

Article

A Method for Detecting Tomato Maturity Based on Deep Learning

Song Wang ¹, Jianxia Xiang ^{1,*} , Daqing Chen ¹  and Cong Zhang ²

¹ School of Mathematics and Computer, Wuhan Polytechnic University, Wuhan 430048, China; wangsong@whpu.edu.cn (S.W.); chendaqing524@163.com (D.C.)

² School of Electrical Engineering, Wuhan Polytechnic University, Wuhan 430048, China; zc@whpu.edu.cn

* Correspondence: xjx990620@163.com; Tel.: +86-1592-612-5201

Abstract: In complex scenes, factors such as tree branches and leaves occlusion, dense distribution of tomato fruits, and similarity of fruit color to the background color make it difficult to correctly identify the ripeness of the tomato fruits when harvesting them. Therefore, in this study, an improved YOLOv8 algorithm is proposed to address the problem of tomato fruit ripeness detection in complex scenarios, which is difficult to carry out accurately. The algorithm employs several technical means to improve detection accuracy and efficiency. First, Swin Transformer is used to replace the third C2f in the backbone part. The modeling of global and local information is realized through the self-attention mechanism, which improves the generalization ability and feature extraction ability of the model, thereby bringing higher detection accuracy. Secondly, the C2f convolution in the neck section is replaced with Distribution Shifting Convolution, so that the model can better process spatial information and further improve the object detection accuracy. In addition, by replacing the original CIOU loss function with the Focal-EIOU loss function, the problem of sample imbalance is solved and the detection performance of the model in complex scenarios is improved. After improvement, the mAP of the model increased by 2.3%, and the Recall increased by 6.8% on the basis of YOLOv8s, and the final mAP and Recall reached 86.9% and 82.0%, respectively. The detection speed of the improved model reaches 190.34 FPS, which meets the demand of real-time detection. The results show that the improved YOLOv8 algorithm proposed in this study exhibits excellent performance in the task of tomato ripeness detection in complex scenarios, providing important experience and guidance for tomato ripeness detection.



Citation: Wang, S.; Xiang, J.; Chen, D.; Zhang, C. A Method for Detecting Tomato Maturity Based on Deep Learning. *Appl. Sci.* **2024**, *14*, 11111. <https://doi.org/10.3390/app142311111>

Academic Editors: Michele Raffaelli, Marco Fontanelli and Daniele Antichi

Received: 22 October 2024

Revised: 20 November 2024

Accepted: 22 November 2024

Published: 28 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: YOLOv8; tomato maturity detection; swin transformer; DSConv; deep learning

1. Introduction

The tomato is not only one of the world's leading vegetables, but also it can be eaten raw as a fruit. It is loved for its rich nutrition and good taste. It is not only rich in fiber but also in powerful antioxidant vitamins A and C. Consumption of tomatoes helps to reduce the possibility of developing osteoporosis, cardiovascular problems, and certain types of cancer [1]. Tomatoes are now widely used as food additives and pharmaceutical ingredients, such as tomato juice and tomato salsa. Various tomato products have specific requirements for their ripening level. Therefore, it is particularly critical to know the level of ripening of tomatoes in each use [2]. The fruit color of tomatoes will change during the ripening process. This change reflects the physiological and biochemical reactions of tomatoes affected by external environmental conditions during their growth. A central element in determining tomato quality is its degree of ripeness. The harvest time of tomatoes is closely related to their ripeness. If tomatoes are harvested too early, they may not reach the desired level of ripeness, thus compromising the maximization of their economic benefits. Conversely, if harvested too late, tomatoes may become overripe and perishable during distribution. Therefore, to ensure the quality of tomatoes during food processing, long-distance transportation, or storage, their ripeness and timing of harvest must be accurately evaluated.

Traditionally, most tomato picking is performed manually. Workers assess the ripeness of tomatoes mainly on the basis of various characteristics such as color, smell, and sensation, which often results in human error due to the limited ability of human perception and judgment. And by the light conditions, background interference, and other environmental factors, the traditional method is very easy to misjudge. Manual detection of tomato ripening degree has a large error, which not only reduces the efficiency of work but also causes a waste of resources. Especially when the ripening degree of tomatoes is quite close, the misjudgment is more common [3,4]. With continuous development in electronic and computer technology, machine inspection has increasingly replaced manual methods, especially in labor-intensive tasks such as fruit and vegetable inspection and harvesting. This shift is seen as a key step in modernizing agriculture and improving the Precision and efficiency of agricultural production. It also marks the successful implementation of automated and intelligent inspection systems for fruits and vegetables [5]. In addition, fruit and vegetable maturity monitoring provides solid theoretical and technological support for the collection of data on their growing environment and real-time access to ecological planting information [6,7].

A method that can accurately and quickly detect the maturity of tomatoes is crucial for harvesting tomato fruits. It can not only help farmers determine the best harvest time and prevent immature or over-mature fruits from entering the market, thereby maximizing the quality and market value of the fruit. In addition, the application of technologies such as maturity detection can promote the automation and intelligence of agricultural production, thereby improving the accuracy and efficiency of detection and reducing the need for manual operation. In addition, by accurately detecting and managing maturity, resource waste can be reduced, which is conducive to environmental protection and sustainable development. At present, scholars at home and abroad have done a lot of work on tomato ripeness recognition and achieved some achievements. However, due to the influence of light, occlusion, background color, and other factors, the model still has some deficiencies in detection accuracy and speed. Therefore, the motivation of this paper is to study an improved YOLOv8 algorithm to reduce the influence of factors such as occlusion and similarity of fruit and background colors, thereby improving the accuracy and speed of tomato maturity detection. The following are this paper's main contributions:

(1) By replacing the original CIOU with Focal-EIOU, the model makes full use of Focal Loss's ability to resolve class imbalance and EIOU's superior bounding box regression, allowing the model to focus more on small objects and subtle changes in complex environments, and thus improving the performance of recognizing tomatoes at different stages of ripening.

(2) The model's ability to model global and local information in tomato images is enhanced by introducing Swin Transformer in the backbone, which enables the model to better capture the fine-grained features in tomato images, thus enhancing the model's ability to extract features.

(3) The Distribution Shifting Convolution fusion at the neck increases the 'interaction' information between the feature maps by performing convolution operations at different locations, which allows the model to adapt to the changes in image features of tomatoes with different ripeness levels and obtain more diversified feature information in different regions of the image, thus improving the accuracy of tomato ripeness detection.

This paper is organized as follows: Section 2 reviews existing work related to tomato ripening detection and discusses the strengths and weaknesses of current methods. Section 3 describes the experimental datasets, experimental configurations, proposed methods, and evaluation metrics used in this paper. Section 4 shows the results of the experiments and compares the performance of the different methods. Section 5 provides an in-depth discussion of the experimental results and analyzes the findings and challenges of the study. Finally, Section 6 summarizes the main contributions of this paper.

2. Related Work

With the rapid development of artificial intelligence, image processing and computer vision systems have been widely applied in the recognition, classification, grading, and quality evaluation of agriculture. In recent years, many studies have been conducted on fruit ripeness detection using computer vision, and certain achievements have been made. But many methods are still in the research stage and cannot be applied in practical applications [8]. Therefore, further exploration is needed in terms of maturity detection.

In 2015, El Bendary et al. [9] developed an automated method based on color features to assess the ripeness of tomatoes. They achieved automated classification of tomatoes at different ripeness levels by transforming the images into color space and then combining support vector machines (SVMs), linear discriminant analysis (LDA), and principal component analysis (PCA) for feature extraction and classification. The method demonstrated the accuracy of maturity assessment using image data. However, due to the limitations of the dataset, the method has some limitations in practical application and is difficult to be widely disseminated. The accuracy of maturity assessment using image data was demonstrated.

To solve the interference of light changes and overlapping objects, Zhao et al. [10] developed a tomato detection algorithm based on multi-feature image fusion. The method consists of extracting two key feature images: an a^* component image from the Lab* color space and an I component image from the YIQ color space. They use wavelet transform to fuse the feature information of the two images and then use the adaptive thresholding technique to segment the tomato from the background and use the morphological operation pass to eliminate a small amount of noise. To further address the effects of illumination and occlusion, Liu et al. [11] proposed a fruit detection algorithm based on multi-feature combination that uses color images and employs HOG descriptors and SVM classifiers for tomato detection. The detection accuracy is improved by a coarse-to-fine scanning framework and a false positive removal (FCR) method. Their method achieved better results with the majority of tomatoes being detected, but the method has limited applicability as the data used in the experiments were near-red tomatoes that were already ripe.

With the rapid development of artificial intelligence technology, deep learning has been widely used and has made significant breakthroughs in fruit detection. Deep learning algorithms are capable of automatically learning and extracting complex features in images, which makes them particularly suitable for tasks such as fruit identification and ripeness estimation. For example, Kim et al. proposed a deep neural network-based tomato ripeness estimation method that uses cross-entropy loss and mean-variance loss to train a model to accurately estimate the ripeness of tomatoes [12]. The continuous innovation of deep learning methods has led to a growing variety of fruit detection algorithms, of which the two most common categories are two-stage detection algorithms and single-stage detection algorithms. Two-stage algorithms, such as FasterR-CNN [13] and maskR-CNN [14], provide high accuracy by first generating candidate regions and then performing fine processing, while single-stage algorithms, such as the YOLO series [15–20], provide faster detection speed through end-to-end processing, which is suitable for real-time detection requirements. The application of these technologies is driving the continuous development of fruit inspection technology towards higher accuracy and efficiency.

For example, Polder et al. applied the MaskR-CNN algorithm to tomato detection in greenhouses [21]. Huang et al. proposed a method that uses fuzzy c-means combined with Hough transform to detect tomato ripeness [22]. In 2021, Zu et al. used Mask R-CNN with a ResNet50-FPN backbone network to detect and segment green tomatoes in the greenhouse [23]. The algorithm uses ROIAlign bilinear interpolation to calculate the target area generated by the training feature map and achieves the detection and segmentation of ripe green tomatoes through the parallel actions of the ROI target category, bounding box regression, and mask. Because they only conducted experiments on green tomatoes, the model is still limited. Kuang WenHsieh et al. used a binocular vision camera based on the R-CNN algorithm [24] to obtain the position of mature tomatoes and estimate their size by combining an object detection network with a depth camera. Minagawa et al. used

MaskR-CNN to detect tomato strings [25] and then used the detected clustering images to predict the harvest time of tomatoes. Due to the fact that the images of its test data were captured during the day, changes in ambient lighting affect the detection of maturity. References [26–28] apply Faster R-CNN to tomato ripeness detection. In 2022, Wang et al. proposed an improved FasterR-CNN algorithm, MatDet, and used it for multi-target tomato ripeness detection in complex scenes [29]. The algorithm uses ResNet-50 as the backbone and RoIAlign to obtain more accurate bounding boxes. In addition, the model also uses Path Aggregation Network (PANet) to solve the difficulty of detecting tomato maturity in complex scenes.

