Journal of Agricultural Engineering

https://www.agroengineering.org/

Tomato detection in natural environment based on improved YOLOv8 network

Wancheng Dong,[1] Yipeng Zhao,[1] Jiaxing Pei,[1] Zuolong Feng,[2] Zhikai Ma,[3] Leilei Wang,[1] Simon Shemin Wang[1]

1.School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan
2. Hebei Provincial Agricultural Mechanization Technology Promotion Station, Langfang
3. College of Mechanical and Electrical Engineering, Hebei Agricultural University, Handan, China

Corresponding author: Leilei Wang, School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan, 056038, China. E-mail: wangll@hebeu.edu.cn

Please cite this article as doi: 10.4081/jae.2025.1732

Submitted: 12 February 2025
Accepted: 15 July 2025

Tomato detection in natural environment based on improved YOLOv8 network

Wancheng Dong,[1] Yipeng Zhao,[1] Jiaxing Pei,[1] Zuolong Feng,[2] Zhikai Ma,[3] Leilei Wang,[1] Simon Shemin Wang[1]

1.School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan
2. Hebei Provincial Agricultural Mechanization Technology Promotion Station, Langfang
3. College of Mechanical and Electrical Engineering, Hebei Agricultural University, Handan, China

Correspondence: Leilei Wang, School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan, 056038, China. E-mail: wangll@hebeu.edu.cn

Conflict of interest: The authors declare that they have no known competing financial interests of this paper.

Funding: this work was supported by the Hebei Province Modern Agricultural Industry Technology System Facility Vegetable Innovation Team Agricultural Machinery and Intelligent Equipment Post Expert Project [HBCT2023100213].

Abstract

In this paper, an improved lightweight YOLOv8 method is proposed to detect the ripeness of tomato fruits, given the problems of subtle differences between neighboring stages of ripening and mutual occlusion of branches, leaves, and fruits. The method replaces the backbone network of the original YOLOv8 with a more lightweight MobileNetV3 structure to reduce the number of parameters of the model; at the same time, it integrates the convolutional attention mechanism module (CBAM) in the feature extraction network, which enhances the network's capability of extracting features of tomato fruits. At the same time, it introduces the SCYLLA-IoU (SIoU) as a bounded YOLOv8 frame regression loss function, effectively solving the mismatch problem between the predicted frame and the actual frame and improving recognition accuracy. Compared with the current mainstream models Resnet50, VGG16, YOLOv3, YOLOv5, YOLOv7, *etc*., the model is in an advantageous position regarding precision rate, recall rate, and detection accuracy. The research and experimental results show that the mean values of precision, recall rate, and average precision of the improved MCS-YOLOv8 model under the test set are 91.2%, 90.2%, and 90.3%, respectively. The detection speed of a single image is 5.4ms, and the model occupies less memory by 8.7 M. The model has a clear advantage in both detection speed and

precision rate and also shows that the improved MCS-YOLOv8 model can provide strong technical support for tomato-picking robots in complex environments in the field.

Keywords: CBAM; loss function; MobileNetV3; ripening; tomato; YOLOv8.

Introduction

As one of the three major world trade vegetables, tomato is very popular in our country and worldwide. Currently, China plays an important role in the global tomato deep-processing industry, both as a major supplier of tomato raw materials and a major exporter of tomato products. In 2022, the tomato planting area in China reached 1,169,200 hectares, and tomato production reached about 69,707,700 tons (Sun *et al*., 2023). Tomato picking has a large workload and, at the same time, faces a variety of problems, such as high labor costs, low efficiency and labor shortage. There is an urgent need to move towards automation and intelligence. Therefore, developing an automatic ripeness detection system with high accuracy is of great practical significance for identifying the different ripening stages of tomatoes and determining their distribution range, thus realizing the automation of tomato picking (Chen *et al.,* 2023).

At present, researchers at home and abroad have made outstanding achievements in fruit ripening, and along with the rapid development of deep learning technology and target detection algorithms, the application scope of machine vision in detection and classification has been gradually expanded. Scholars at home and abroad have used machine vision to realize the detection of fruit ripeness in mangoes, passion fruit, apple, and bananas (Mulyani *et al*., 2017; Mim *et al*., 2018; Mazen and Nashat, 2019; Luo *et al*., 2024). These studies have significantly improved the detection accuracy and efficiency by improving the target detection model. For example, Luo *et al*. (2024) proposed a model for strawberry fruit recognition in complex environments by enhancing the YOLOv8 neck network using methods such as SPD-Conv. The model achieves 93.5% recognition accuracy and 86.0% recall for strawberry fruits in complex scenes. Regarding computational efficiency, the model takes 17.2ms to detect a single strawberry image in a GPU environment, and the model

volume is reduced to 66% of the original size after lightweight processing. Qiu *et al*. (2024) developed an accurate recognition model for dragon fruit fruits at different ripening stages. By optimizing the backbone network of YOLOv8 and incorporating the attention mechanism, the model's performance was significantly improved, and the mAP reached 90.9%, which effectively reduces misclassification and omission in complex environments. Li *et al*. (2023) proposed a tomato fruit ripeness grading and counting model, which improves the feature diversity extraction capability by improving the backbone network and introduces the MHSA attention mechanism in the YOLOv8 backbone network. The precision and recall of the improved model reached 80.6% and 80.7%, respectively. Although the precision and recall of the model are improved over the original model, the increase in model memory is not favorable for real-time detection applications.

Zhao *et al*. (2023) proposed a lightweight, improved backbone network for RT-DETR to detect the ripeness of spike-harvested cherry tomatoes with an accuracy of 90%, which achieves fast detection (41.2 f/s). To solve the problem of low accuracy and slow detection of balsam pear recognition in unstructured environments. Tan *et al*. (2024) proposed a method based on improved YOLOv8. The method applies simSPPF to optimize the SPPF module and introduces PConv convolution and contribution weight parameters to achieve light-weighting of the detection head. The improved model has a recognition accuracy as high as 94.7%, which combines high recognition accuracy and detection speed, effectively improving the efficiency and accuracy of balsam pear recognition. Lv *et al*. (2024) proposed the green citrus ripeness detection based on the improved YOLOv5 in the field complex environment, the model through the improvement of the backbone network, and so on to realize the green citrus ripeness in the field complex environment to achieve fast and accurate detection, to provide technical support for the citrus harvesting robot in complex situations. Miao *et al*. (2023) introduced MobileNetV3 into the YOLOv7 model as the backbone feature extraction network, reducing the number of network parameters while improving the model accuracy and optimizing the model memory usage.

