lightweight convolution method, to substitute the standard convolution operation within the neck. These enhancements collectively yielded an improvement in both accuracy and computational efficiency. However, the adoption of slim-neck also resulted in a 16% decrease in FPS on the GPU compared to the baseline.

Coordinate Attention, a novel lightweight attention mechanism designed for mobile networks, was introduced before various sizes of detection heads to enhance the representations of objects of interest. Integrating the CA attention mechanism into the baseline YOLOv8n yielded a 0.05% increase in mAP with minimal parameter growth and computational overhead.

By incorporating the three enhancements of MobileNetV3, Slim-neck, and Coordinate Attention, our proposed MRD-YOLO model achieves a mAP of 97.4%, requiring only 4.8 G FLOPs and 2.06 M parameters. This model comprehensively outperforms the baseline YOLOv8n in terms of accuracy and computational efficiency.

While MRD-YOLO model maintains real-time detection performance, it exhibits a certain degree of FPS loss on GPU. However, not all devices are equipped with powerful GPUs, making it equally important to assess our model's inference speed on a CPU. With the integration of MobileNetV3, the FPS of the baseline YOLOv8n model increased by 24.3%, and the inference speed gap between MRD-YOLO and the baseline vanished in the CPU environment.

Overall, the lightweight and efficient design of MRD-YOLO makes it the preferred choice for melon ripeness detection, particularly for applications requiring

deployment on mobile and embedded devices with limited computational resources.

**Ablation experiments of the MRD-YOLO and coordinate attention**

We then conducted a more specific ablation experiment to analyze the impact of Coordinate Attention on the effectiveness of the MRD-YOLO. We integrated the Coordinate Attention before the different sizes of detection heads to assess its influence on our proposed method, with the experiment results depicted in Table 3. The inclusion of Coordinate Attention has a negligible effect on the number of parameters and the computational load. This aligns perfectly with our requirement for a lightweight design for our models. Additionally, the highest mAP of 97.4% was achieved by adding the CA mechanism before all different sizes of detection heads. These findings underscore the efficacy of Coordinate Attention in emphasizing melon features.

**Comparison experiments**

To thoroughly validate the performance of the proposed MRD-YOLO for melon ripeness detection, we conducted comparisons between MRD-YOLO and various lightweight backbone, attention mechanisms, as well as six state-of-the-art detection methods.

**Comparison experiments of the lightweight backbones**

In this section, we assess the efficacy of various lightweight backbones in comparison to MobileNetV3 for detecting melon ripeness. These backbones include MobileNetV2 [26], ShuffleNetV2 [27], and VanillaNet with distinct layer configurations [28]. Subsequently, we integrate these backbone networks into YOLOv8 and present the comparative outcomes in Table 4.

ShuffleNetV2 is a convolutional neural network architecture designed for efficient inference on mobile and embedded devices. It demonstrates outstanding inference speed in GPU environments while maintaining a low number of parameters and computational

**Table 3** Comparison of adding Coordinate Attention to different detection heads in MRD-YOLO

Small	Medium	Large	mAP(%)	Parameters(M)	FLOPs(G)	FPS(GPU)
✓			96.8	2.05	4.8	63.98
	✓		97.0	2.05	4.8	65.02
		✓	96.9	2.05	4.8	63.81
✓	✓		97.1	2.05	4.8	61.31
	✓	✓	96.6	2.06	4.8	61.44
✓		✓	97.0	2.05	4.8	62.09
✓	✓	✓	97.4	2.06	4.8	61.74



**Table 4** Comparison of integrating lightweight backbone networks into YOLOv8

Methods	P(%)	R(%)	mAP(%)	Params(M)	FLOPs(G)	FPS	
						GPU	CPU
ShuffleNetV2	94.3	91.0	96.4	1.83	5.1	87.88	25.99
VanillaNet-6	94.0	92.8	96.6	2.11	5.9	100.42	25.85
VanillaNet-8	93.7	93.6	96.7	2.15	6.1	97.59	23.26
VanillaNet-10	95.0	91.9	96.8	2.22	6.4	91.99	24.04
VanillaNet-12	93.0	93.8	96.9	2.23	6.6	84.22	20.35
MobileNetV2	93.3	93.3	96.7	1.79	5.4	58.40	24.86
MobileNetV3	95.0	92.6	96.8	2.25	5.6	73.79	28.76

load. However, its performance in terms of accuracy is not as strong.

VanillaNet, characterized by its avoidance of high depth, shortcuts, and intricate operations such as self-attention, presents a refreshingly concise yet remarkably powerful approach. VanillaNet with different configurations demonstrates robust performance in terms of efficiency. For instance, VanillaNet-6 achieves an fps of over 100 in a GPU environment with only 5.9 G FLOPs and 2.11 M parameters, albeit with a slightly lower accuracy of 96.6%. Conversely, VanillaNet-12 attains a higher mAP value of 96.9%. However, it does not match the computational efficiency of VanillaNet-6, and its model complexity is higher. Furthermore, in a CPU environment, VanillaNet-12's fps metrics are merely 65% of MobileNetV3.