In 2020, Liu [30] and others proposed an improved tomato detection model based on YOLOv3 [17]. This model incorporates dense architecture [31] into YOLOv3, making it easier for the model to reuse features and learn richer tomato feature representation. They also used C-Bbox instead of traditional R-Bbox to obtain more accurate IoU. The proposed model has significant improvements in the impact of lighting, but the occlusion problem is not well solved, and the accuracy is expected to be improved. In 2022, Su et al. [32] proposed an improved real-time tomato ripeness detection model based on YOLOv3. To improve the detection accuracy and speed, they used the K-means clustering algorithm [33] and the squeezing and incentive attention mechanism module [34], and their proposed model outperforms the traditional YOLOv3 and YOLOv5 in terms of detection speed and has a high detection accuracy. However, their approach does not address the problem of small tomato targets and the similarity of the fruit color to the background color, resulting in low detection accuracy. Li et al. [35] investigated a tomato ripeness detection method based on the YOLOv5s tomato model. They improved the performance of small target detection using a data augmentation method, and their method has a significant improvement in terms of improved model size and detection speed. However, the images used for their training were all from the same greenhouse, so further research is needed for tomato ripeness detection in complex environments. In 2023, Li et al. [36] proposed an automated ripening grading and counting model that improves the backbone network of YOLOv8 and enhances the network's ability to extract diverse features. The method provides good technical support for accurate grading, picking, and counting of tomato fruit. However, the model focuses mainly on the detection of large tomatoes and does not consider the detection of small target tomatoes, so the model still has limitations.

Although many researchers have studied tomato maturity detection methods, there are still some research gaps and areas to be explored. For example, different types of data (such as images, spectra, temperature, and humidity) can be fused to improve the accuracy of maturity detection. Currently, most studies focus on a single data source, and multimodal data fusion can provide more comprehensive information. In addition, although deep learning performs well in image recognition, it consumes a lot of computing resources. Therefore, it is also an important direction to study lightweight tomato maturity detection models to adapt to mobile devices while maintaining high accuracy and meeting real-time detection.

Overall, although existing methods for tomato ripeness detection have made significant progress in terms of accuracy and effectiveness, many researchers and engineers have successfully applied various deep learning techniques and image processing algorithms to improve the accuracy of detection. However, these methods still face some challenges in practical applications, especially in terms of real-time performance, computational resource consumption, and adaptation to complex environments. For example, in highly dynamic agricultural environments, factors such as climate change, differences in lighting conditions, and background clutter can affect the performance of the detection system. In addition, most of the existing methods have a high demand for computational resources, a problem that is particularly prominent in mobile devices or edge computing environments, limiting the widespread application of these techniques. To address these challenges, this study proposes a new detection method by introducing Distribution Shifting Convolution [37] and the Swin Transformer module. The main advantage of using Distribution Shifting

Convolution is its ability to effectively reduce the computational complexity, thus allowing the model to be deployed on resource-constrained devices more easily. Meanwhile, the hierarchical self-attention mechanism introduced by the Swin Transformer module enables the model to better capture the global information in the image, which is important for improving the accuracy of detection. This combination not only improves the efficiency of detection but also enhances the adaptability of the model in variable and complex environments, showing good application prospects.

3. Materials and Methods

3.1. Material

3.1.1. Dataset Acquisition and Processing

In this study, we used the Laboro–Tomato dataset [38] provided by Laboro. It is a very valuable dataset. And multiple researchers have used this dataset for experimental research. Asim Khan et al. [39] conducted comparative experiments on tomato maturity classification using this dataset and their newly introduced tomato dataset. This dataset covers two subsets of tomato images of different sizes. There are a total of 804 images of tomato fruits, including 442 images of large tomatoes and 362 images of small tomatoes. This image dataset presents tomatoes at various stages of ripeness, which can be categorized into three types based on their degree of ripeness: unripe, semi-ripe, and fully ripe. The skin of immature tomato fruit is almost completely green, or relatively less red, filled with 30% red. Semi-ripe tomatoes are lighter in color, about 30–89% red, and it takes some time to fully mature. When the tomato fruit is 90% or more red, the tomato fruit is fully mature. At this point, it is ready to harvest. Figure 1 shows the 6 maturity categories of the dataset we used.

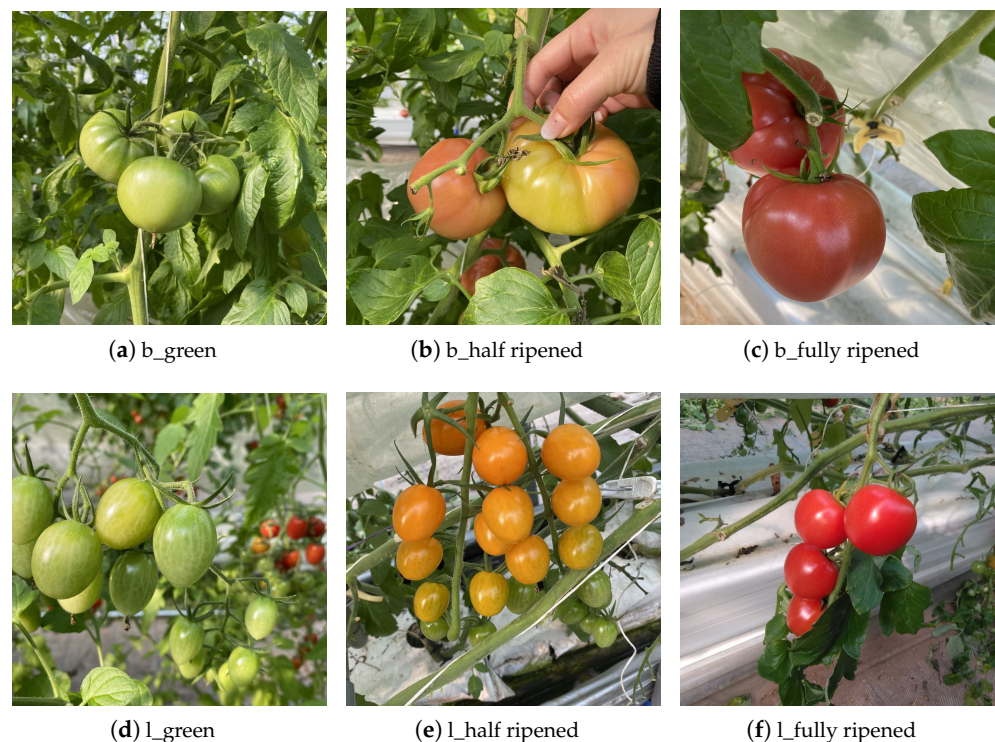


Figure 1. Tomato ripeness categories. (a) Immature normal sized tomatoes. (b) Semi mature normal sized tomatoes. (c) Fully ripe normal sized tomatoes. (d) Immature little tomatoes. (e) Semi ripe little tomatoes. (f) Fully ripe little tomatoes.

3.1.2. Image Preprocessing and Dataset Partitioning

After obtaining the dataset, we need to label the detection targets in the dataset. LabelImg was used to label tomato images in this study. LabelImg is a lightweight open source image annotation tool that is very simple and direct to use, and it can run efficiently on various systems. It is more suitable for our dataset than other annotation software, so we use it to annotate this experimental dataset. Three categories—unripe (green), semi-ripe (half_ripened), and fully ripe (fully_ripened)—were used to label large (normal size) and little (cherry tomato) tomatoes. And the dataset is divided into a training set, a validation set, and a test set in the ratio of 7:2:1. Table 1 shows the number of categories in the dataset. There are 6702 targets in the training set, 2032 targets in the validation set, and 1043 targets in the test set. The specific number of targets for each category is shown in Table 1.

Table 1. Number of experimental categories.

Categories	Train Set	Val Set	Test Set
total	6702	2032	1043
b_green	1275	396	183
b_half ripened	442	122	72
b_fully ripened	281	98	41
l_green	3166	915	515
l_half ripened	693	215	112
l_fully ripened	845	286	120

3.2. Training Platform and Parameter Settings

This study was conducted on Ubuntu 22.04 operating system using PyTorch2.1.0 framework for developing deep learning models and trained on NVIDIA V100 GPUs. The detailed configuration of the experimental environment is shown in Table 2, and the parameter settings for the training phase are shown in Table 3. During the training process, we used the SGD optimizer and set the size of the input images uniformly to 640×640 . To improve the model's ability to detect targets of different sizes, the image enhancement operation consists of randomly scaling the image within a range of 50%, i.e., the image is scaled between 50% and 150% of the original size. The training period (epoch) is set to 400, and the batch size is set to 16. In addition, the initial learning rate is set to 0.01, and warm_epochs is set to 3 so that the learning rate is gradually increased in the first 3 epochs to ensure stability and convergence of the training.

Table 2. Experimental environment configuration.

Category	Configuration
GPU	NVIDIA V100
System environment	ubuntu22.04
Framework	Pytorch2.1.0
Programming voice	Python 3.10.13

Table 3. Parameter Settings for model training.

Parameter	Value
Image-size	640×640
Epochs	400
Batch-size	16
lr	0.01
warmup_epochs	3
Optimizer	SGD

3.3. YOLOv8 Model

YOLOv8 [40] is a highly efficient deep learning model in the field of target detection, which inherits the tradition of YOLO series algorithms and is capable of achieving high accuracy and high-speed target detection. YOLOv8 is based on the CSP (Cross-Stage Partial) structure of YOLOv5 [41] in the backbone network, but with optimization of the module design. YOLOv8 has replaced the C3 module in YOLOv5 with the C2f module to further reduce the model size and improve computational efficiency. YOLOv8 replaces the C3 module in YOLOv5 with the C2f module to further reduce the model size and improve computational efficiency. Meanwhile, it adopts the improved Path Aggregation Network (PANet) as the feature fusion module and introduces the CIoU loss function, which effectively improves the accuracy of target frame regression. In addition, YOLOv8 has also optimized the data enhancement and training strategy, which makes the model's performance more stable under different scales and backgrounds. The model structure of YOLOv8 is shown in Figure 2.