Although existing research has achieved significant results in fruit ripeness detection, some urgent problems still need to be solved in the field of tomato ripeness detection. Currently, there are fewer studies for multi-stage tomato ripeness recognition, and the dense

distribution of tomato fruits in complex natural environments in the field, the existence of mutual shading between fruits and branches and leaves, as well as light differences due to the shading of branches and leaves, and other problems. These problems seriously affect the detection accuracy and robustness of existing models. Therefore, developing a detection model that can efficiently and accurately recognize multi-stage tomato ripening under complex natural environments is of great practical significance.

Accordingly, this paper proposes a tomato ripeness detection method based on improved YOLOv8, which improves the backbone network of YOLOv8 to achieve light weighting, selects an efficient attention mechanism, and also improves the bounding box regression function to improve the recognition accuracy and processing speed of the model while reducing the model complexity. The results of this research provide technical solutions for tomato-picking robots to carry out picking activities in complex environments on farmland.

Image data acquisition and preprocessing

Data set collection

The images were collected from an agricultural planting base in Anyang City, Henan Province, and the image acquisition device was RedmiK60Ultra, which finally captured 3255 tomato images in JPG format with a resolution of 3072 × 4096 pixels. The typical growth state of tomato is shown in Figure 1, and its growth in the field exists a variety of complex situations such as branch and leaf shading, fruit overlapping and backlighting.

a. Shading by branches and foliage

b. Shading by branches and foliage

c. Unobstructed

d. Backlighting

Figure 1. Part of the captured image.

Image data preprocessing

In order to ensure that the tomato dataset has diversity, this study uses data enhancement methods to preprocess the captured tomato images, including mirroring horizontally, flipping vertically, randomly changing brightness, randomly changing contrast, and adding Gaussian noise. By expanding the dataset, the diversity of the images is enhanced and improved, thus improving the robustness of the model.

The images also need to be labeled, and the image dataset was labeled through the online website *Make Sense*: five categories of labels were classified according to the color of the skin and flesh, i.e., unripe, discolored, first ripe, medium ripe, and ripe, which correspond to the five classes "0", "1", "2", "3", and "4", respectively. ", "2", "3" and "4", respectively. In view of the research focus on non-destructive detection of tomato ripeness in the hanging branch state, this paper classifies the peel color change into five classes of ripeness according to the actual growth and development of tomatoes (see Table 1). Since the ripening stage of a tomato has a significant impact on its processing, storage, and transportation methods, the selection of the appropriate picking time should be based on

different purposes of use, which has a key role in maintaining the quality of the fruit and extending its shelf life.

Table 1. Description of the 5 levels of tomato maturity.

(Military) rank	Maturity level	Descriptive	Picking situation
0	Immature stage	Fruits are set in size and green in color	Fruits are not ripe and should not be picked
1	Commutation period	Starts to change color around the umbilicus, gradually appears yellow or light red	Can be harvested and stored for long distance transportation
2	Early ripe	Fruit surface red coverage has reached half, reddening rate 30% to 60%	Can be picked for artificial ripening
3	Middle ripe	The surface of the fruit is mostly red, with 60% to 90% reddish coloration.	Ready to pick and sell fresh every other day
4	Mature	More than 90% of the surface of the fruit is red	Available for picking and same day sale

Analysis of data sets

For the dataset augmented and expanded with 19530 images, it is divided into three mutually independent classifications of the training set, test set, and validation set according to 8:1:1. There are 15624 images in the training set, 1953 images in the testing set and 1953 images in the validation set after the division.

In this dataset, each image has more than one label, with 8,810 immature labels, 3,990 labels at the color change stage, 4,760 labels at the first maturity stage, 9,670 labels at the medium maturity stage, and 4,410 labels at the ripening stage.

YOLOv8 target detection algorithm

In this paper, we use the YOLOv8 algorithm; after a long development period, the YOLO version has now been updated and iterated to YOLOv10 (Wang *et al*., 2024), and YOLOv8 has a relatively lightweight network structure among many versions. The algorithm, released by Ultralytics in 2023, has high accuracy and fast inference speed, making it one of the best

target detection models available today (Ge *et al*., 2021). This makes it more widely used in target detection and applicable to several vision tasks such as instance segmentation, target tracking, pose estimation and image classification. It provides five versions of YOLOv8n/s/m/l/x. Among them, YOLOv8s is specially optimized to fit NVIDIA Jetson Nanodevices. It has a balance of high accuracy and speed in embedded deployment scenarios, so YOLOv8s is used in this paper for tomato detection.

The YOLOv8 model consists of 4 parts: input end, backbone network, neck network and head network. The mosaic data enhancement technique is used on the input side (Tian *et al*., 2024). In addition to this, YOLOv8 introduces more diverse enhancement strategies to enrich the diversity of the training set by mixing different images or cutting and pasting image blocks and dynamically adjusts the enhancement strategies according to the performance during the training process, which further improves the ability to detect small targets. The backbone layer consists of 5 convolutional blocks, 4 C2f blocks, and 1 SPPF block, which extract the features from the image. Compared with YOLOv5, YOLOv8 adopts a lighter-weight C2 block instead of a C3 block while maintaining the advantages of the CSP network structure, effectively reducing the computational cost without sacrificing the quality of feature extraction (Xu *et al*., 2024). Regarding the feature fusion layer, Neck partially realizes the fusion of different layers of features through a path aggregation network combined with a feature pyramid network (PAN-FPN) (Roy *et al*., 2022). This design enhances the model's ability to handle diverse and complex tasks. Moreover, it enhances the global performance of target detection. The Head layer is responsible for the final target detection and classification task. The original coupled head structure is discarded, and the current mainstream decoupled head structure is adopted to handle the target detection and classification tasks separately. This makes the Head part more flexible and, at the same time, enables the model to better adapt to different detection tasks.

Improvements to the YOLOv8 algorithm

Detection of targets in complex environments in the field, both to ensure the rapidity of tomato fruit ripeness classification detection and to maximize the accuracy of the network for fruit recognition.YOLOv8, as a single-stage target detector, can be directly divided into a

grid on the image, and the detection task is regarded as a regression task to achieve end-to-end training. In this paper, the following improvements are made based on the YOLOv8 model:

(1) Replace the original model backbone network with the more lightweight MobileNetV3 structure.

(2) Introducing CBAM (convolutional block attention module) attention mechanism in the backbone network.

(3) Replace the bounding box loss function in the original model with a new loss function SIoU.

The improved network is named MCS-YOLOv8 target detection network, which is mainly composed of four parts: the input layer, the backbone feature extraction network, the path aggregation network, and the output layer, and its network structure is shown in Figure 2. First, the input layer receives a tomato image of 640 × 640 pixels; then, the image is fed into the backbone network for feature extraction, and the extracted tomato feature map is subsequently passed to the path aggregation network, where the shallow and deep features are effectively fused; finally, the fused feature maps are fed into the output layer to generate the prediction frames and to recognize the classes of tomatoes.