MobileNetV3 emerges as the top performer among the MobileNet series networks. Compared to its predecessors, MobileNetV3 excels in maintaining detection accuracy while enhancing inference speed. The integration of squeeze-and-excitation blocks and efficient inverted residuals further enhances its efficiency and accuracy, rendering it particularly well-suited for resource constrained environments, also making it ideal for melon ripeness detection.

#### Comparison experiments of the attention mechanisms

Attention mechanisms serve as a powerful tool for enhancing the performance, interpretability, and efficiency of deep learning models across various tasks and domains. In this section, we conducted comparative experiments on six attention mechanisms to ascertain their efficacy in enhancing the performance of our melon ripeness detection task. These mechanisms include SE attention (Squeeze-and-Excitation Networks) [29], CBAM attention (Convolutional Block Attention Module) [30], GAM attention (Global Attention Mechanism) [31], Polarized Self-Attention [32], NAM attention (Normalization-based Attention Module) [33], Shuffle attention [34], SimAM attention (Simple, Parameter-Free Attention Module) [35] and CA attention (Coordinate Attention) [22].

As depicted in Table 5, with the exception of GAM attention and Polarized Self-Attention, the remaining attention mechanisms lead to negligible increases in the model's parameter count and computational requirements. When considering the trade-off between model speed and detection accuracy, neither SE attention nor NAM attention demonstrates satisfactory performance. SE attention suffers from a low mAP score and an imbalance between precision and recall. Similarly, NAM attention also fails to achieve higher detection precision

**Table 5** Comparison of adding different attention mechanisms in YOLOv8

Methods	P(%)	R(%)	mAP(%)	Parameters(M)	FLOPs(G)	FPS(GPU)
GAM	94.5	93.4	97.1	3.70	9.6	81.76
PolarizedSelf	92.2	95.0	97.2	3.22	8.6	82.30
NAM	93.6	93.3	96.7	3.01	8.2	86.85
SE	96.1	91.6	96.9	3.01	8.2	83.45
Shuffle	93.1	93.7	97.1	3.01	8.2	82.44
SimAM	95.3	90.9	97.0	3.01	8.2	89.13
CBAM	94.3	93.4	97.2	3.11	8.3	81.97
CA	93.1	94.3	97.3	3.03	8.2	84.96

despite its fast fps rate. Notably, Shuffle attention, SimAM attention, CBAM attention and CA attention demonstrate comparable performance improvements, yet CA attention exhibits a slight advantage across all metrics.

Coordinate Attention, which operates by attending to different regions of the input based on their spatial coordinates, explicitly encodes positional information into the attention mechanism. This enables the model to focus on specific regions of interest based on their coordinates in the input space. CA attention achieves the highest mAP value of 97.3% and effectively balances inference speed with detection accuracy. The findings of the aforementioned comparison experiments underscore the significance of CA attention, establishing it as an integral component of melon ripeness detection task.

**Comparison of performance with state-of-the-art detection methods**

To further analyze the effectiveness of MRD-YOLO in implementing the melon ripeness detection task, we conducted a comprehensive performance analysis. This analysis involved comparing our proposed MRD-YOLO model with six state-of-the-art object detection methods: YOLOv3-tiny [36], YOLOv5n [37], YOLOR [38], YOLOv7-tiny [39], YOLOv8n [16], and YOLOv9 [40]. The comparison results are presented in the Table 6.

YOLOv3-tiny demonstrates commendable performance in terms of FPS, particularly on GPU. However, it exhibits higher parameter counts and lower accuracy compared to MRD-YOLO. Similarly, while YOLOv5n boasts minimal floating-point operations, its insufficient accuracy fails to meet our requirement for precise detection.

YOLOv7-tiny demonstrates a relatively balanced performance across all metrics. However, when compared to the baseline YOLOv8n, it significantly lags behind in all metrics, except for a slight lead in the mAP value.

YOLOv8n exhibits comprehensive and overall strong performance in our melon ripeness detection task, achieving a mAP of 96.8% with a relatively low number of parameters and computational complexity. Moreover, it demonstrates fast inference capability in both GPU and CPU environments. These factors are why we selected it as our baseline model.

Although YOLOv9, RT-DETR and YOLOR achieve high mAP, they come with substantially higher computational requirements, and their inference speeds do not meet the real-time detection needs of our task. The latest model in the YOLO series, YOLOv10, demonstrates commendable performance, comparable to that of YOLOv8.

Our proposed lightweight model, MRD-YOLO, achieves a mAP of 97.4%, which is only slightly lower than that of YOLOv9, while the latter requiring high computational resources. The incorporation of Slim-neck and MobileNetV3 contributes to the model's parameter count being the lowest among all the compared models, standing at 2.06 M. Additionally, MRD-YOLO maintains real-time detection capability in a GPU environment. Furthermore, in comparison to its inference speed in a GPU environment, MRD-YOLO demonstrates significantly improved performance in a CPU environment with lower computational power. These reaffirm the suitability of our proposed model for the melon ripeness detection task, as it successfully balances detection speed and accuracy while also performing admirably in resource constrained environments.