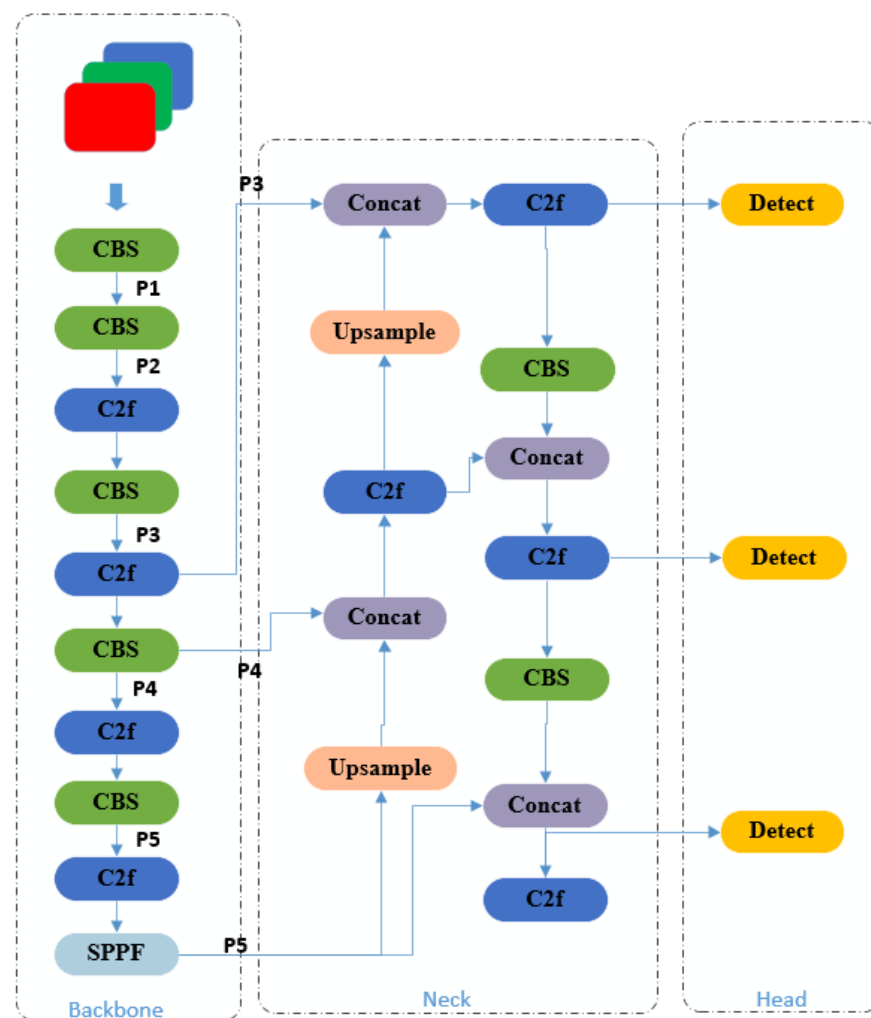


Figure 2. The model structure of YOLOv8.

The network structure of YOLOv8 can be divided into several important parts: the backbone, the Feature Pyramid, and finally the head. First, YOLOv8's backbone network uses an efficient structure (CSPDarkNet) to extract features from images. This network architecture is optimized to extract richer image features while ensuring lower computational complexity, providing more accurate information for subsequent detection tasks. Next is the Feature Pyramid, or neck part, where YOLOv8 adopts a more advanced feature fusion approach. In the target detection task, targets of different scales can produce features of

different sizes in the image, and the Feature Pyramid Network (FPN) can effectively deal with this problem. The combination of FPN and PANet (Path Aggregation Network) is used in YOLOv8 to further enhance the fusion of multi-scale features. With this design, YOLOv8 is able to better handle both small and large targets and maintain good detection accuracy in both multi-scale images. Finally, there is the head network of YOLOv8, which is responsible for the final target classification and bounding box regression. YOLOv8 uses the Adaptive Feature Layer and the new detection head for both classification and localization tasks, making the network not only able to handle more complex scenarios effectively but also ensuring efficient computational performance.

However, although YOLOv8 performs well in most application scenarios, there are still some performance bottlenecks in some specific cases. For example, in terms of small target detection and robustness in complex backgrounds, YOLOv8 may suffer from leakage and false detection problems. This is mainly due to the limited ability of its feature extraction network to capture fine-grained small targets, and it may be difficult to effectively differentiate between targets and interferences in complex backgrounds. Therefore, this study proposes further improvements based on YOLOv8, aiming to enhance the performance of small target detection, improve the robustness of the model in complex backgrounds, and optimize the computational efficiency to adapt to a wider range of practical applications.

3.4. Improved YOLOv8 Model

In the complex growth environment, due to the influence of many factors, such as similar background color, mutual occlusion between branches and fruit, or different fruit size, it is difficult to detect the target accurately, or the detection accuracy is not high by using the traditional detection method in the process of detecting tomato maturity. In order to more accurately locate tomato fruits and identify tomato maturity categories, we propose an improved YOLOv8s method. This method can improve the detection accuracy and efficiency of tomato ripeness.

In this study, we made several key improvements to the YOLOv8 model to further enhance its performance in tomato ripeness detection. The structure of the improved model is shown in Figure 3. First, we introduced the Swin Transformer module in the backbone network part of YOLOv8. The core idea of this improvement is to enhance the global modeling capability of features with the help of the transformer mechanism. Especially when dealing with long-range dependencies, the self-attention mechanism of Swin Transformer can capture fine-grained contextual information more effectively. This feature is particularly important in the tomato ripeness detection task, as ripeness judgments are often based on subtle color variations and texture differences on the surface of tomato fruits, and Swin Transformer is able to improve the ability to discriminate between small and similar targets (e.g., tomatoes at different ripeness stages) in the presence of complex light variations and background noise, thus significantly improving the accuracy of ripeness detection.

Second, in the neck section, we make an important adjustment to the convolution structure in C2f by using Distribution Shifting Convolution instead of the original standard convolution. Distribution Shifting Convolution enhances feature diversity and invariance through shifting operations, which is particularly effective in multi-scale tomato detection. Especially in the case of tomato fruits with irregular shapes and large size differences, the Distribution Shifting Convolution can better capture the local features at different scales, which improves the robustness and adaptability of the model, and makes it more discriminative when dealing with tomatoes at different stages of ripeness.

Finally, for the loss function, we replace the original CIOU loss function with the Focal-EIOU loss function. This new loss function combines the weighting mechanism of Focal Loss with the accuracy-enhancing property of EIOU, which is more effective in facing the category imbalance problem. In the tomato ripeness detection task, there is a category imbalance between unripe and overripe tomatoes and the traditional loss function often finds it difficult to accurately distinguish between these categories. The Focal-EIOU loss

function is able to dynamically adjust the weights of the difficult-to-classify samples, paying special attention to those tomatoes that are at the boundary of the ripeness judgment, which significantly improves the model's detection accuracy on the complex dataset. Compared with traditional CIOU, Focal-EIOU can more accurately optimize target localization and classification when dealing with small and distant targets, which further improves the accuracy of tomato ripeness determination.

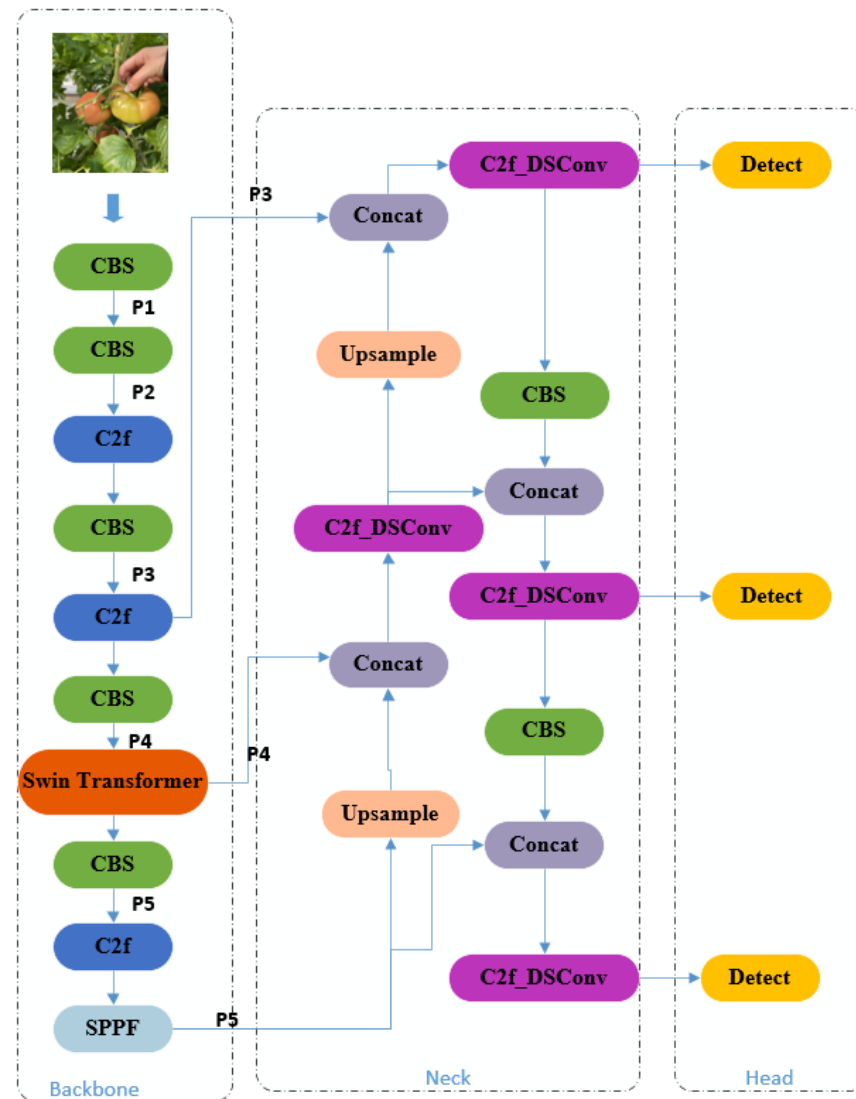


Figure 3. The model structure of improved YOLOv8.

3.4.1. Swin Transformer

In object detection tasks, targets in images often have different scales, and this scale difference may be due to the distance of the object in the image, mutual occlusion between objects, or the size difference of the object itself. Due to the different scale sizes of targets, effective object detection becomes more challenging. Therefore, we introduced the Swin Transformer module in the backbone part of the network to enhance the network's ability to extract features of objects of different scales.

Swin Transformer [42] is a novel visual transformer architecture that solves the computational and memory bottleneck problems of traditional transformers in high-resolution image processing by introducing a local window self-attention mechanism and a shift operation. It divides the image into multiple fixed-size windows and computes self-attention within each window to reduce computational complexity, while shifting the windows allows features between different windows to communicate with each other, effectively

modeling long-range dependencies. Swin Transformer also adopts a hierarchical design to gradually reduce the resolution of the feature map and capture features at different scales, and this design can improve both computational efficiency and memory bottleneck in traditional transformers. This design improves computational efficiency and retains the ability to model global information, which is widely used in image classification, target detection, semantic segmentation, and other visual tasks, and shows excellent performance and efficiency. The structure of Swin Transformer is shown in Figure 4a.