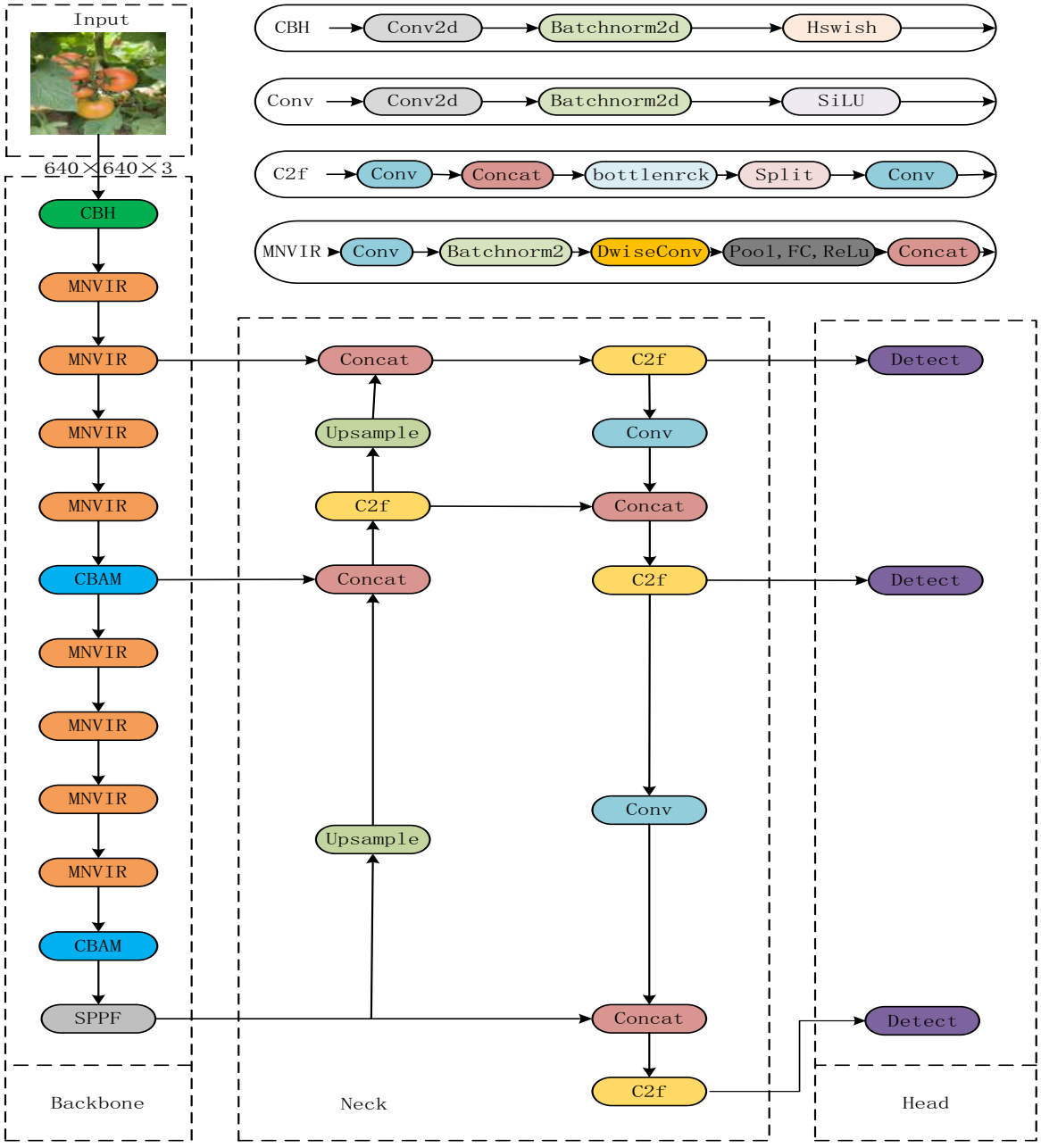

Figure 2. Diagram of the improved YOLOV8 lightweight network model. Conv2d is convolution, Concat is feature fusion by adding the number of channels, BN is batch normalization, CBAM is convolutional attention block, MNVIR stands for MNIneck, and Upsample stands for upsampling.

MobileNetV3 network architecture

Google's MobileNetV3, introduced in 2019, is a convolutional neural network optimized for mobile devices, which effectively increases the depth and nonlinearity of the network by employing lightweight depth-separable convolutional and inverse covariance

modules while reducing the amount of computation and the number of parameters for lightweight (Qian *et al*., 2021). MobileNetV3 also introduces the Hard-swish (Peng *et al*., 2023) activation function, an improvement of ReLU that enhances the model performance by enhancing the nonlinear characteristics.MobileNetV3 network combines the advantages of the previous two generations. It integrates the SE (Squeeze-and-Excitation) module on top of V2 (Sandler *et al*., 2018), and Depthwise Convolution (abbreviated as DwiseConv) is one of the key components in the MobileNetV3 family of models. It significantly reduces the computational complexity and the number of parameters by decomposing the standard convolution process into two steps while maintaining the model performance as much as possible. First, deep convolution applies a separate convolution kernel for each input channel, meaning convolution operations are performed independently for each input channel. This effectively reduces the large number of redundant computations processed simultaneously across all input channels in traditional convolutional methods. Second, point-by-point convolution further processes the results of the deep convolution using a 1x1-sized convolution kernel. The primary purpose of this step is to adjust the number of output channels and integrate the different feature mappings generated by the deep convolution. In this way, point-by-point convolution enables the exchange of information between the different channels and allows precise control of the output dimension. This design allows MobileNetV3 to significantly reduce the computational requirements and model size while ensuring model accuracy and performance, particularly suitable for mobile devices and embedded systems application requirements. Its network structure parameters are shown in Table 2.

The columns in the table represent the input vector size of the feature layer, the type of operation performed, the number of lift and output channels in the Bottleneck inverse residual structure, whether the SE module is included or not, the type of activation function used, and the step size of the convolution. Figure 3 shows the bneck structure in MobileNetV3. The operations in the network include ordinary convolution (Conv2d), inverse residual structure (benck, i.e., inverted residual blocks with linear bottlenecks), and pooling layer (pool). The processing flow of MobileNetV3 is as follows: the input image is first preprocessed by a 1×1 convolution and batch normalization, and then feature extraction is

performed by the inverted residual structure with dimensionality reduction to reduce the amount of computation. Then, the feature map is globally pooled by the SE module, and finally, the Hard-Swish activation function is used instead of Swish to reduce the computation further and improve the performance. This design allows MobileNetV3 to improve accuracy and operation speed while keeping the model lightweight. MobileNetV3 has two versions, MobileNetV3-large and MobileNetV3-small, which suit different application requirements. The paper adopts the MobileNetV3-small network structure as the backbone network of the YOLOv8 model for feature extraction.