**Comparison of detection results with state-of-the-art detection methods**

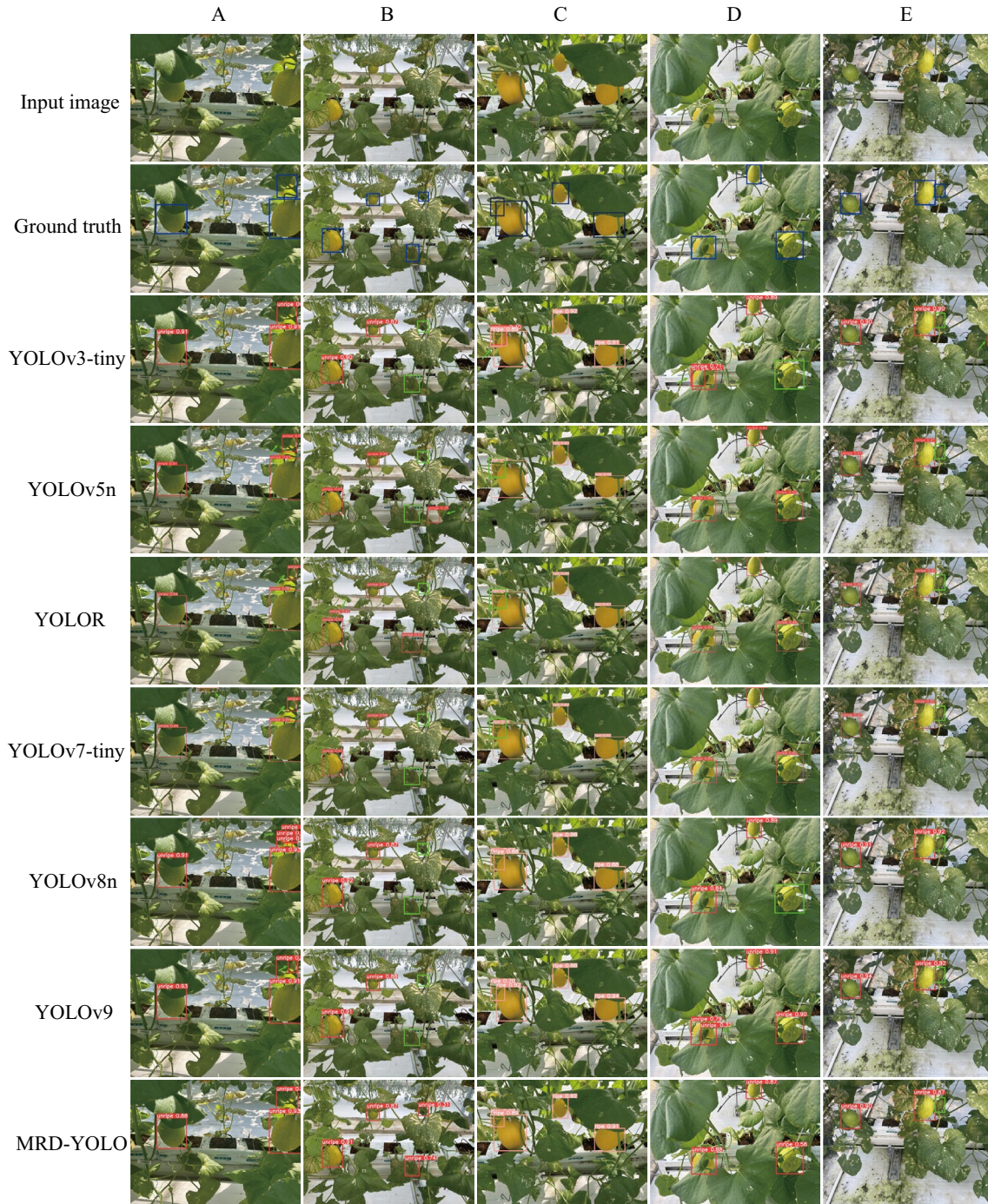
To further demonstrate the effectiveness of the proposed MRD-YOLO, we conducted experiments to compare the actual detection results of seven object detection methods for melon ripeness detection. The input images

**Table 6** Performance comparison of MRD-YOLO with state-of-the-art detection methods

Methods	P(%)	R(%)	mAP(%)	Params(M)	FLOPs(G)	FPS	
						GPU	CPU
YOLOv3-tiny	93.8	92.5	96.2	12.13	19.0	131.13	25.16
YOLOv5n	93.0	91.6	95.7	3.10	4.3	72.99	23.7
YOLOvR	95.7	91.7	97.3	52.50	119.7	37.59	3.92
YOLOv7-tiny	93.8	92.2	96.9	6.02	13.2	64.94	12.32
YOLOv8n	91.9	94.5	96.8	3.01	8.2	84.47	23.13
YOLOv9	94.0	94.3	97.7	60.80	266.1	27.47	3.51
YOLOv10	94.5	92.7	97.1	2.71	8.4	80.42	21.34
RT-DETR	92.6	93.9	96.1	32.8	108.3	76.20	11.56
MRD-YOLO	95.0	94.1	97.4	2.06	4.8	61.74	22.65

encompassed varying numbers of melons, distinct ripeness levels, varying degrees of occlusion, and diverse shooting angles and light intensities, aiming to comprehensively assess the detection capabilities of the seven methods.