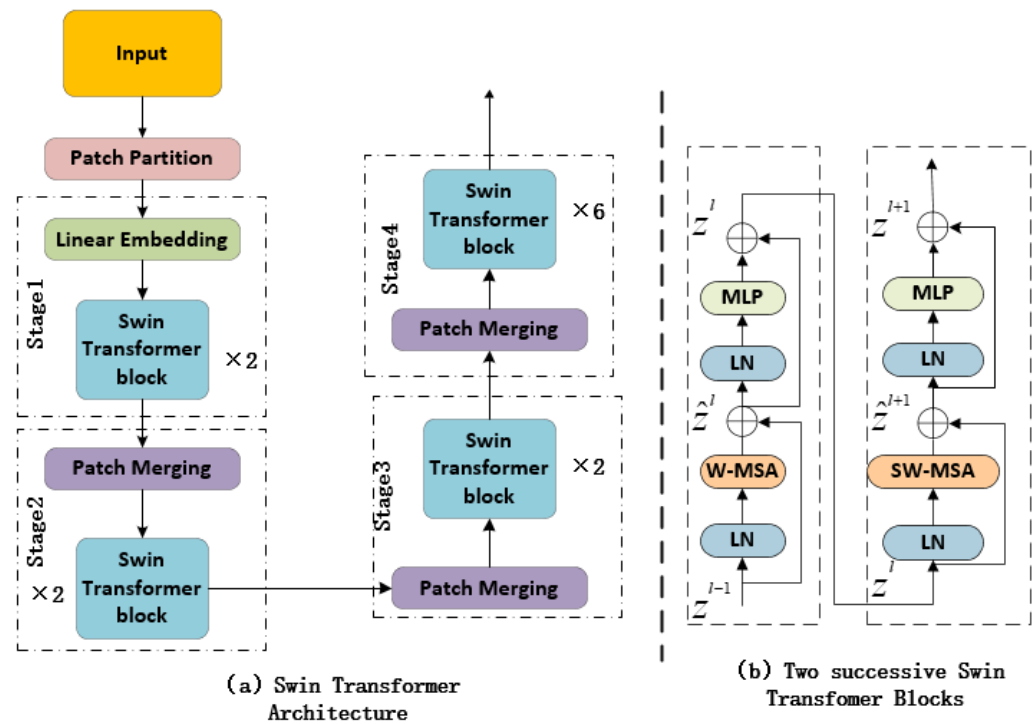


Figure 4. (a) The architecture of a Swin Transformer (Swin-T) [42]; (b) two successive Swin. W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively, in Transformer Blocks.

It is divided into four stages. Before entering Stage 1, the input image is split into non-overlapping, equal-sized patches by Patch Partition, and each patch is regarded as a “token”, and then a feature sequence is obtained. This feature sequence is then sent to the linear embedding layer for projection to obtain a new feature dimension, and after two Swin Transformer blocks for self-attention computation, it enters Stage 2. In Stage 2, the feature dimension is first reduced by patch fusion, and then the feature is transferred to a deeper layer after two Swin Transformer Blocks to compute the attention. Then go to Stage 3 and Stage 4 sequentially to get the final features. In the latter two stages, the features are fused by patch fusion, and then the Swin Transformer Block calculates the self-attention to transform the features to deeper layers. The difference is that in Stage 3, six Swin Transformer Blocks (two of them in a group) are passed through. Each stage changes the dimensionality of the sequence, resulting in a hierarchical representation.

Through this hierarchical multi-scale feature representation, Swin Transformer is able to effectively capture the information at different scales in the image and improve the performance of the model.

The C2f module in YOLOv8 excels in local feature extraction but lacks in capturing global context information and is less effective when dealing with complex backgrounds or lighting changes. The Swin Transformer module has hierarchical feature extraction capabilities and can effectively extract and integrate information on multiple scales, which is particularly important for processing tomatoes at various maturity stages. Moreover, Swin Transformer shows good stability and robustness when handling complex backgrounds

or lighting conditions. Therefore, this study introduces a Swin Transformer module in the high-level feature processing stage of the backbone section and retains the C2f module in the low-level feature extraction stage. Adopting this hierarchical feature extraction strategy can not only improve the global context understanding of the model and improve the accuracy of the model in detecting tomato maturity but also effectively control the consumption of computing resources of the model.

3.4.2. C2f_DSConv Module

Conventional C2f modules usually consist of ordinary convolutional layers that are fixed to certain spatial position relationships when processing images, and the fixed convolutional kernel may miss some subtle local changes or texture information when applied to different regions of the image. In this study, we propose an improved C2f module, the C2f_DSConv module (shown in Figure 5), to improve the performance of the model in the tomato ripeness detection task. The core innovation of this module is to improve the perception of subtle differences in tomato ripeness by adjusting the spatial properties of the convolution operation so that the model can better capture local and global features in the image. The design concept of the C2f_DSConv module is to introduce Distributed Shift Convolution (DSConv) [37] on top of the original C2f module in order to improve the response of the convolutional layer to local variations during the feature extraction process while avoiding the loss or insufficiency of information that standard convolution can introduce when processing complex images. In the tomato ripeness detection task, changes in the appearance of tomatoes are typically gradual and locally dependent, with subtle changes in ripeness including gradual changes in color, textural details of the skin, and small differences in morphology. These changes are often difficult to adequately capture with conventional convolution, and therefore a feature extraction method that can accommodate these detailed changes needs to be developed. By adjusting the convolutional structure in the C2f module, the C2f_DSConv module is able to improve the perception of spatial relationships between different regions in the image while retaining the advantages of the original convolutional operation. This improvement does not rely solely on a simple transformation of the convolution kernel, but by improving the convolution path, the convolution operation allows for a more flexible and accurate perception of spatial relationships.

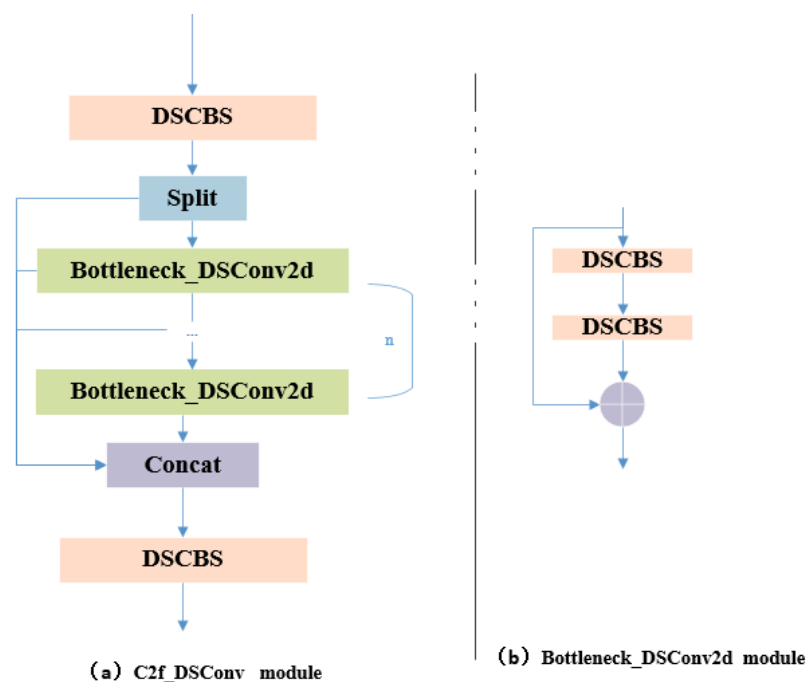


Figure 5. The structure of C2f_DSConv.

3.4.3. Loss Function

In tomato ripeness detection tasks, accurate localization and classification are key to model performance, especially when dealing with tomato images with different ripeness levels. To improve the performance of the model in such tasks, we have improved the loss function of the original YOLOv8 model. The CIOU loss function originally used in the model, as a commonly used target image regression loss, is able to take into account geometric relationships between target images, including factors such as position, size, and relative angle, during the training process. However, the performance of CIOU is not always ideal when dealing with targets with highly imbalanced category distributions and hard-to-detect targets, especially in cases where objects have large scale differences or are occluded. To address this problem, we introduce the Focal-EIOU Loss, which combines the idea of Focal Loss and the optimization of EIOU and is able to further improve the accuracy of small and hard-to-detect objects. The Focal-EIOU Loss firstly mitigates the effect of the background region by adding an adjustment factor to reduce the dominance of the easy-to-detect objects on the loss, which allows the model to pay more attention to the harder-to-detect and smaller tomato targets. At the same time, the EIOU loss incorporates more geometric information into the calculation process, not only considering the degree of overlap between the target frames but also enhancing the shape and orientation of the target frames and other factors, which makes the regression accuracy more refined, especially when dealing with tomatoes with different morphologies, and can effectively improve the fit of the frames. With this improvement, Focal-EIOU is able to guide the model to focus more on the actual location and shape of the tomatoes during the optimization process, rather than being distracted by a large number of background and easily identifiable regions.

In object detection, the target in the image is predicted by the regression of the bounding box. A common metric for assessing the degree of overlap between the predicted bounding box and the true bounding box is the Intersection over Union (IOU). It measures the degree of overlap by calculating the ratio of the area of the intersection region of the predicted and real boxes to the area of their concatenated region. It can be calculated by Formula (1), where $A, B \subseteq S \in R^n$ represent any two boxes.

$$IOU = |A \cap B| / |A \cup B| \quad (1)$$

The Focal-EIOU loss function consists of two parts: one is Focal loss, and the other is EIOU loss. The formula for Focal loss is

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (2)$$

where p_t is the model's predicted probability for the target category. The γ is an adjustment factor that is mainly used to adjust the weights between easy and hard-to-categorize samples. It can usually be adjusted by observing the performance of the model during training. Typically, the value of γ is set between [0,5]. α_t is the weight of the category, usually used to adjust for category imbalances.

The EIOU loss function consists of three parts. They are IOU loss(L_{IOU}), distance loss(L_{dis}), and aspect loss(L_{asp}). The EIOU loss function is defined as Formula (3):

$$\begin{aligned} L_{EIOU} &= L_{IOU} + L_{dis} + L_{asp} \\ &= 1 - IOU + \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \frac{\rho^2(w, w^{gt})}{(w^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2} \end{aligned} \quad (3)$$

where b and b^{gt} represent the center points of the anchor box and the real box, respectively. w and w^{gt} represent the width of the anchor box and the real box, respectively. h and h^{gt} represent the height of the anchor box and the real box, respectively. w^c and h^c are

the width and height of the smallest box covering the anchor box and the real box. The Focal–EIOU loss function is expressed as Equation (4):

$$L_{Focal-EIOU} = \alpha_t(1 - IOU)^\gamma L_{EIOU} \quad (4)$$

where IOU is Formula (1). The γ is an adjustment factor that is mainly used to adjust the weights between easy and hard-to-categorize samples. It can usually be adjusted by observing the performance of the model during training. In this experiment, it is set to the default value of 2 recommended by Focal loss, as it can already give better results.