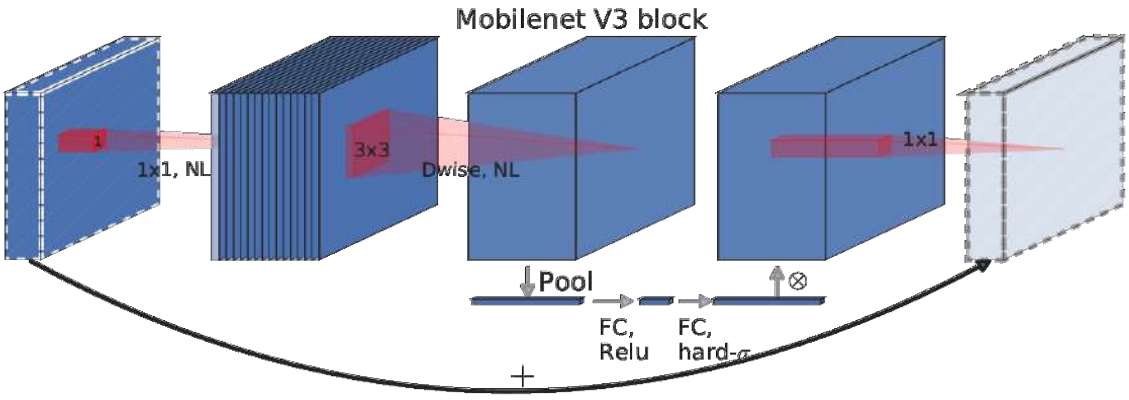

Figure 3. The bneck structure in MobileNetV3.

Table 2. MobileNetV3 network parameters.

Input	Operator	Exp size	out	SE	NL	s
$224^2 \times 3$	conv2d, 3×3	-	16	-	HS	2
$112^2 \times 16$	bneck,3×3	16	16	√	RE	2
$56^2 \times 16$	bneck,3×3	72	24	-	RE	2
$28^2 \times 24$	bneck,3×3	88	24	-	RE	1
$28^2 \times 40$	bneck,5×5	96	40	√	HS	2
$14^2 \times 40$	bneck,5×5	140	40	√	HS	1
$14^2 \times 40$	bneck,5×5	140	40	√	HS	1
$14^2 \times 40$	bneck,5×5	120	48	√	HS	1
$14^2 \times 48$	bneck,5×5	144	48	√	HS	1
$14^2 \times 48$	bneck,5×5	288	96	√	HS	2
$7^2 \times 96$	bneck,5×5	576	96	√	HS	1
$7^2 \times 96$	bneck,5×5	576	96	√	HS	1
$7^2 \times 96$	conv2d,1×1	-	576	√	HS	1
$7^2 \times 576$	pool,7×7	-	-	-	-	1
$1^2 \times 576$	conv2d 1×1,NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1×1,NBN	-	k	-	-	1

"√" indicates that the SE module has been introduced in this layer, and "×" indicates that the SE module has not been introduced in this layer.

Mechanisms for increasing attention

In tomato greenhouses, each tomato plant grows crosswise, and there is a certain degree of occlusion between fruits and fruits and between fruits and branches and leaves, which leads to the target detection algorithm not being able to correctly recognize tomato ripeness, so the convolutional attention mechanism module is added to the YOLOv8 model to dynamically adjust the input image features in order to obtain a higher recognition accuracy.

The convolutional attention mechanism module (CBAM) is an attention module that combines the channel attention mechanism and the spatial attention mechanism (Long *et al*., 2023). CBAM is a lightweight generalization model (Woo *et al*., 2018) , which removes a large number of convolutional structures from its interior and retains only a small number of pooling layers and feature fusion operations. This design reduces the heavy burden brought by convolutional computation, thus reducing the complexity and computation of the module. Meanwhile, due to the simple and flexible mechanism of CBAM, it is very versatile and can be seamlessly integrated with any CNN structure for many different neural network structures. Due to the high efficiency of CBAM, it can be less computationally

intensive for the network.

The operation flow of the tomato feature map in CBAM is shown in Figure 4. The specific process of the operation flow of the tomato feature map in CBAM is divided into: first, the global maximum pooling and global average pooling of the tomato feature map are performed respectively, and the feature mapping generates two sets of feature representations with different dimensions by compressing them in two dimensions.

After pooling, the feature maps share a multilayer perceptual network that is first downscaled by a 1×1 convolutional kernel and subsequently upscaled by another 1×1 convolutional kernel. The two tomato feature maps are merged and stacked using the layers.add() function, and the feature map weights for each channel are normalized by a sigmoid activation function. The normalized weights are multiplied with the input feature maps, followed by spatial domain processing of the feature maps after processing by the channel attention mechanism. Specifically, the feature maps are subjected to maximum pooling and average pooling in the channel dimension, respectively, and then these two output feature maps are stacked in the channel dimension using the layers.concatenate() function. After that, the number of channels is adjusted by 1×1 convolution, and the weights are normalized by the sigmoid function. The normalized weights are multiplied by the input feature map. After the input feature map passes through the channel attention mechanism, the obtained weights are multiplied by the input feature map and then sent to the spatial attention mechanism. Finally, the normalized weights are multiplied by the input feature map of the spatial attention mechanism to get the final feature map.

