applied sciences

MDPI

Article

The Fruit Recognition and Evaluation Method Based on Multi-Model Collaboration

Mingzheng Huang, Dejin Chen and Dewang Feng *

College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China; 000tf21004@fafu.edu.cn (M.H.); 18605971301@163.com (D.C.)
* Correspondence: fdw87@fafu.edu.cn

Abstract: Precision agriculture technology based on computer vision is of great significance in fruit recognition and evaluation. In this study, we propose a fruit recognition and evaluation method based on multi-model collaboration. Firstly, the detection model was used to accurately locate and crop the fruit area, and then the cropped image was input into the classification module for detailed classification. Finally, the classification results were optimized by the feature matching network. In the method, the detection model was based on YOLOv8, and the model was improved by introducing a TripletAttention structure and an Attention Mechanism-Based Feature Fusion (AFM) structure. The improved YOLOv8 model improves the P, R, mAP50, and MAP50-95 indicators by 2.4%, 2.1%, 1%, and 1.3%, respectively, compared with the baseline model on only one generalized "fruit" label dataset. The classification model Swin Transformer used in this study has a classification accuracy of 92.6% on a dataset of 27 fruit categories, and the feature matching network based on cosine similarity can calibrate the classification results with low confidence. The experimental results show that the proposed method can be applied to the maturity assessment of apples and tomatoes, as well as to the non-destructive testing of apples.

Keywords: multi-model collaboration; fruit recognition and evaluation; improved YOLOv8 detection model; swin transformer classification model; feature matching network

Academic Editor: Douglas O'Shaughnessy

Received: 11 December 2024
Revised: 11 January 2025
Accepted: 16 January 2025
Published: 20 January 2025

Citation: Huang, M.; Chen, D.; Feng, D. The Fruit Recognition and Evaluation Method Based on Multi-Model Collaboration. *Appl. Sci.* **2025**, *15*, 994. https://doi.org/10.3390/app15020994

1. Introduction

Today, labor costs in the fruit production process still account for a significant portion of the total cost. Therefore, researching intelligent and automated fruit production processes is of great significance and practical value. Fruit recognition technology with high accuracy and broad adaptability can not only accurately identify and classify different types of fruits, helping supermarkets and retail stores with automated fruit vending, but it also plays an important role in yield estimation, automated harvesting, and ripeness detection.

With the development of computer vision technology, various machine vision detection algorithms have been widely applied in fruit recognition and evaluation. GOEL et al. [1] used two color descriptors, the red–green difference and the red–green ratio, to detect different ripening stages of tomatoes, enabling the assessment of tomato ripeness. YU et al. [2] utilized color and texture features to train a random forest binary classification model, effectively distinguishing between different ripeness stages of lychees. LU et al. [3] used local binary pattern features and hierarchical contour analysis, utilizing texture and intensity distribution to identify immature citrus fruits. Traditional machine learning methods primarily rely on extracting features such as color, shape, and texture from fruit images to achieve fruit recognition and ripeness evaluation. However, since manually designed

features often struggle to handle complex scenes with mixed backgrounds and varying environmental factors, it is difficult for them to meet the needs of intelligent and automated fruit recognition and evaluation.

Deep learning-based models can automatically extract relevant features without manual intervention, making them more widely applicable in the agricultural field. JIA et al. [4] proposed an improved Mask R-CNN visual detector model, which enables the faster and more accurate recognition of overlapping apples. ZHANG et al. [5] improved the YOLOv4 model by adding coordinate attention modules to both the feature extraction module and the feature pyramid, which enhanced the accuracy of fruit recognition. GU et al. [6] introduced a transformer-based BRA sparse attention module into the backbone network of YOLOv8 and improved the detection head and feature fusion network to achieve mango recognition. LU et al. [7] improved the performance of the YOLOv4 model in apple ripeness assessment tasks by adding a convolutional attention mechanism module to the detector. ANANTHANARAYANA et al. [8] used the SSD network and MobileNetV2 network for the recognition and freshness assessment of three types of fruits: apples, oranges, and bananas. KANG et al. [9] utilized a Feature Pyramid Network and dilated spatial pyramid pooling to achieve apple detection in orchards. XUE et al. [10] proposed a fruit image classification framework, where the first part uses a convolutional autoencoder (CAE) for pretraining the images, and the second part employs an attention-based DenseNet to extract features from the images. CHEN et al. [11] aimed to detect citrus fruits and assess fruit ripeness. In the first stage, YOLOv5 was used to recognize citrus fruits in the images. In the second stage, an improved visual saliency detection algorithm was applied to generate saliency maps of the fruits, which combined RGB image information with saliency maps. A ResNet34 network was then used to determine the ripeness level of the fruits.