Therefore, in this experiment, we use the Focal–EIOU loss function [43] to replace the CIOU loss function in the original model to improve the convergence speed and positioning accuracy of the model and solve the problem of imbalance between difficult and easy samples. By introducing the focus mechanism (Focal) and the improved bounding box overlap evaluation index (EIOU), the Focal–EIOU loss function can better handle samples with different difficulty levels, thereby improving the performance and robustness of the model.

3.5. Establishment and Evaluation Indicators of Model

Object detection is an important task in the field of computer vision, and evaluation indicators play a vital role in object detection. It provides an objective and quantitative evaluation of the performance of the model, which can help researchers understand the performance of the model in identifying and locating targets. Through the analysis of evaluation indicators, problems and deficiencies in the model can be identified, and guidance can be provided for further improvement of the model. This study focuses on the detection accuracy and detection speed of tomato ripeness. After the model is built, use several commonly used evaluation indicators for object detection, such as Precision, Recall, mAP, and other indicators to evaluate the performance of the model. These evaluation indicators are calculated as shown in Formulas (5)–(9) [36]:

$$Precision = \frac{TP}{TP + FP} \cdot 100\% \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

Precision measures the proportion of boxes that are actually correct among all the boxes predicted to be positive. Recall measures the proportion of true positive boxes that the model can correctly predict, where TP (True Positive) indicates the correctly predicted target box, FP (False Positive) indicates the incorrectly predicted target box, and FN (False Negative) indicates the missed target box.

$$AP = \int_0^1 P(R) dR \quad (7)$$

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c \quad (8)$$

AP is an indicator for a comprehensive evaluation of Precision and Recall. It is obtained by calculating the area of the Precision–Recall curve. Generally speaking, the higher the AP, the better. AP is generally calculated for each category. mAP is the average of APs of all categories. It is the most common evaluation criterion in object detection tasks, especially in multi-category detection tasks. For each category, its AP is first calculated, and then the average of APs of all categories is calculated. The higher the mAP, the better the overall

performance of the model, where $P(R)$ is the Precision at Recall R . C represents the number of categories, and AP_c is the average Precision of category C .

$$FPS = \frac{Num}{Time} \quad (9)$$

FPS (Frames Per Second) is also an important evaluation metric. where Num indicates the number of frames and Time indicates the time.

In addition to the above evaluation metrics, the confusion matrix is also a common metric for evaluating the performance of target detection models.

It can realistically present the relationship between real categories and predicted categories. In the multi-classification problem, the confusion matrix is a matrix of $N \times N$, whose structure is shown in Figure 6. Each row represents a true category, and each column represents a predicted category. Values on the diagonal represent the number of correctly categorized samples, and values on the off-diagonal represent the number of incorrectly categorized samples.

		Predicted category			
		Category 1	Category 2	...	Category N
Actual Category	Category 1	TP(1, 1)	FP(1, 2)	...	FP(1, N)
	Category 2	FP(2, 1)	TP(2, 2)	...	FP(2, N)
	...	...	...	...	...
	Category N	FP(N, 1)	FP(N, 2)	...	TP(N, N)

Figure 6. The structure of the confusion matrix.

4. Results and Analysis

4.1. Training Results of the Proposed Model

During the training process of the proposed model, the loss curve and accuracy curve reveal the key dynamics of training, as shown in Figure 7. First, at the beginning of training (i.e., the first few epochs), the loss curve decreases rapidly, indicating that the model is effectively learning from the training data and gradually improving the fit. As training progresses, the training and validation losses gradually level off, showing that the model has fitted the training data better. In terms of Precision, Recall, and mAP, the correlation curves show a similar upward trend and eventually stabilize at a higher level, further validating the effectiveness of the model. This series of performances indicates that the proposed tomato ripeness detection model has not been overfitted during the training process, the indexes have maintained good stability, and the model has demonstrated ideal convergence. The smoothness of the loss curve reflects that the model has effectively fitted the training data, while the continuous improvement of the validation metrics indicates that the model's performance on the validation set has also been significantly improved.

Therefore, it can be inferred that the model exhibits good generalization ability and stability in both the training and validation phases.

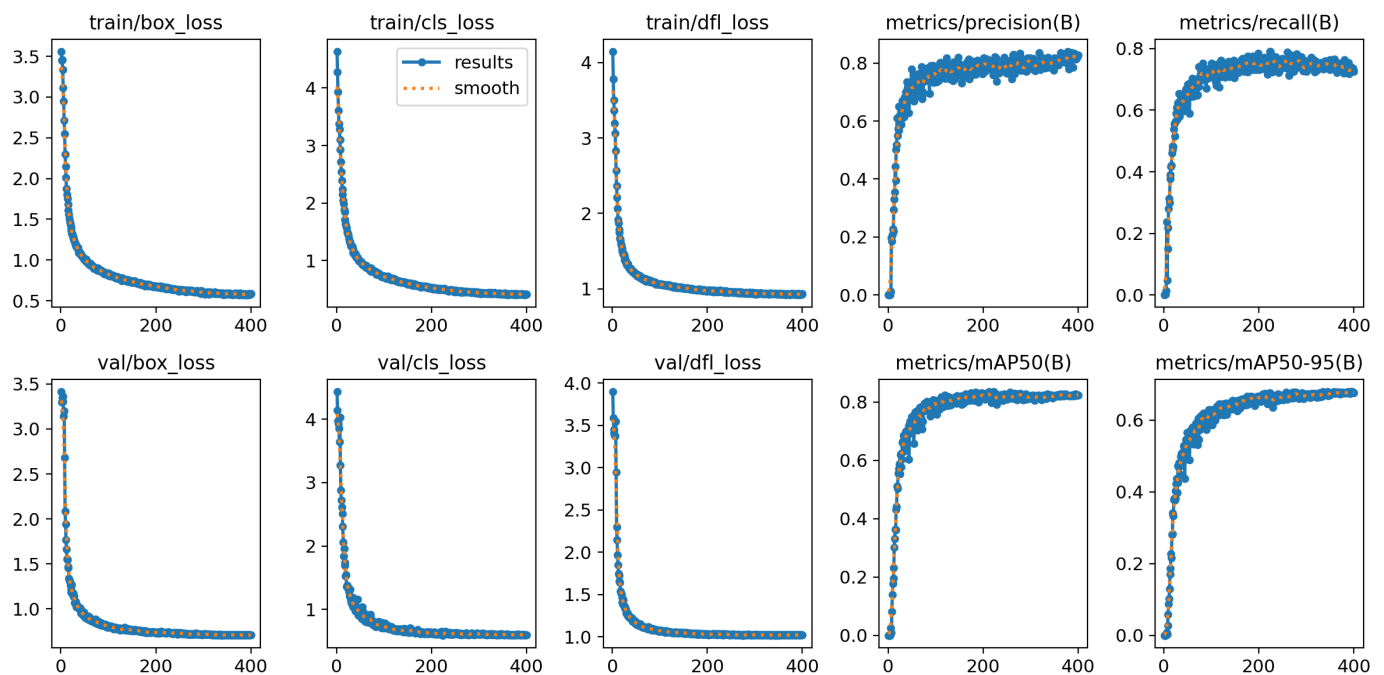


Figure 7. Training and validation process curves for the proposed model. The horizontal coordinate is the training Epoch.

4.2. Improvement of YOLOv8 Ablation Experiment

Ablation experiments are a scientific experimental methodology designed to assess the contribution of specific components of a system to the overall functioning of the system by gradually removing or altering those components. Through the ablation experiment, we can understand the working mechanism of the model more deeply, find the advantages and disadvantages of the model, and provide guidance for the further optimization of the model. To evaluate the effectiveness of the improved model, this study conducted five sets of ablation experiments on the same dataset and equipment. These five experiments include the original YOLOv8s model, the addition of the Swin transformer module to YOLOv8s, the replacement of the original convolution with the DSConv module in YOLOv8s, the use of the Focal-EIOU loss function in YOLOv8s, and our proposed method of synthesizing the three modules. The results of the ablation experiments are shown in Table 4. The bold numbers in the table represent the optimal value.

Table 4. Results of Ablation experiment. Bold values indicate the optimal values in the results. A checkmark indicates that the module is used.

Method	+EIoU	+Swin Transformer	+DSConv	Recall	mAP50	mAP50-95	Inference Time (ms)	Params (M)
YOLOv8s				0.752	0.846	0.713	4.0	11.13
Method 1	✓			0.759	0.851	0.721	4.1	11.13
Method 2		✓		0.783	0.863	0.732	4.4	11.00
Method 3			✓	0.786	0.866	0.728	4.1	10.31
Proposed Method	✓	✓	✓	0.820	0.869	0.727	4.4	10.18

Method 1 in Table 4 uses the Focal-EIOU loss function alone to replace the CIOU loss function used in the original model YOLOv8s. Method 2 refers to the introduction of the Swin Transformer module separately in YOLOv8s. Method 3 refers to replacing the convolution DSConv module separately in the original model. In the case of using

YOLOv8s for maturity detection without any improvement. The Recall rate of the model, mAP50, mAP50-95, inference time, and the number of parameters are 75.2%, 84.6%, 71.3%, 4.0 ms, and 11.13 M, respectively. In Method 1, after using Focal-EIOU to replace CIOU, the Recall of the model, mAP50, and mAP50-90 increased by 0.7%, 0.5%, and 0.8%, respectively. This is because the Focal-EIOU loss function can effectively reduce the influence of the background and make the model pay more attention to the target detection, thus improving the detection accuracy of the model. For the inference time, the inference time of Method 1 is only 0.1 ms more than that of the original model YOLOv8s. It does not have much impact on the overall performance of the model and can be ignored. In terms of the number of parameters, the number of parameters of Method 1 does not change because the loss function does not change the structure of the model. In general, it is effective to use Focal-EIOU to improve the model.

Method 2, a model in which Swin Transformer alone was introduced into the original YOLOv8, improved Recall, mAP50, and mAP50-95 by 3.1%, 1.7%, and 1.9%, respectively, compared with the original YOLOv8 model. The main reason for the accuracy improvement may be that the Swin Transformer is able to capture the subtle features of tomato fruit at different ripening stages more effectively, improving the model's detection accuracy in complex environments. At the same time, the parameter volume of the model was reduced by 0.13 M, which is due to the fact that Swin Transformer optimizes the number of parameters of the model by reducing redundant calculations and adopting the local window mechanism, which reduces the computational complexity, resulting in a reduction in the number of parameters while improving the accuracy. However, the inference time of the model increased by 0.4 ms compared with the original model, probably due to the increase in the local receptive field. Despite the slight increase in inference time, the increase in performance more than compensates for this change, so the model remains valid.