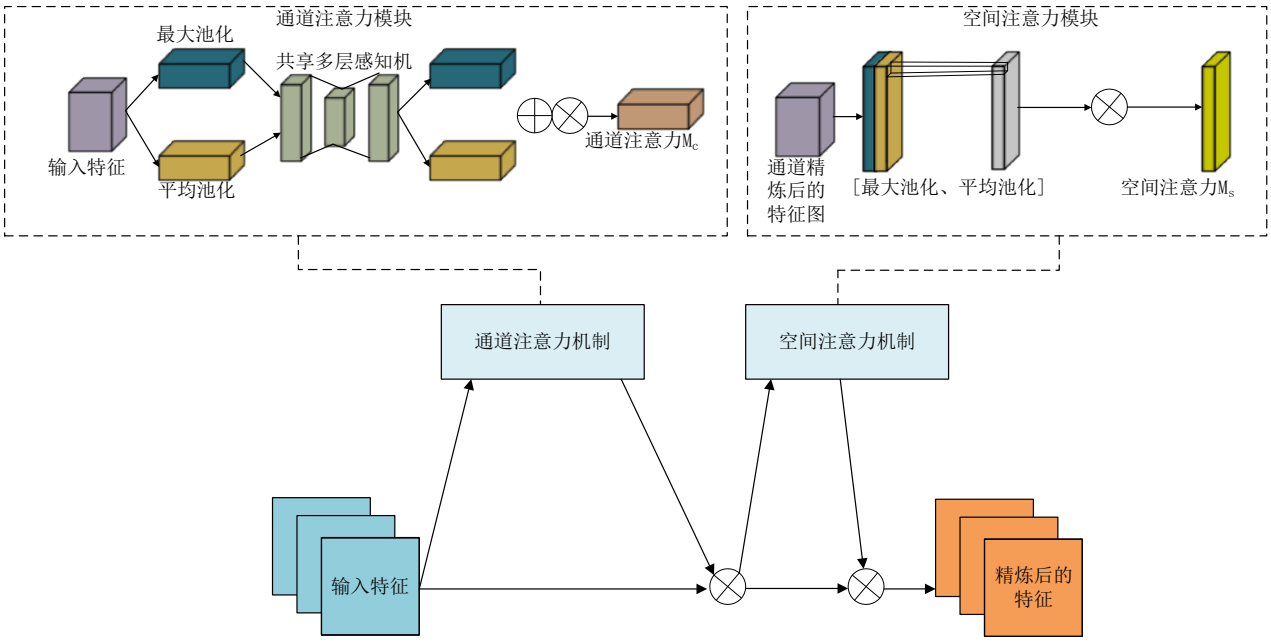

Figure 4. Convolutional attention mechanism module,

Preferred loss function

Traditional loss functions for target detection algorithms such as GIoU, ICIoU, and CIoU used in YOLOv8 (*Rezatofighi et al.*, 2019; Wang *et al.*, 2021; Zheng *et al.*, 2021) etc., mainly focus on regression metrics such as the distance between the predicted frame and the real frame, the overlap region and the aspect ratio. However, these methods ignore the orientation mismatch between the predicted and real frames. This limitation may lead to the instability of the prediction frames during the model training process, which affects the convergence speed and detection efficiency, making the final model perform poorly. In order to solve this problem, and also for must-change overfitting, this study proposes the SIoU loss function to replace the traditional loss function. The SIoU loss function consists of four components: angular loss, distance loss, shape loss, and intersection-parallel ratio loss.

(1) Angle loss refers to the deviation between the angle predicted by the model and the actual angle. Its calculation formula is shown below:

$$\Lambda = 1 - 2\sin^2(\arcsin(x) - \frac{\pi}{4}) \qquad \text{(Eq. 1)}$$

$$x = \frac{c_h}{\sigma} = \sin(\alpha) \quad \text{(Eq. 2)}$$

$$\sigma = \sqrt{(b_{c_x}^{gt} - b_{c_x})^2 + (b_{c_y}^{gt} - b_{c_y})^2} \quad \text{(Eq. 3)}$$

$$c_h = \max(b_{c_y}^{gt}, b_{c_y}) - \min(b_{c_y}^{gt}, b_{c_y}) \quad \text{(Eq. 4)}$$

Where Λ is the angular loss, $(\mathrm{b}_{c_x}^{gt}, b_{c_y}^{gt})$ denotes the center coordinate of the real frame, and (b_{c_x}, b_{c_y}) denotes the coordinates of the center point of the predicted frame.

(2) Distance loss is the deviation between the predicted and actual position of the model, which is calculated as shown below:

$$\Delta = 2 - e^{-\gamma P_x} - e^{-\gamma P_y} \quad \text{(Eq. 5)}$$

$$\rho_x = (\frac{b_{c_x}^{gt} - b_{c_x}}{c_{wb}})^2, \rho_y = (\frac{b_{c_y}^{gt} - b_{c_y}}{c_{hb}})^2, \gamma = 2 - \Lambda \quad \text{(Eq. 6)}$$

Where Δ denotes the distance loss, c_{wb} , c_{hb} are the minimum outer rectangle width and height of the real and predicted frames.

(3) Shape loss measures the similarity between the target area predicted by the gain/loss model and the real target shape. The calculation formula is shown below:

$$\Omega = \sum_{t=w,h} (1 - e^{-\omega_t})^{\theta} \quad \text{(Eq. 7)}$$

$$\omega_w = \frac{\left|\mathrm{w} - w_{gt}\right|}{\max(w, w_{gt})} \quad \text{(Eq. 8)}$$

$$\omega_h = \frac{\left|h - h_{gt}\right|}{\max(h, h_{gt})} \quad \text{(Eq. 9)}$$

Where Ω denotes the shape loss, w and h are the width and height of the predicted frame, and denote the width and height of the real frame, respectively, andθcontrols the degree of attention to the shape loss.

(4) The ratio of the intersection area of the model-predicted bounding box and the actual bounding box to their concatenation area is noted as the intersection-concatenation ratio loss, which is calculated using the formula shown below.

$$IoU = \frac{\mathrm{IntersectionA}}{\mathrm{UnionB}} \quad \text{(Eq. 10)}$$

In summary, the edge loss function of SIoU is finally defined as

$$L_{box} = 1 - IoU + \frac{\Delta + \Omega}{2} \quad \text{(Eq. 11)}$$

Test environment and assessment indicators

In this study, the hardware configurations of the testbed used for training and testing tomato data are: the CPU is Intel(R)i5-10400F@2.90GHz, the GPU is NVIDIA GeForce RTX 3060ti (8G video memory), the RAM is 16GB, the operating system is Windows 10 (64-bit), the CUDA version is 12.1, compilation platform is Pycharm, compilation language is Python 3.9, and Pytorch version is 2.2.2.

All comparison tests in this paper are conducted in the same environment. YOLOv8s was chosen as the original model for target detection. The training configuration is as follows: the image size of the network input is 640×640 pixels, the stochastic gradient descent algorithm is used (Charkroun *et al*., 2017), the initial learning rate is set to 0.01, the momentum of the SGD optimizer is 0.937, the weight decay is 0.0005, and the epochs are set to 300.

The target detection used in the paper grades tomato maturity using precision, recall, average precision (AP), mean average precision (mAP), model occupied memory, and detection speed as evaluation metrics.

Results

Improved YOLOv8 model test results

In order to validate the performance of the improved YOLOv8 model, 1,953 tomato images in the test set were tested and analyzed. Table 3 shows the detection results of this paper's algorithm for tomatoes of different ripeness levels. As can be seen from Table 3, the average precision mean of this paper's algorithm can reach 90.3%, the precision rate is 91.2%, and the recall rate is 90.2%.