As illustrated in Fig. 7, a portion of the melon situated in the upper right corner of Fig. 7A and the lower left corner of Fig. 7 is occluded by leaves, a common scenario in real agricultural environments. This occlusion creates a truncated effect, wherein a complete



**Fig. 7** Comparison of the detection results. The green arrow indicates instances of incorrect detection, and the green box denotes the missing object

melon appears to be fragmented into multiple parts. Moreover, some melons depicted in Fig. 7B and Fig. 7E exhibit a color resemblance to the leaves and are diminutive in size due to their early growth stage, posing challenges for the model’s detection capabilities. Additionally, the two melons on the left in Fig. 7C exhibit significant overlap. Through an analysis of the detection results produced by the seven detection methods for these intricate images, a comprehensive assessment of each method’s detection capabilities can be made.

In Fig. 7A, only the YOLOv3-tiny and MRD-YOLO detections were accurate. The other models either interpreted melons truncated by leaves as two or more separate melons or were affected by occlusion, resulting in prediction boxes containing only some of the melons. In Fig. 7B, only MRD-YOLO accurately predicted the location and category of four unripe melons, including two targets that were particularly challenging to detect. In Fig. 7C, where the overlap problem occurred, YOLOv3-tiny, YOLOv5n, YOLOR, and YOLOv7-tiny all produced inaccurate detection results. On the other hand, in Fig. 7D, only MRD-YOLO, YOLOv5n and YOLOv7-tiny made correct predictions. A target in Fig. 7E evaded detection by all models due to its extreme color similarity to the leaf and being obscured by shadows.

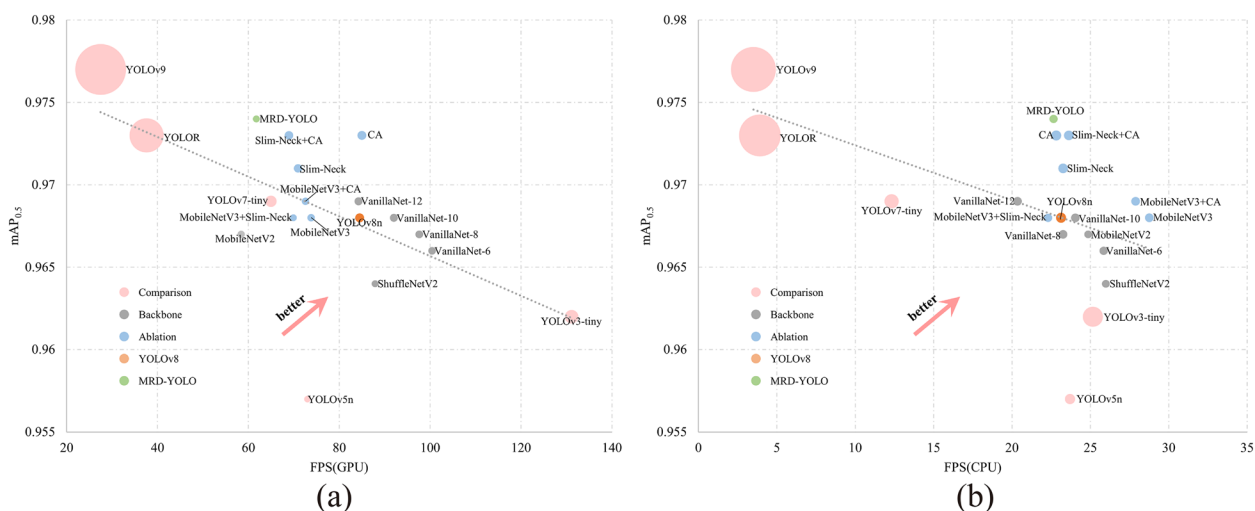
These findings suggest that the proposed lightweight MRD-YOLO demonstrates high accuracy and robustness, effectively addressing the challenges of melon ripeness detection tasks under complex natural environmental conditions.

**Summary of ablation and comparison experiments**

In this section, we provide a comprehensive comparison of 19 different models in the ablation and comparison experiments. In Fig. 8, models favoring the upper right corner indicate superior detection performance. Smaller circle sizes correspond to fewer parameters or computations.

It is evident that all our improvements to the baseline YOLOv8 have yielded promising results, as evidenced by our improved models occupying the top of overall performance, denoted by the distinctive blue circles. Our improved models exhibit a more pronounced advantage in CPU environments. While in GPU environments, MRD-YOLO does not demonstrate an inference speed advantage over other models, it nonetheless maintains high accuracy with minimal computational load. In comparison to GPU environments, MRD-YOLO exhibits the best detection performance in CPU settings. Models closer to MRD-YOLO significantly lag behind in terms of parameter count and computational complexity.

In conclusion, our proposed lightweight model demonstrates excellent performance in the task of melon ripeness detection, achieving exceptionally high detection accuracy while preserving real-time inference speed and minimizing both parameter count and computational demands. The advantages of our improved MRD-YOLO model are particularly evident in computationally constrained environments. These findings suggest promising application prospects for our model in resource constrained mobile and edge devices, such as picking robots or smartphones.