Comparing Method 3 with the original model, after using DSConv to replace the original convolution, the Recall rate of the model, mAP50, and mAP50-95 increased by 3.4%, 2%, and 1.5%, respectively. The number of parameters decreased by 0.82 M. This is because DSConv helps the model to learn richer and more accurate feature representation in the image, thus improving the detection accuracy of the model. Moreover, compared with the original convolution, DSConv reduces the number of parameters and computation by quantization and distribution shift operations. Thus, the computational cost is saved. Although the inference time is extended by 0.1 ms, this module still has an advantage on the whole. Taken together, each of our improved modules has improved the performance of the model. So every module we improve is effective.

In addition, comparing our model with the baseline model YOLOv8s, we can see that our model has a significant improvement in Recall, mAP50, mAP50-90, and the number of parameters. The Recall, mAP50, and mAP50-90 increased by 6.8%, 2.3%, and 1.4%, respectively. The parameter is reduced by 0.95 M. The experimental results show that our model not only improves the detection accuracy but also is more lightweight than YOLOv8s. Although the inference time of the improved model needs more refined calculation, it is negligible compared with the improved model. To sum up, our model shows obvious effectiveness in the tomato maturity detection task.

4.3. Performance Comparison of Different Detection Algorithms

In this paper, several improvements are made to the original YOLOv8s model with the aim of increasing the accuracy and efficiency of the tomato ripeness detection task. The performance of the improved model was evaluated through a series of experiments and compared and analyzed with several benchmark models such as YOLOv5n, YOLOv7-tiny, YOLOv8n, and YOLOv8m. The experimental results are detailed in Table 5, which shows the comparative performance of the improved model with the other models on various metrics.

Table 5. Results of Comparative experiment. Bold values indicate the optimal values in the results.

Model	mAP50	mAP50-95	Inference Time (ms)	FPS
YOLOv5n	0.826	0.648	7.7	169.1
YOLOv7-tiny	0.845	0.643	17.1	156.2
YOLOv8n	0.824	0.681	2.6	288.05
YOLOv8s	0.846	0.713	4.0	204.13
YOLOv8m	0.848	0.721	6.6	134.25
Proposed Method	0.869	0.727	4.4	190.34

From the experimental results, the detection accuracy of the improved model is higher than that of other models. The mAP50 of the improved model is 4.3%, 2.4%, 4.5%, 2.3%, and 2.1% higher than that of YOLOv5n, YOLOv7-tiny, YOLOv8n, YOLOv8s, and YOLOv8m, respectively. The mAP50-90 of the improved model is 7.9%, 8.4%, 4.6%, 1.4%, and 0.6% higher than YOLOv5n, YOLOv7-tiny, YOLOv8n, YOLOv8s, and YOLOv8m, respectively. From the point of view of FPS, although the improved model is not the fastest, it is second only to YOLOv8n and YOLOv8s and faster than YOLOv5n, YOLOv7-tiny, and YOLOv8m. Because we use the Focal-EIOU loss function, more detailed calculations are needed when focusing on targets that are difficult to predict and locate. And use Swin Transformer to increase the local receptive field to help the network capture more image features. So more inference time is needed. Although the inference speed of YOLOv8n and YOLOv8s is relatively faster than our model's, their accuracy is lower. This shows that the improved model not only improves the detection accuracy but also maintains a relatively high inference speed and has a good performance balance. Generally speaking, the improved YOLOv8s model has achieved good performance in the tomato maturity detection task, and its mAP50 and mAP50-95 indexes are significantly better than those of other models and maintain a high inference speed while maintaining a high accuracy. This shows that the improved model has obvious advantages in improving the accuracy and efficiency of tomato maturity detection.

In addition, Table 6 shows the Recall and detection accuracy of different models for each category. It can be seen from the table that the Recall and accuracy of the improved model detection b_half ripened, b_fully ripened, and l_green are better than those of other models. Category b_green also has higher Recall than other models, and its accuracy is second only to YOLOv8m. Although the Recall and accuracy of categories l_half ripened and l_fully ripened are not the highest, they are also improved compared with the baseline model YOLOv8s. These results show that our improved model strengthens the ability to extract important features in complex environments. Overall, the improved model shows superior performance in most categories. Although the model still has room for improvement in the detection of small targets, overall the improved model has improved detection Precision and Recall compared with other benchmark models.

Table 6. Comparison of Recall rates and accuracy for each category of different models. Bold values indicate the optimal values in the results.

Model \ Categorie	b_Green		b_Half Ripened		b_Fully Ripened		l_Green		l_Half Ripened		l_Fully Ripened	
	R(%)	AP50(%)	R(%)	AP50(%)	R(%)	AP50(%)	R(%)	AP50(%)	R(%)	AP50(%)	R(%)	AP50(%)
YOLOv5n	75.9	88.2	77.8	87.5	82.9	79.5	72.6	82.8	67.3	76.0	73.0	81.8
YOLOv7-tiny	77.8	87.5	84.7	88.6	80.5	81.6	73.6	84.4	77.7	82.0	70.8	82.7
YOLOv8n	76.4	86.0	77.6	81.4	75.6	74.0	77.6	83.3	75.0	83.7	82.6	86.0
YOLOv8m	81.5	91.3	77.0	81.9	80.5	79.5	77.2	86.3	73.2	83.9	80.0	85.9
YOLOv8s	78.7	87.3	81.9	86.3	75.6	81.5	72.2	86.0	73.2	79.0	69.5	87.5
Proposed method	82.0	89.8	86.1	90.1	89.0	84.6	80.2	87.2	75.1	83.1	79.9	86.4

To evaluate the detection performance of our improved model on the Laboro–Tomato dataset, we also used the confusion matrix. This is shown in Figure 8. Through the confusion matrix, we are able to get a comprehensive understanding of the model's prediction

accuracy on different categories and infer the overall performance of the model from it, which provides an important reference for further improvements and optimizations, where (a) is the confusion matrix for detection using YOLOv8s; (b) is the confusion matrix after matrix (a) has been normalized in the column direction; (c) is the confusion matrix for detection by our method; and (d) is the confusion matrix after (c) has been normalized in the column direction. The rows of the matrix represent the predicted categories. The columns of the matrix denote the true categories. The elements on the diagonal lines indicate the number of correct predictions made by the model on the corresponding category, while the elements on the off-diagonal lines indicate the misclassification of the model on a certain category. It is clear from Figure 8 that there are two categories that have a lower probability of being correctly recognized with our method than YOLOv8s. One of these categories is *b_green*. For this category, the probability of our method misrecognizing it as *b_half ripened* and *background* has decreased. But the probability of misrecognizing it as *l_green* has increased. This may be due to the fact that some of the features of the large green tomato in the picture are not obvious due to the shooting angle, distance, and other factors. Thus causing the model to recognize more tomatoes of category *b_green* as *l_green*. The other one is category *b_half ripened*. The probability of our method to recognize this category correctly is slightly lower than that of YOLOv8s, which may be due to the fact that some of the features of the fruits with category *b_half ripened* are too similar to those of *b_green* and *b_fully ripened*, which causes the model to fail to distinguish them correctly. However, for the other four categories, the probability of correct recognition by our model is greater than that of YOLOv8s. It can also be seen from the last row and column of the confusion matrix that our model is able to better distinguish the target from the background and reduce the probability of recognizing the target as the background or recognizing the background as the target. This is due to the fact that our improved model is able to better capture the feature information of the target and is able to perform finer target localization, which improves the accuracy of the model detection.

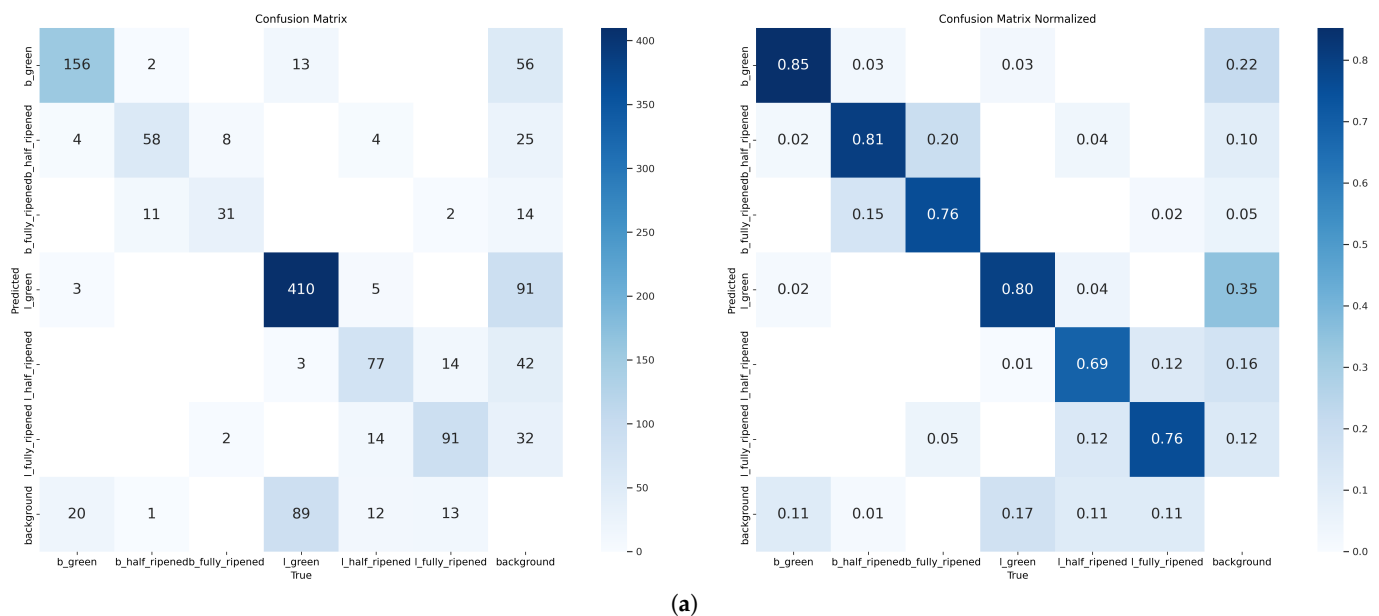


Figure 8. Cont.