Some of the detection graphic examples are shown in Figure 4, from which it can be seen that the algorithm in this paper can more accurately detect tomatoes with different ripeness for the occlusion between fruit and fruit as well as the occlusion between branches and leaves and fruits in this paper recognition effect is also better. Due to the unsynchronized fruit ripening time, there are multiple ripening stages in the image; for the fruit that has just

entered the early ripening stage and the fruit trees that have already entered the middle ripening stage, their features do not differ much, and they are easy to be confused. Therefore, the improved algorithm is able to extract the subtle features of the shape of the fruit surface in order to recognize the different ripening stages of obsolescence accurately. Meanwhile, as seen in Figure 5, the improved YOLOv8 model is also able to detect both multi-targets and light effects with better results accurately.

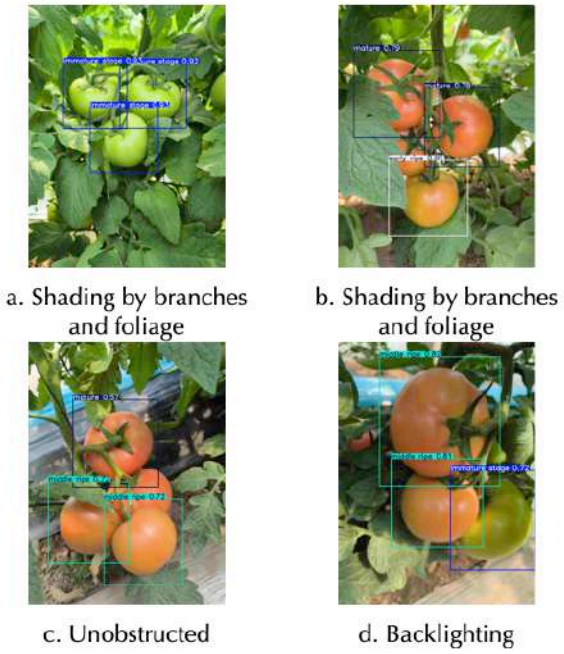

Figure 5. Improved YOLOv8 model detection results.

Table 3. Improved YOLOv8 algorithm different maturity detection results.

	Precision/%	Recall/%	mAP@0.5/%
Immature stage	90.3	89.9	87.5
Commutation period	88.7	88.5	88.7
Early ripe	90.5	89.6	89.9
Middle ripe	92.9	91.1	92.7
Mature	93.6	91.9	92.9
Mean value	91.2	90.2	90.3

Comparative analysis of different attention mechanism algorithms

Respectively, based on the YOLOv8s original model to add mainstream attention mechanisms CBAM, EMA (Ouyang *et al*., 2023), CA (Hou *et al*., 2021), ECA (Wang *et al*.,

2020), GAM (Liu *et al*., 2021). After the comparison test with the original YOLOv8s, the comparative evaluation indexes are precision, recall, and mean average precision (mAP), and the algorithmic precision under this paper's dataset after the introduction of different attention mechanisms in YOLOv8s is shown in Table 4.

Table 4. Comparative trials of different attention mechanisms.

Model	Precision / %	Recall / %	mAP / %
YOLOv8s	89.1	89.4	89.2
YOLOv8s+CBAM	90.1	90.1	90.0
YOLOv8s+EMA	89.9	89.6	89.3
YOLOv8s+CA	89.9	89.5	89.7
YOLOv8s+ECA	89.5	89.5	89.3
YOLOv8s+GAM	89.7	89.5	89.7

From this table, it can be seen that after introducing the attention mechanisms in YOLOv8s, the precision of each network is improved in different degrees compared to the original YOLOv8s network. Among the five attention mechanisms introduced, the CBAM attention mechanism has the most obvious improvement in precision, with a 1-percentage-point increase in precision and a 0.8-percentage-point increase in recall; the EMA and CA attention mechanisms have the same improvement in precision with a 0.2-percentage-point increase in recall and a 0.1-percentage-point increase in recall, respectively, and the ECA attention mechanism has a smaller improvement in precision and recall compared with the original model, with a 0.4-percentage-point increase in precision and recall, and the GAM has a smaller improvement compared with the original model with a 0.4-percentage-point increase in precision and recall. Percentage points and GAM have an improvement of 0.6 and 0.1 percentage points compared to the original model precision and recall, respectively. The average precision means CBAM, EMA, CA, ECA, and GAM attention mechanisms are improved by 1, 0.1, 0.5, 0.1, and 0.5 percentage points compared to the original model. It is concluded that adding the CBAM attention mechanism is the most effective way to improve the accuracy of tomato target detection. Therefore, the final decision was to

introduce the CBAM attention mechanism into the YOLOv8s model.

Improved model ablation test

In order to verify the performance enhancement effect of the MCS-YOLOv8 model proposed in this study, the ablation test was designed by comparing the MCS-YOLOv8 with the original YOLOv8s in a step-by-step manner:

1) the original YOLOv8s model; 2) replacing the backbone network with the MobileNetV3 backbone network based on YOLOv8s; 3) introducing the CBAM attention mechanism based on YOLOv8s; 4) modifying the loss function to the SIoU Loss Function based on YOLOv8s; and 5) making all three of these modifications simultaneously based on YOLOv8s by modifying the backbone network to MobileNetV3, introducing the CBAM attention mechanism, and modifying the loss function to two of the SIoU loss functions; 6) making all three modifications at the same time.

According to the above design content, under the same experimental conditions, relying on this paper on the tomato dataset for the test, the test results are shown in Table 5. The table shows that using the original model of YOLOv8s, the average accuracy of the tomato fruit ripening classification recognition of the mean value is 89.2%. In the YOLOv8s model, the core feature extraction part is replaced with the lightweight MobileNetv3 architecture, whose detection accuracy is 89.1%, which shows that MobileNetv3 loses part of its detection accuracy for the purpose of lightweight. By introducing the CBAM attention mechanism on the original YOLOv8s model, the average accuracy of the model is improved by 0.8 percentage points. By replacing the backbone feature extraction network with MobileNetv3 while introducing the CBAM attention mechanism, the mean average accuracy is improved by 0.2 percentage points compared to the original model and by 0.3 percentage points compared to replacing only the backbone network with MobileNetv3, which indicates that the CBAM attention mechanism can be effective for feature extraction in complex environments. Replacing the backbone network of the original YOLOv8 model with the lightweight MobileNetv3 structure, introducing the CBAM attention mechanism, and replacing the bounding box loss function with the SIoU which has a faster convergence speed, the mean average accuracy is 90.3, which is 1.1 percentage points higher than the original model. is the highest in the ablation test.

The above analysis of the experimental results confirms the significant effectiveness of the optimization model proposed in this study on the tomato dataset of this paper.

Table 5. Ablation test.