**Fig. 8** Comparison of mAP and inference speed with batch size 1. GPU: Tesla V100S-PCIE. CPU: Intel® Xeon® Gold 6230R. The size of the circle is related to the parameters of each method in (a) and the FLOPs of each method in (b), respectively

**Generalization experiment**

Generalization in deep learning refers to the ability of a trained model to perform well on unseen data or data on which it hasn't been explicitly trained. It is a crucial aspect because a model that merely memorizes the training data without learning underlying patterns will not perform well on new, unseen data.

To assess the generalization ability of the MRD-YOLO model, we utilized images from two melon datasets with a total of 132 images obtained from Roboflow as our test set [17, 18]. Since the acquired datasets consist of only one category, melon class, we labeled them accordingly. None of the data in this dataset were involved in the model training process, and the varieties of the melons in these datasets are entirely different from those in our dataset. Despite the relatively poor resolution and quality of the images in the dataset, it presents an excellent opportunity to evaluate the generalization ability of our MRD-YOLO model.

Due to the dataset's imbalanced categories and the limited number of ripe melons, we present the experimental results for both ripe and unripe categories. The results are illustrated in Table 7. MRD-YOLO achieves a precision of 89.5% in recognizing unripe melons, with a mAP of 88.6%. Additionally, for ripe melons, the mAP reaches 83.1%. These results underscore the MRD-YOLO model's robust generalization ability and its proficiency

in handling unfamiliar datasets. The detection outcomes of the generalization experiments are depicted in Fig. 9. The model exhibits strong detection capabilities for melons with occlusions, overlaps, and scale differences. Furthermore, it accurately distinguishes ripe and unripe melons, even when their colors are similar. This robustness is attributed to the inclusion of numerous images with complex backgrounds in our original training set, enabling the MRD-YOLO to effectively adapt to diverse real-world scenarios.

In conclusion, the results of the generalization experiments reaffirm the effectiveness of MRD-YOLO in detecting melon ripeness in authentic agricultural environments.

**Discussion**

**Difference in inference speed between GPU and CPU environments**

MRD-YOLO exhibits a significant improvement in both parameter count and floating-point operations when compared to the baseline model. In theory, a smaller model would typically imply faster inference. However, the experimental findings contradict this expectation. Instead, the enhanced MRD-YOLO exhibited a reduction in inference speed within the GPU environment, yet maintained remarkable performance within the CPU environment.

The main reason for this phenomenon lies in the network architecture of MobileNetV3, originally designed for mobile CPUs. MobileNetV3 employs depth-separable convolutions, which decompose a standard convolution into two stages (depthwise convolution followed by pointwise convolution), thereby reducing the number of parameters and total computation required. Consequently, the resulting model exhibits fewer parameters and reduced computational demands,

**Table 7** Performance of MRD-YOLO in generalization experiment

Class	Instances	P(%)	R(%)	mAP(%)
Unripe	373	89.5	81.8	88.6
Ripe	63	81.2	82.4	83.1
All	436	85.3	82.1	85.9



**Fig. 9** Detection results of MRD-YOLO in generalization experiment

albeit with an increased number of layers [20]. GPUs excel in the parallel processing of large datasets, making them well-suited for intensive deep learning tasks. However, despite their parallel computing prowess, GPUs incur overhead in initiating and coordinating parallel tasks. If the computational workload of MobileNetV3 does not fully leverage GPU parallelism, this overhead may offset the benefits of GPU acceleration. Conversely, CPUs efficiently handle smaller workloads without significant parallelism overhead. MobileNetV3's architectural optimizations align more closely with the characteristics of CPU environments, leading to superior performance. The disparity in inference speed between MobileNetV3 isn't limited to GPU versus CPU scenarios; compared to GPUs, which offer substantial arithmetic power, MobileNetV3 may perform well on hardware with limited arithmetic precision. These attributes make MobileNetV3 an ideal candidate for lightweight models tailored for mobile and edge devices.

The advantages of MobileNetV3 are particularly pronounced in real-world settings where not all devices possess powerful GPUs. Therefore, we incorporated MobileNetV3 into our baseline model. The enhanced lightweight MRD-YOLO exhibits promising prospects for applications in resource constrained environments and achieves exceptional performance in the task of melon ripeness detection.

### Visualization of MRD-YOLO and YOLOv8

To more intuitively showcase the improvement effect of our proposed method over the baseline, we visualized the detection results of YOLOv8n and MRD-YOLO using the Grad-CAM tool [41]. Grad-CAM is a deep learning technique employed for visualizing the regions of an image crucial for predicting a particular class. It offers valuable insights into the decision-making process of Convolutional Neural Networks, aiding in the understanding of which parts of an image influence the network's predictions. The Grad-CAM visualization results are depicted in Fig. 10.

The baseline YOLOv8n exhibits a more dispersed area of interest and is more susceptible to interference from non-melon areas, such as leaves. In contrast, the enhanced MRD-YOLO model demonstrates superior capability in focusing on melon features and accurately localizing each target, even in scenarios with multiple targets within a single image. The integration of the Coordinate Attention mechanism in the improved model enables it to concentrate on key melon features. This further validates the effectiveness of our proposed MRD-YOLO model for melon ripeness detection tasks.