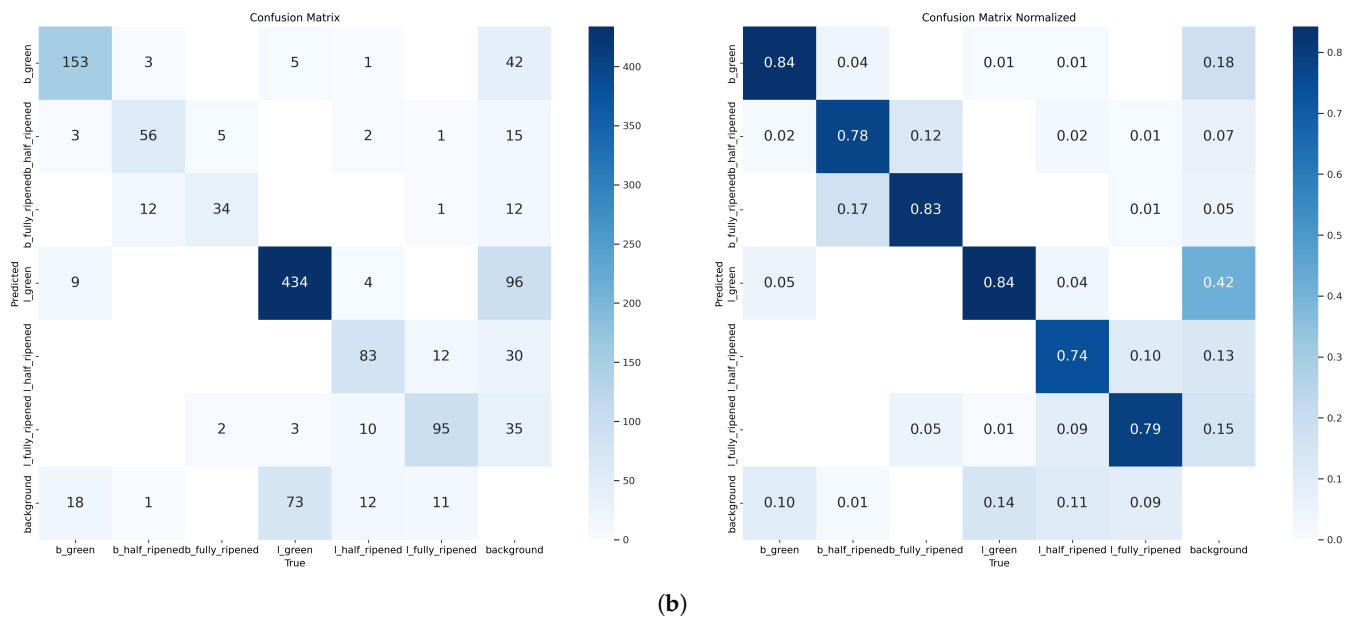


Figure 8. Confusion matrix of YOLOv8s and improved model. The left side of (a) is the confusion matrix detected using YOLOv8s, and the right side is its normalized confusion matrix. The left side of (b) is the confusion matrix detected using the improved model, and the right side is its normalized confusion matrix.

Display of Some Test Results

Some of the detection results are shown in Figure 9. In these images, the background is very complex, and the fruits are obscured from each other, and the color of the fruits in the green ripening stage is similar to the background color, which is not easy to distinguish. In Figure 9, the left side shows the object labels of the input image, the middle image shows the detection results using YOLOv8s, and the right image shows the detection results using an improved model. The red box in the picture represents category *b_green*, the pink box represents category *b_half_ripened*, the yellow box represents category *l_green*, the yellow-green box represents category *l_half_ripened*, and the green box represents category *l_full_ripened*. At the same time, we use arrows to mark the parts that are false detections and missed detections.

Comparing the top three images, we can see that the YOLOv8s model mistakenly detects category *b_green* as *b_half_ripened*, while the test results of the improved model are all correct. From the six pictures of the second row and the third row, it is obvious that the missed detection is caused by the occlusion of branches, and the color of tomato fruit is similar to that of leaves. In the second row of the picture, YOLOv8s missed detecting a fruit of category *b_green*. Our model correctly detects the types of tomato maturity. In the third row of the picture, there are not only missed detections but also false detections of YOLOv8s. It missed detecting a fruit with the category *b_green* at the bottom of the picture. And it also incorrectly detected the leaves at the top of the picture as fruit. In addition, YOLOv8s incorrectly detected one of the fruits with the category *b_green* as *b_half_ripened*. Although our method also missed detecting a fruit with category *b_green* in this image, our method's detection results are still more accurate than YOLOv8s. From the bottom three images, we can observe that the tomato fruit in the picture is smaller and denser, resulting in false detection in the YOLOv8s model. While the improved model can effectively avoid missed detection and false detection. To sum up, although our model also has the problem of inaccurate detection, our method is more accurate than YOLOv8s. Therefore, our method is more suitable to detect the maturity of tomatoes in a complex environment. Because our improved model can more easily learn the characteristic information of tomato fruit from the complex environment, so as to detect and locate tomato fruit more accurately.

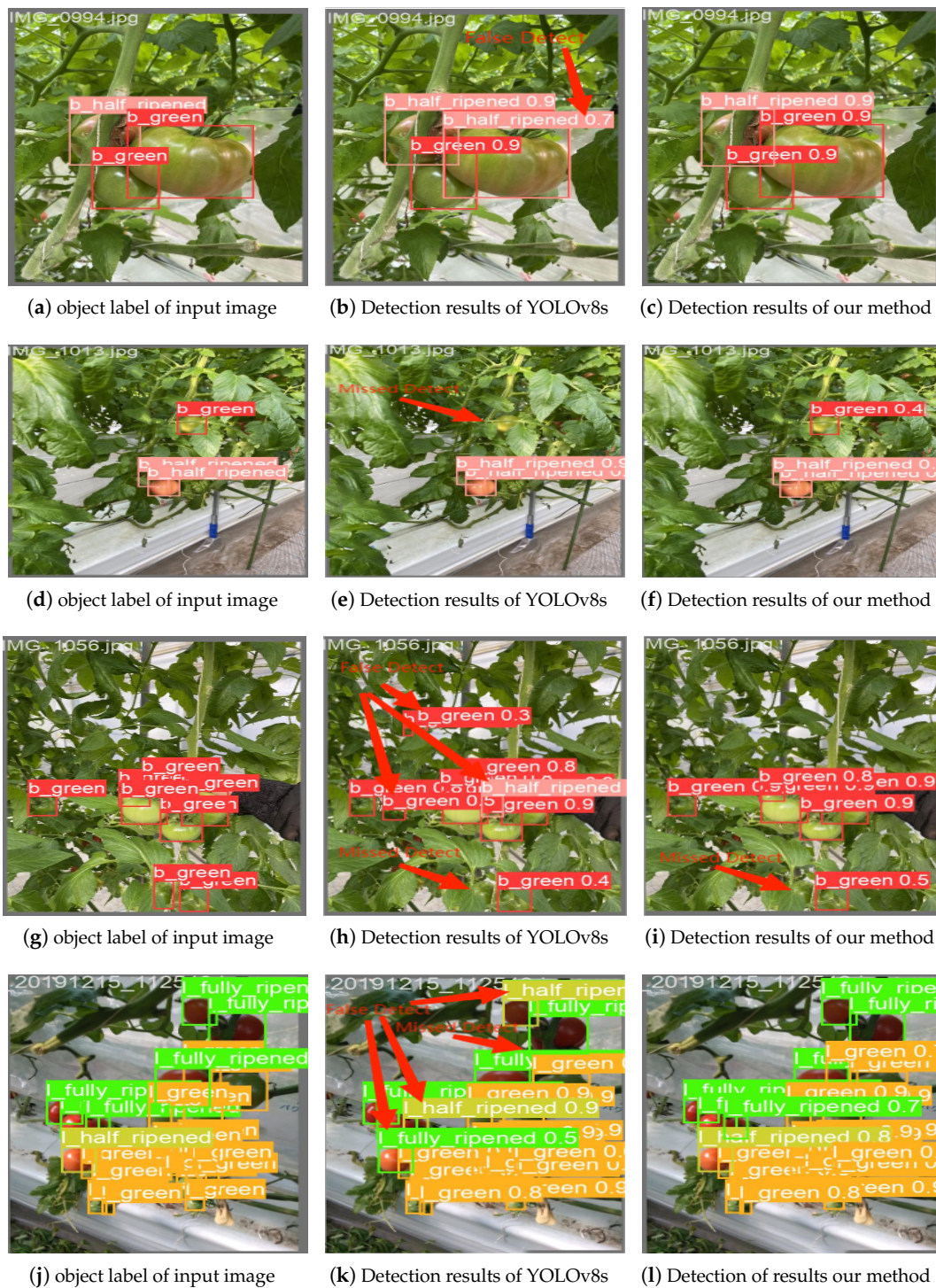


Figure 9. Partial test result. (a,d,g,j) are object labels of input image; (b,e,h,k) are detection results of YOLOv8s; (c,f,i,l) are detection results of our method. We have marked false detection and missed detection with arrows.

Therefore, the results of the above comparison indicate that our improved model can better solve the problems of missed or false detections in complex backgrounds, objects with similar background colors, and occlusion, thereby improving the accuracy of model detection. This improvement is of great significance for tomato ripeness detection tasks, as it can improve the reliability and stability of object detection and provide more effective solutions for practical applications.

5. Discussion

In this study, we propose an advanced deep learning model designed for tomato ripening detection with significant improvements in accuracy and processing speed. Traditional image processing techniques (e.g., Liu et al. [11]) often struggle with computational complexity and slow processing speed. In contrast, our model exploits the power of deep learning to efficiently handle the tomato ripeness detection problem in a complex context. Our approach is especially suitable for scenarios with tomatoes of different sizes, as it accurately detects the ripeness of both large and small tomatoes. This is a key advantage over existing models, which usually focus only on large tomatoes or are limited to controlled environments such as greenhouses (e.g., Li Ping et al. [36]). By using a more diverse dataset that includes tomatoes of various sizes, our model demonstrates high accuracy across a range of targets, addressing an important gap in the current literature. Furthermore, our model is designed for real-time performance, optimizing both detection accuracy and processing speed. This ensures that it can be effectively used in real-world applications that are time-sensitive.

As can be seen from the experimental results, our model significantly outperforms traditional methods and existing deep learning models in several key areas. First, the accuracy of ripeness classification reaches 86.9%, which is 2.3% higher than the baseline method, indicating a significant improvement in classification accuracy. This result suggests that the improved model is more effective in capturing subtle changes in tomato ripeness, resulting in more accurate detection results. Second, our model is robust to occlusion and background changes. In various test scenarios, including the case where the tomato surface is occluded and the tomato color is similar to the background (e.g., Figure 9), our model is still able to detect the ripeness of tomatoes more accurately. This indicates that our model is less sensitive to occlusion and background interference and maintains stable performance under different environmental conditions. In addition, our improved model shows significant improvement in computational efficiency. Compared with the previous method, our model has an inference speed of 4.4 ms per image, which is faster than models such as YOLOv5n and YOLOv7-tiny (as shown in Table 5). These improvements make our model more suitable for real-time surveillance applications and large-scale production environments. By optimizing the network architecture and computational algorithms, we have reduced the consumption of computational resources, and the number of model parameters has been reduced from 11.13 M to 10.18 M, thus improving the overall processing power and efficiency.