Mobile Netv3	CBAM	SIoU	Accuracy precision/%	Recall/%	Average precision					Average Precision Mean
					Immature stage	Commutation period	Early ripe	Middle ripe	Mature	
×	×	×	89.1	89.7	81.6	89.9	90.8	92.6	91.2	89.2
√	×	×	88.3	88.5	85.9	88.6	87.6	91.1	92.6	89.1
×	√	×	90.1	90.1	88.9	88.6	87.9	92.0	92.6	90.0
×	×	√	89.9	89.9	87.2	87.8	88.3	92.1	91.5	89.4
√	√	×	89.5	89.6	86.9	88.1	87.8	91.5	92.7	89.4
√	×	√	87.9	89.1	86.3	88.2	87.4	91.8	91.9	89.1
×	√	√	90.2	90.1	86.3	87.8	89.3	91.6	91.6	89.3
√	√	√	91.2	90.2	87.5	88.7	89.9	92.7	92.9	90.3

"×" indicates that this improvement strategy is not used; "√" indicates that this improvement strategy is used.

Comparison of different models

In order to objectively present the advantages of the improved model proposed in this paper, the experiment compares and analyzes the improved model with many current mainstream models. The comparative models involved in the experiment include the two-stage algorithm Faster R-CNN (Ren et al., 2016), in which the Resnet network (He et al., 2016) and the VGG16 network (Simonyan and Zissermann, 2014) are based on the Faster R-CNN implementation, YOLOv3tiny, YOLOv5s, YOLOv6n, YOLOv7tiny, YOLOv8m, YOLOv9 and YOLOv10. This paper's tomato dataset was comparatively analyzed under the same experimental conditions. The results of the comparative analysis of each model are shown in Table 4.

From the data in Table 4, it can be seen that the average accuracy mean of MCS-YOLOv8 compared to Faster R-CNN (Resnet), Faster R-CNN (VGG16), YOLOv3-tiny,

YOLOv5s, YOLOv6, YOLOv7-tiny, YOLOv8s, YOLOv9, and YOLOv10 models (mAP@0.5) is improved by 19.9, 18.6, 2.4, 0.9, 2.9, 0.9, 0.7, 0.3 percentage points, respectively. Both values of precision and recall are also higher than those of other mainstream models. MCS-YOLOv8 improves precision by 2.1 percentage points and recall by 0.5 percentage points compared to the pre-improvement YOLOv8s and mAP@0.5 improved by 1.1 percentage points. Although the detection speed is slightly slowed, the memory occupied by the model is significantly reduced. The average precision mean (mAP@0.5) of the MCS-YOLOv8 model is improved to varying degrees compared to several other models. In terms of detection speed, MCS-YOLOv8 is 2.1ms slower at 5.4ms than the fastest YOLOv3tiny, but it still meets the requirements of real-time detection.

Table 6. Comparison test between the model in this paper and other mainstream models.

Model	Accuracy / %	Recall rate / %	mAP@0.5 / %	mAP@0.5:0.95 / %	Model memory usage / MB	Detection speed / ms	FPS
Faster R-CNN (VGG16)	66.5	73.6	70.4	55.1	115.3	169	5.9
Faster R-CNN (Resnet50)	67.5	75.7	71.7	56.3	98	187	5.3
YOLOv3-tiny	88.3	89.1	87.9	77.7	34.4	3.3	303
YOLOv5s	89.6	89.9	89.4	81.1	18.5	4.4	227.2
YOLOv6	86.7	86.1	87.4	77.1	8.7	15	66.7
YOLOv7-tiny	88.7	88.7	89.4	80.2	12.3	5.1	196
YOLOv8s	89.1	89.7	89.2	82.6	22.5	4.2	238
YOLOv9	88.1	88.7	89.6	78.5	4.6	3.2	312.5
YOLOv10	89.2	89.1	90.0	81.2	5.8	3.4	294
MCS-YOLOv8	91.2	90.2	90.3	82.2	13.8	5.4	185.1

In order to fully illustrate the effectiveness of the improved model proposed in this paper, some test set images in Faster R-CNN (Resnet), Faster R-CNN (VGG16), YOLOv3tiny, YOLOv5s, YOLOv6, YOLOv7tiny, YOLOv8s, YOLOv9, and YOLOv10 were selected are

compared with the improved model in MCS-YOLOv8 in this paper, and the results of the confidence score are used to demonstrate the detection performance of the detection model in this paper. The comparison results are shown in Figure 6.

The recognition results in Figure 6 show that the MCS-YOLOv8 algorithm exhibits significant advantages in recognizing tomato ripeness in complex environments. In the case of branch and leaf occlusion (Figure 6a), MCS-YOLOv8 can distinguish tomatoes at different ripening stages more accurately, significantly improving recognition accuracy and confidence compared to other existing models. In fruit-obscured environments (Figure 6b), the YOLOv7tiny model suffers from leakage detection, while MCS-YOLOv8 still maintains a high confidence level thanks to its advanced network structure and optimization strategy. Even under more complex branch and fruit occlusion conditions (Figure 6c), MCS-YOLOv8 achieves 93% recognition accuracy for medium-ripening tomatoes. Under backlighting (Figure 6d), YOLOv10 and YOLOv5s performed well, but MCS-YOLOv8 still outperformed the other models, albeit slightly. In addition, when processing tomato images mixed with different ripening stages (Figure 6e), MCS-YOLOv8 demonstrates excellent recognition accuracy with an average confidence level of more than 90%, showing its high reliability and accuracy in target detection tasks.

The confidence scores from the test images in Figure 6 further validate the advantages of the improved network for tomato fruit detection in natural environments.

Figure 6. Comparison results of different target detection networks.

Conclusions

i) This study proposes an improved lightweight model MCS-YOLOv8s based on YOLOv8s for tomato fruit ripeness detection. The improvement measures are to replace the original model backbone network of YOLOv8s with a more lightweight structure, introduce the attention mechanism in the backbone network, and replace the bounding box loss function of the original model with the SIoU loss function. Not only can we fully utilize the global interaction features, but also effectively balance the problem between the computing speed of the network and the model complexity.

ii) In order to verify the performance of the improved YOLOv8 model, this paper designs an ablation test to analyze eight sets of data quantitatively, and the mean values of precision, recall, and average precision of MCS-YOLOv8 have improved by 2.1, 0.5, and 1.1 percentage points compared with YOLOv8s. The experimental results show that the improved MCS-YOLOv8 model is higher than other models in terms of model detection precision and model detection time.