### Prospects and limitation

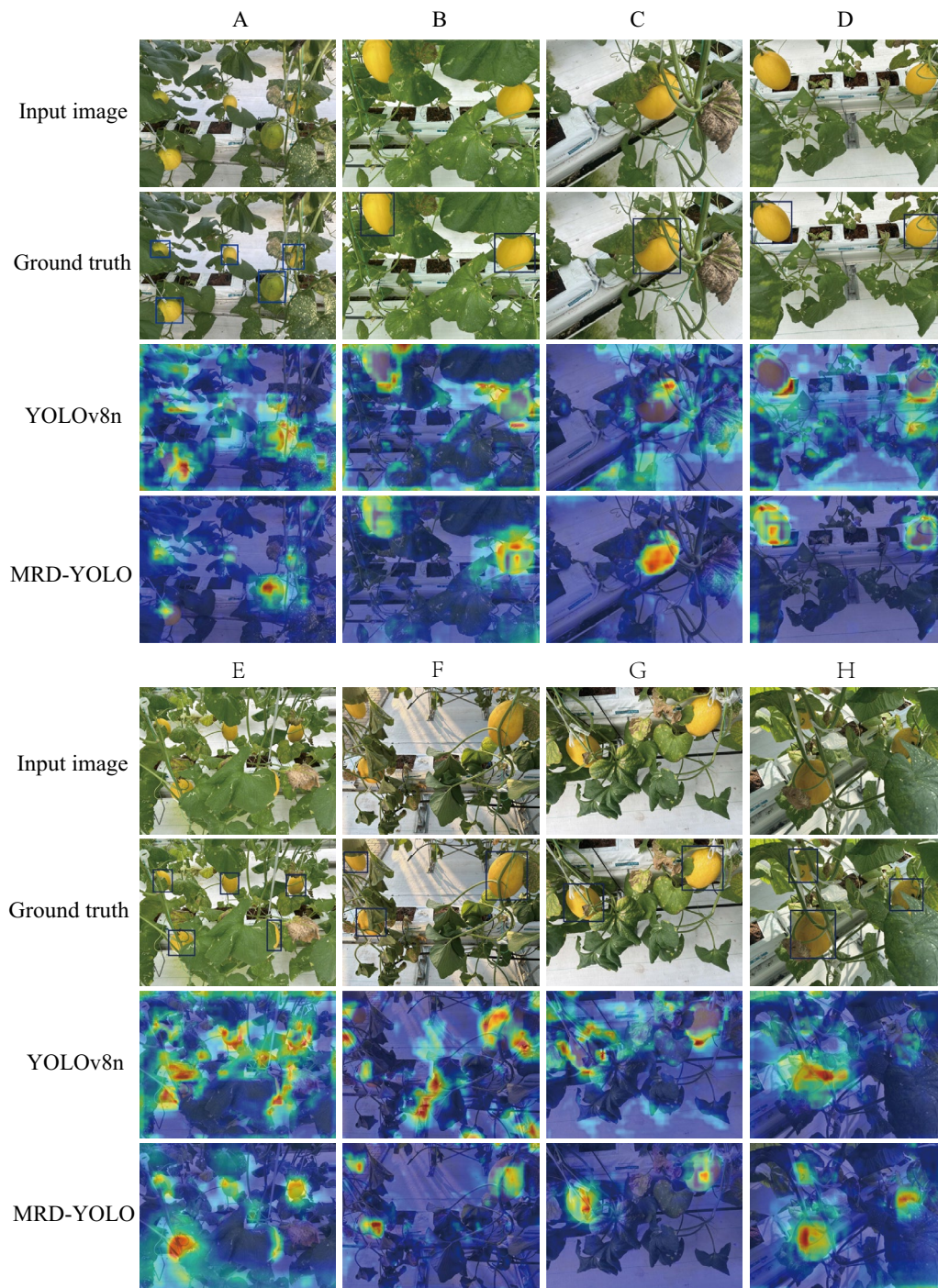
In this study, we have created a large scale, meticulously labeled, and classified melon dataset based on ripeness. This dataset was collected from a real greenhouse environment, encompassing prevalent challenges encountered in the field, including occlusion, overlapping, and variations in light intensity. Such complexities in the dataset provide authentic and comprehensive data for training object detection algorithms. Our dataset exhibits a broad spectrum of applications, particularly in phenotype detection.

For future endeavors, we propose a more nuanced categorization of ripeness to offer detailed insights into the harvesting process. Moreover, unlike existing melon datasets, the melon instances in our dataset exhibit sharp focus, facilitating instance segmentation. This feature enables us to conduct more profound investigations into the phenotypes of individual melons. Beyond phenotype detection, predicting attributes such as fruit ripeness or color using methods like time series analysis and generative networks emerges as a pivotal research avenue. Our dataset encompasses individual melons across various growth stages, exhibiting substantial disparities in size, color, and ripeness. The extensive dataset size provides ample support for diverse future studies.

The images we gathered exhibit minimal blurring, this circumstance raises concerns regarding the potential limitations of MRD-YOLO in detecting targets that are both small and blurred. Originally tailored for discerning the ripeness of melons, our model relies on clear images to accumulate sufficient data for distinguishing between targets at various ripeness stages. Consequently, the majority of images in our initial training dataset are meticulously focused, resulting in a dearth of small, low-resolution targets. This deficiency contributes to the model's inability to detect small and blurred objects in low-quality images, thereby reducing the recall rate in generalization experiments.