Overall, this study has made some progress in identifying tomato ripeness, but there are still some challenges and limitations. For example, tomato ripeness is not only affected by planting variety but is also closely related to environmental factors such as temperature and humidity, and these changes may affect the stability and accuracy of the model. Therefore, future research can further explore how to improve the adaptability of the model to environmental changes, including improving the generalisability of the model under different conditions through data enhancement and model optimization. By addressing these issues, we expect to further improve the usefulness of the model and expand its potential for application in agricultural automation, intelligent detection, and other fields.

6. Conclusions

In this study, an improved tomato ripeness detection method is proposed that significantly improves the detection accuracy and efficiency of the model by combining Swin Transformer, Distributed Shift Convolution, and Focal-EIOU loss function in the YOLOv8 architecture. The method uses the multi-scale feature extraction capability of Swin Transformer to improve the model's ability to capture details in tomato images, especially the small differences between tomatoes at different ripening stages. Meanwhile, the introduction of distribution-shift convolution is used to efficiently handle feature extraction under different location and scale variations of tomatoes in the image while reducing the computational overhead. In addition, the Focal-EIOU loss function further improves the

localization accuracy by focusing on difficult-to-detect tomato targets and optimizing the bounding box regression. The experimental results show that the improved model outperforms the traditional method in several evaluation metrics, demonstrates strong practicality and stability, and can better support the task of real-time tomato ripening monitoring in automated agriculture.

The experimental results showed that the improved model achieved significant performance improvement, with mAP50 and Recall reaching 86.9% and 82.0%, respectively. Compared with the base YOLOv8s model, the improved model achieved a 2.3% improvement in mAP50 and a 6.8% improvement in Recall, showing a substantial improvement in the accuracy of our method in tomato ripeness detection. The improved model also performed very well in detection speed, reaching 190.34 FPS, which can meet the demand of real-time fast detection. Meanwhile, the number of parameters of the model is reduced from 11.13 M to 10.18 M, which means that the model is not only more lightweight but also effectively reduces the resource consumption while improving the computational efficiency, providing more possibilities for practical deployment on mobile devices. Through the validation of the ablation experiments, we demonstrate that the introduction of each module plays a positive role in improving the model performance, further confirming the effectiveness of each module in optimizing the model performance.

This experiment also found that the improved YOLOv8s model showed better performance in tomato ripeness detection tasks by comparing it with the YOLOv5, YOLOv7-tiny, YOLOv8n, and YOLOv8m models. Compared with other models, the improved YOLOv8s model achieved significant advantages in both mAP50 and mAP50-95 metrics, showing higher detection accuracy. Meanwhile, the improved model shows better-balanced performance by maintaining higher inference speed while maintaining higher accuracy. These results indicate that the improved YOLOv8s model has significant advantages in the tomato ripening detection task and that the adopted improvement strategy plays an important role in the model's performance.

In summary, the improved YOLOv8 algorithm proposed in this paper demonstrates excellent performance in tomato ripeness detection tasks, especially in complex scenarios. By optimizing the model structure and improving the detection accuracy, this study provides an effective technical tool for the agricultural field and promotes the progress of real-time tomato ripeness detection. These results not only provide a technical guarantee for the precise management of tomato planting but also provide new ideas for improving the efficiency and quality of agricultural production, which has important practical application value.

Author Contributions: Conceptualization, S.W. and J.X.; methodology, S.W. and J.X.; software, J.X.; validation, S.W., J.X. and D.C.; formal analysis, S.W.; investigation, J.X.; resources, C.Z.; data curation, D.C.; writing—original draft preparation, S.W. and J.X.; writing—review and editing, S.W. and J.X.; visualization, J.X.; supervision, S.W.; project administration, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the digital repository laboroai:LaboroTomato Dataset—<https://github.com/laboroai/LaboroTomato>, accessed on 1 November 2023.

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

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
DSCConv	Distribution Shifting Convolution
IoU	Intersection over Union
AP	Average Precision
mAP	mean Average Precision
FPS	Frames Per Second
FP	False Positive
TP	True Positive
FN	False Negative
TN	True Negative

References

- 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. [\[CrossRef\]](#)
- Begum, N.; Hazarika, M.K. Maturity detection of tomatoes using transfer learning. *Meas. Food* **2022**, *7*, 100038. [\[CrossRef\]](#)
- Zhaoxin, G.; Han, L.; Zhijiang, Z.; Libo, P. Design a robot system for tomato picking based on yolov5. *IFAC-PapersOnLine* **2022**, *55*, 166–171. [\[CrossRef\]](#)
- Lawal, M.O. Tomato detection based on modified YOLOv3 framework. *Sci. Rep.* **2021**, *11*, 1447. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ahlin, K.; Joffe, B.; Hu, A.P.; McMurray, G.; Sadegh, N. Autonomous leaf picking using deep learning and visual-servoing. *IFAC-PapersOnLine* **2016**, *49*, 177–183. [\[CrossRef\]](#)
- Bondé, L.; Ouédraogo, O.; Traoré, S.; Thiombiano, A.; Boussim, J.I. Impact of environmental conditions on fruit production patterns of shea tree (*Vitellaria paradoxa* CF Gaertn) in West Africa. *Afr. J. Ecol.* **2019**, *57*, 353–362. [\[CrossRef\]](#)
- Høye, T.T.; Årje, J.; Bjerger, K.; Hansen, O.L.; Iosifidis, A.; Leese, F.; Mann, H.M.; Meissner, K.; Melvad, C.; Raitoharju, J. Deep learning and computer vision will transform entomology. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2002545117. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sahu, D.; Potdar, R.M. Defect identification and maturity detection of mango fruits using image analysis. *Am. J. Artif. Intell.* **2017**, *1*, 5–14.
- El-Bendary, N.; El Hariri, E.; Hassanien, A.E.; Badr, A. Using machine learning techniques for evaluating tomato ripeness. *Expert Syst. Appl.* **2015**, *42*, 1892–1905. [\[CrossRef\]](#)
- Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. Robust tomato recognition for robotic harvesting using feature images fusion. *Sensors* **2016**, *16*, 173. [\[CrossRef\]](#)
- Liu, G.; Mao, S.; Kim, J.H. A mature-tomato detection algorithm using machine learning and color analysis. *Sensors* **2019**, *19*, 2023. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kim, T.; Lee, D.H.; Kim, K.C.; Choi, T.; Yu, J.M. Tomato Maturity Estimation Using Deep Neural Network. *Appl. Sci.* **2022**, *13*, 412. [\[CrossRef\]](#)
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
- Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
- Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
- Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
- Afonso, M.; Fonteijn, H.; Fiorentin, F.S.; Lensink, D.; Mooij, M.; Faber, N.; Polder, G.; Wehrens, R. Tomato fruit detection and counting in greenhouses using deep learning. *Front. Plant Sci.* **2020**, *11*, 571299. [\[CrossRef\]](#)
- Huang, Y.P.; Wang, T.H.; Basanta, H. Using fuzzy mask R-CNN model to automatically identify tomato ripeness. *IEEE Access* **2020**, *8*, 207672–207682. [\[CrossRef\]](#)
- Zu, L.; Zhao, Y.; Liu, J.; Su, F.; Zhang, Y.; Liu, P. Detection and segmentation of mature green tomatoes based on mask R-CNN with automatic image acquisition approach. *Sensors* **2021**, *21*, 7842. [\[CrossRef\]](#) [\[PubMed\]](#)

24. Hsieh, K.W.; Huang, B.Y.; Hsiao, K.Z.; Tuan, Y.H.; Shih, F.P.; Hsieh, L.C.; Chen, S.; Yang, I.C. Fruit maturity and location identification of beef tomato using R-CNN and binocular imaging technology. *J. Food Meas. Charact.* **2021**, *15*, 5170–5180. [CrossRef]
25. Minagawa, D.; Kim, J. Prediction of harvest time of tomato using mask R-CNN. *AgriEngineering* **2022**, *4*, 356–366. [CrossRef]
26. Sun, J.; He, X.; Ge, X.; Wu, X.; Shen, J.; Song, Y. Detection of key organs in tomato based on deep migration learning in a complex background. *Agriculture* **2018**, *8*, 196. [CrossRef]
27. Hu, C.; Liu, X.; Pan, Z.; Li, P. Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. *IEEE Access* **2019**, *7*, 154683–154696. [CrossRef]
28. Widiyanto, S.; Wardani, D.T.; Pranata, S.W. Image-Based tomato maturity classification and detection using Faster R-CNN method. In Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 21–23 October 2021; pp. 130–134.
29. Wang, Z.; Ling, Y.; Wang, X.; Meng, D.; Nie, L.; An, G.; Wang, X. An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios. *Ecol. Inform.* **2022**, *72*, 101886. [CrossRef]
30. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* **2020**, *20*, 2145. [CrossRef]
31. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
32. Su, F.; Zhao, Y.; Wang, G.; Liu, P.; Yan, Y.; Zu, L. Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment. *Agronomy* **2022**, *12*, 1638. [CrossRef]
33. Burkhart, J. *K-Means Clustering*; Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics: Blacksburg, VA, USA, 2009.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
35. Li, R.; Ji, Z.; Hu, S.; Huang, X.; Yang, J.; Li, W. Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy* **2023**, *13*, 603. [CrossRef]
36. Li, P.; Zheng, J.; Li, P.; Long, H.; Li, M.; Gao, L. Tomato maturity detection and counting model based on MHSA-YOLOv8. *Sensors* **2023**, *23*, 6701. [CrossRef]
37. Nascimento, M.G.d.; Fawcett, R.; Prisacariu, V.A. DSConv: Efficient Convolution Operator. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
38. Laboro. Laboro Tomato: Instance Segmentation Dataset. Available online: <https://github.com/laboroai/LaboroTomato> (accessed on 1 November 2023).
39. Khan, A.; Hassan, T.; Shafay, M.; Fahmy, I.; Werghe, N.; Mudigansalage, S.; Hussain, I. Tomato maturity recognition with convolutional transformers. *Sci. Rep.* **2023**, *13*, 22885. [CrossRef]
40. Jocher, G.; Chaurasia, A.; Stoken, A.; Michael, K. Ultralytics YOLOv8. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 7 November 2023).
41. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Yifu, Z.; Montes, D.; et al. *ultralytics/yolov5: v6. 2-yolov5 Classification Models, Apple m1, Reproducibility, Clearml and deci. AI Integrations*; Zenodo: Geneva, Switzerland, 2022.
42. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
43. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.