iii) Under the same test conditions, by comparing the two-stage detection algorithm Faster R-CNN (VGG16, Resnet) and several mainstream models such as the single-stage algorithms YOLOv3tiny, YOLOv5s, and YOLOv7, the average accuracy mean of the improved MCS-YOLOv8 achieves 19.9, 18.6, 2.4, 0.9, 0.9 percentage point improvement, the model memory is reduced by 30% compared to YOLOv8s, and the detection speed is 5.4ms. The improved model achieves better results on the tomato dataset. The experimental results fully confirm that the lightweight model proposed in this study improves the evaluation indexes and achieves more satisfactory results in visual performance.

iv) Although MCS-YOLOv8 performs well in tomato ripeness detection tasks, it may face multiple challenges in practical applications. For example, in complex natural environments, factors such as weather changes (e.g., rain, fog, snow), background disturbances (e.g., weeds, other plants), and differences in fruit sizes and shapes, in addition to light and shading, may have an impact on model performance. Considering the uncertainties in the agricultural operating environment, it is a great challenge for this study's subsequent work to consider the impact of environmental factors, optimize the algorithm for environmental uncertainties, and achieve the algorithmic assistance of piggybacked agricultural robots for the automated

picking of tomatoes to increase the speed of tomato picking and to save the cost of workforce.

Through this study, the ripeness detection technology can be applied to the intelligent picking of agricultural products, providing a basis for the subsequent work of visual recognition, target localization and grading of the picking robot. Moreover, by using the tomato target detection model, a more automated and efficient tomato-picking robot can be developed, which is of great significance and practical value for constructing intelligent modern tomato greenhouses.

References

Chakroun, I., Haber, T., Ashby, T.J., 2017). SW-SGD: the sliding window stochastic gradient descent algorithm. Procedia Comput. Sci. 108:2318-2322.

Chen, Q., Yin, C., Guo, Z., Wang, J., Zhou, H., Jiang, X., 2023. Research status and development trend of key technology of apple picking robot. T. CSAE 39:1-15.

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. Yolox: Exceeding yolo series in 2021. arXiv :2107.08430.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas; pp. 770-778.

Hou, Q., Zhou, D., Feng, J., 202). Coordinate attention for efficient mobile network design. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Nashville.; pp. 13713-13722.

Li, P., Zheng, J., Li, P., Long, H., Li, M., Gao, L., 2023. Tomato maturity detection and counting model based on MHSA-YOLOv8. Sensors (Basel) 23:6701.

Liu, Y., Shao, Z., Hoffmann, N., 2021. Global attention mechanism: retain information to enhance channel-spatial interactions. arXiv: 2112.0556.

Long, Y., Yang, Z., He, M., 2023.[Recognizing apple targets before thinning using improved YOLOv7.[[Article in Chinese with English abstract]. T. CSAE 39:191-199.

Luo, Q., Wu, C., Wu, G., Li, W., 2024. A small target strawberry recognition method based on improved YOLOv8n model. IEEE Access 12:14987-14995.

Luo, Z., He, C., Chen, D., Li, P., Sun, Q., 2024. A rapid detection model for passion fruit based on lightweight YOLOv8. T. J. Agr. Machin. 1-12.

Lv, Q., Lin, G., Jiang, J., Wang, M., Zhang, H., Yi, S., 2024. Green citrus fruit detection in natural scenes based on improved YOLOv5s model. T. CSAE 40:147-154.

Mazen, F.M.A., Nashat, A.A., 2019. Ripeness classification of bananas using an artificial neural network. Arab. J. Sci. Eng. 44:6901-6910.

Miao R, Li Z, Wu J. 2023. A lightweight cherry tomato ripening detection method based on improved YOLO v7. T. J. Agr. Machin. 54:225-233.

Mim, F.S., Galib, S.M., Hasan, M.F., Jerin, S.A., 2018. Automatic detection of mango ripening stages application of information technology to botany. Sci. Hortic. 237:156-163.

Mulyani, E.D.S., Susanto, J.P., 2017. Classification of maturity level of fuji apple fruit with fuzzy logic method. Proc. 5th Int. Conf. on Cyber and IT Service Management (CITSM), Denpasa; pp. 1-4.

Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., Huang, Z., 2023. Efficient multi-scale attention module with cross-spatial learning. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island; pp. 1-5.

Peng, X., Pan, Q., Tian, N., 2023. WCF-MobileNetV3:Lightweight image recognition network for CXR images of new coronary pneumonia. Comput. Eng. Appl. 59:224-231.

Qian, S., Ning, C., Hu, Y., 2021. MobileNetV3 for image classification. Proc. 2nd Int. Conf. on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang; pp. 490-497

Qiu, Z., Huang, Z., Mo, D., Tian, X., Tian, X., 2024 . GSE-YOLO: a lightweight and high-precision model for identifying the ripeness of pitaya (dragon fruit) based on the YOLOv8n improvement. Horticulturae 10:852.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE T. Pattern Anal. 39: 1137-1149.

Rezatofighi, H., Tsoi, N., Gwak, J.Y., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: a metric and a loss for bounding box regression. Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition; pp. 658-666.

Roy, A.M., Bose, R., Bhaduri, J., 2022. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. Neural Comput. Appl. 34:3895-3921.

Sandler, M., Howard, A., Zhu, M., Andrey, Z., Chen, L., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition; pp. 4510-4520.

Simonyan, K., Zissermann, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Sun, Y., He, J., Wei, F., Yang, W., 2023. Development of tomato industry in China and evaluation of its international competitiveness in the 13th Five-Year Plan. China Cucurbit 36:112-116.

Tan, H., Ma, W., Tian, Y., Zhang, Q., Li, M., Li, M., Yang, X., 2024. Target detection method for balsam pear based on improved YOLOv8n. T. CSAE 40:178-185.

Tian, Y., Qin, S., Yan, Y., Wang, .J, Jiang, F., 2024. Detection of blueberry ripeness in field complex environment based on improved YOLOv8. T. CSAE 40:153-162.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G., 202). Yolov10: Real-time end-to-end object detection. arXiv:2405.14458.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q., 2020. ECA-Net: Efficient channel attention

for deep convolutional neural networks. Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition; pp. 11534-11542.

Wang, X., Song, J., 2021. ICIoU: Improved loss based on complete intersection over union for bounding box regression[J]. IEEE Access 9: 05686-105695.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.), Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11211. Cham, Springer.

Xu, X., Zhang, L., Yue, J., Zhong, H., Wang, Y., Liu, J., Qiao, H., 2024. Parallel splicing recognition algorithm for UAV images in farmland environment T. CSAE 40:154-163.

Zhao, B., Liu, S., Zhang, W., Zhu, L., Han, Z., Feng, X., Wang, R., 2024. Performance optimization of lightweight Transformer architecture for cherry tomato picking. Chin. J. Agr. Machin. 1-13.

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., Zuo, W., 2021. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE T. Cybernetics 52:8574-8586.