### Conclusion

Detecting melon ripeness using object detection algorithms poses a significant challenge owing to the scarcity of high-quality training datasets and the inherent complexity of the field environment. This study aims to bridge the gap in high-quality datasets for melon ripeness detection by establishing a comprehensive dataset and proposes a lightweight ripeness detection method, MRD-YOLO, suitable for resource constrained environments. The method utilizes MobileNetV3 as the backbone network, reducing the model size and improving its inference efficiency in computationally constrained environments. The integration of a Slim-neck design



**Fig. 10** Visualization results of MRD-YOLO and YOLOv8 using Grad-CAM

paradigm further optimizes the model, ensuring detection accuracy while minimizing parameters and computations. Additionally, we introduce Coordinate Attention to enhance the model’s ability to detect targets of different ripeness levels in complex scenarios. Experimental validations in real field environments demonstrate

the superior performance of MRD-YOLO compared to existing state-of-the-art methods. Its lightweight design makes it promising for deployment in various resource constrained field devices. Furthermore, MRD-YOLO exhibits efficient generalization capabilities, as evidenced by testing on two datasets sourced from Roboflow. In

conclusion, this study presents an efficient method for melon ripeness detection, offering valuable data, technical insights, and references for ripeness detection across not only different melon varieties but other fruits as well.

#### Acknowledgements

Not applicable.

#### Author contributions

XJ and YW conceived the idea, designed the experiments, performed field data acquisition, conducted experiments and wrote the manuscript. XJ conducted validation and visualization. DL supervised the project and revised the manuscript. WP supervised the project, contributed funding assistance and revised the manuscript.

#### Funding

This work has been supported by Basic Research Programs of Shanxi Province (202303021211069), National Natural Science Foundation of China (grant no. 32100501), Shenzhen Science and Technology Program (grant no. RCBS20210609103819020) and Innovation Program of Chinese Academy of Agricultural Sciences.

#### Availability of data and materials

The weight of the MRD-YOLO model, original datasets, log files during the experiments and source codes are in GitHub (<https://github.com/XuebinJing/Melon-Ripeness-Detection>). The melon datasets from the Roboflow are available in <https://universe.roboflow.com/test-ai/melon-ai>, <https://universe.roboflow.com/test-ai/melon-1.6-t5wxh>. Information on the varieties of melon involved in the original dataset can be obtained by visiting the Variety Validation Information Query website of Guangxi Province, China.

#### Declarations

#### Ethics approval and consent to participate

Not applicable.

#### Consent for publication

All authors have consented to the publication of this manuscript.

#### Competing interests

The authors declare no Conflict of interest.

#### Author details

<sup>1</sup>College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China. <sup>2</sup>Shenzhen Branch, Guangdong Laboratory of Lingnan Modern Agriculture, Genome Analysis Laboratory of the Ministry of Agriculture and Rural Affairs, Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen 518120, China.

Received: 15 June 2024 Accepted: 8 August 2024

Published online: 16 August 2024

#### References

- Sun M, Zhang D, Liu L, Wang Z. How to predict the sugariness and hardness of melons: a near-infrared hyperspectral imaging method. *Food Chem*. 2017;218:413–21. <https://doi.org/10.1016/j.foodchem.2016.09.023>.
- Laur LM, Tian L. Provitamin A and vitamin C contents in selected California-grown cantaloupe and honeydew melons and imported melons. *J Food Comp Anal*. 2011;24(2):194–201. <https://doi.org/10.1016/j.jfca.2010.07.009>.
- Kader AA. Fruit maturity, ripening, and quality relationships (1999). <https://api.semanticscholar.org/CorpusID:89021673>.
- Prasad K, Jacob S, Siddiqui MW. Chapter 2 - Fruit Maturity, Harvesting, and Quality Standards. In: Siddiqui MW, editor. *Preharvest Modulation of Postharvest Fruit and Vegetable Quality*. Academic Press; 2018. p. 41–69. <https://www.sciencedirect.com/science/article/pii/B978012809807300020>
- Verma L, Joshi V. Post-harvest technology of fruits and vegetables. *Post Harvest Technol Fruits Veg*. 2000;1:1–76.
- Doerflinger FC, Rickard BJ, Nock JF, Watkins CB. An economic analysis of harvest timing to manage the physiological storage disorder firm flesh browning in 'Empire' apples. *Postharvest Biol Technol*. 2015;107:1–8. <https://doi.org/10.1016/j.postharvbio.2015.04.006>.
- Taniwaki M, Takahashi M, Sakurai N. Determination of optimum ripeness for edibility of postharvest melons using nondestructive vibration. *Food Res Int*. 2009;42(1):137–41. <https://doi.org/10.1016/j.foodres.2008.09.007>.
- Taniwaki M, Tohro M, Sakurai N. Measurement of ripening speed and determination of the optimum ripeness of melons by a nondestructive acoustic vibration method. *Postharvest Biol Technol*. 2010;56(1):101–3. <https://doi.org/10.1016/j.postharvbio.2009.11.007>.
- Yang S, Tian Q, Wang Z, Guo W. Relationship between optical properties and internal quality of melon tissues during storage and simulation-based optimization of spectral detection in diffuse reflectance mode. *Postharvest Biol Technol*. 2024;213:112935. <https://doi.org/10.1016/j.postharvbio.2024.112935>.
- Ripardo Calixto R, Pinheiro Neto LG, da Silveira Cavalcante T, Nascimento Lopes FG, Ripardo de Alexandria A, de Oliveira Silva E. Development of a computer vision approach as a useful tool to assist producers in harvesting yellow melon in Northeastern Brazil. *Comput Electron Agric*. 2022;192:106554. <https://doi.org/10.1016/j.compag.2021.106554>.
- Ni X, Li C, Jiang H, Takeda F. Deep learning image segmentation and extraction of blueberry fruit traits associated with harvestability and yield. *Horticulture Res*. 2020. <https://doi.org/10.1007/s11119-022-09895-2>.
- Halstead M, McCool C, Denman S, Perez T, Fookes C. Fruit quantity and ripeness estimation using a robotic vision system. *IEEE Robot Automation Lett*. 2018;3(4):2995–3002. <https://doi.org/10.1109/LRA.2018.2849514>.
- Wan P, Toudeshki A, Tan H, Ehsani R. A methodology for fresh tomato maturity detection using computer vision. *Comput Electron Agric*. 2018;146:43–50. <https://doi.org/10.1016/j.compag.2018.01.011>.
- Tu S, Xue Y, Zheng C, Qi Y, Wan H, Mao L. Detection of passion fruits and maturity classification using Red-Green-Blue depth images. *Biosyst Eng*. 2018;175:156–67. <https://doi.org/10.1016/j.biosystemseng.2018.09.004>.
- Chen S, Xiong J, Jiao J, Xie Z, Huo Z, Hu W. Citrus fruits maturity detection in natural environments based on convolutional neural networks and visual saliency map. *Precis Agric*. 2022;23(5):1515–31. <https://doi.org/10.1007/s11119-022-09895-2>.
- Wang A, Qian W, Li A, Xu Y, Hu J, Xie Y, et al. NVW-YOLOv8s: an improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages. *Comput Electron Agric*. 2024;219:108833. <https://doi.org/10.1016/j.compag.2024.108833>.
- Ai T. Melon AI Dataset [Open Source Dataset]. Roboflow. <https://universe.roboflow.com/test-ai/melon-ai>.
- Ai T. Melon 1.6 Dataset [Open Source Dataset]. Roboflow. <https://universe.roboflow.com/test-ai/melon-1.6-t5wxh>.
- Jocher G, Chaurasia A, Qiu J. Ultralytics YOLO. <https://github.com/ultralytics/ultralytics>
- Howard G, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for mobilenetv3. In: *Proceedings of the IEEE/CVF international conference on computer vision*; 2019. p. 1314–1324.
- Li H, Li J, Wei H, Liu Z, Zhan Z, Ren Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles; 2022. arXiv preprint [arXiv:2206.02424](https://arxiv.org/abs/2206.02424).
- Hou Q, Zhou D, Feng J. Coordinate Attention for Efficient Mobile Network Design. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2021. p. 13708–13717.
- Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely Connected Convolutional Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2017. p. 2261–2269.
- Lee Y, Hwang Jw, Lee S, Bae Y, Park J. An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; 2019. p. 752–760.
- Wang CY, Mark Liao HY, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; 2020. p. 1571–1580.



26. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–4520.
27. Ma N, Zhang X, Zheng HT, Sun J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. *Comput Vis ECCV 2018*. Cham: Springer International Publishing; 2018. p. 122–38.
28. Chen H, Wang Y, Guo J, Tao D. VanillaNet: the Power of Minimalism in Deep Learning.
29. Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. p. 7132–7141.
30. Woo S, Park J, Lee JY, Kweon IS. CBAM: Convolutional Block Attention Module. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. *Computer Vision - ECCV 2018*. Cham: Springer International Publishing; 2018. p. 3–19.
31. Liu Y, Shao Z, Hoffmann N. Global attention mechanism: retain information to enhance channel-spatial interactions.
32. Liu H, Liu F, Fan X, Huang D. Polarized self-attention: towards high-quality pixel-wise regression.
33. Liu Y, Shao Z, Teng Y, Hoffmann N. NAM: Normalization-based attention module.
34. Zhang QL, Yang YB. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2021. p. 2235–2239.
35. Yang L, Zhang RY, Li L, Xie X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In: Meila M, Zhang T, editors. *Proceedings of the 38th International Conference on Machine Learning*. vol. 139 of *Proceedings of Machine Learning Research*. PMLR; 2021. p. 11863–11874. <https://proceedings.mlr.press/v139/yang21o.html>.
36. Adarsh P, Rathi P, Kumar M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); 2020. p. 687–694.
37. Jocher G. YOLOv5 by ultralytics. <https://github.com/ultralytics/yolov5>.
38. Wang CY, Yeh IH, Liao HYM. You only learn one representation: unified network for multiple tasks.
39. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.
40. Wang CY, Yeh IH, Liao HYM. YOLOv9: learning what you want to learn using programmable gradient information.
41. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017. p. 618–626.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.