Deep learning classification models have gradually evolved from traditional CNN-based models (such as ResNet [12], EfficientNet [13], etc.) to Transformer-based classification models. Vision Transformer [14] directly applies the self-attention mechanism to the entire image, effectively handling long-range dependencies, but it comes with high computational cost. The Swin Transformer [15] classification model, based on a Transformer architecture with visual attention, is particularly well-suited for handling large-sized images and long-range information interactions. It introduces a hierarchical, window-based attention mechanism to process information at different scales, thereby enhancing the model's ability to understand images. The YOLO algorithm [16], based on deep learning, has the feature of single-stage object detection, which allows for the fast and accurate detection of different targets, making it highly suitable for fruit recognition and evaluation tasks. The improvements to the YOLO algorithm are primarily reflected in speed, accuracy, robustness, and flexibility. Through optimizations in network architecture and computational efficiency, YOLO has further enhanced detection speed. The introduction of new feature extraction networks, anchor box mechanisms, and multi-scale feature fusion techniques has significantly improved accuracy. Enhancements in robustness allow the model to better handle occlusion, complex backgrounds, and varying lighting conditions during detection tasks. Furthermore, YOLO offers multiple configuration options, making the model more flexible, allowing users to select the most suitable version based on their specific requirements and computational resources. These improvements have enabled the YOLO algorithm to exhibit greater adaptability and reliability in practical applications. Numerous studies have shown that enhancing the model's feature extraction and fusion capabilities can effectively improve the accuracy and stability of fruit recognition. The introduction of attention mechanisms allows the model to selectively focus on important, relevant features while ignoring less important information. HU et al. [17] proposed SE-Net, a channel attention mechanism consisting of three parts: compression, excitation, and

channel weight updating. First, global average pooling is used to compress the feature map of each channel. Then, a fully connected layer learns the correlation between the channels and uses an activation function (sigmoid) to generate the weight for each feature map group, enhancing the important features and suppressing the less important ones. Finally, the channel weight vector is updated by multiplying it with the original input. Considering that SE-Net performs mapping at a single scale, LI et al. [18] proposed SK-Net, which operates at multiple scales and enables the network to autonomously learn to select and fuse feature map information from different receptive fields. While SE-Net only considers the contribution of feature map channels, it neglects the fact that the spatial location of objects in an image also plays a crucial role in object detection. WOO et al. [19] proposed CBAM, which combines channel attention and spatial attention mechanisms in series, effectively improving the network's feature extraction and representation capabilities. However, although CBAM integrates channel attention and spatial attention mechanisms, it does not consider cross-dimensional interaction. MISRA et al. [20] established interactions from three branches: the width and height of the image, the dimension and width, and the dimension and height. The proposed CTAM (Cross-Transformer Attention Module) demonstrated outstanding performance. To avoid the loss of spatial information caused by 2D global pooling in SE-Net, HOU et al. [21] proposed CA, which decomposes channel attention into two parallel 1D feature encodings and embeds spatial position information into the channel attention. This approach allows the model to capture more information while avoiding significant overhead. Due to the large side effects of dimensionality reduction and the inefficiency of unnecessary channel interactions, WANG et al. [22] proposed ECA-Net, a local cross-channel interaction strategy that does not reduce dimensionality, maintaining performance while reducing model complexity.

Currently, most research on fruit recognition and evaluation is focused on several key areas: the precise recognition and evaluation of single fruit varieties, the detection and evaluation of multiple fruit types, and improving recognition capabilities in indoor and orchard environments. Tasks such as fruit ripeness assessment, quality grading, damage detection, and disease identification provide more comprehensive and accurate solutions for agricultural production and fruit quality control. In fruit recognition and classification tasks, single models are widely used due to their simple structure and ease of implementation. However, single models often suffer from limitations such as poor generalization ability and stability when recognizing multiple fruit types. To overcome these limitations, collaborative strategies involving multiple models have been gradually introduced. Compared to single models, the use of multiple models typically leads to significant improvements in performance. Multi-model strategies primarily include model ensemble, model fusion, and model collaboration. These approaches combine the strengths of independent models, further enhancing the system's robustness, accuracy, and generalization ability.

Many researchers have improved model performance from different angles to achieve more accurate recognition and evaluation. Currently, most studies focus on recognizing a few types of fruits, and once the data labels are determined, the model's recognition categories are restricted. In this paper, during the construction of the fruit recognition dataset, only a generalized "fruit" label is assigned. The YOLO-based detection model is improved to locate the fruit regions in the image. Then, a classification model is employed to perform detailed classification on the detected fruit regions. By separating the detection and classification tasks, the method achieves multi-model collaboration for better overall performance. Furthermore, when new fruit categories need to be added for recognition and evaluation tasks, only the classification model needs to be updated, reducing the labeling workload for detection. The detection model only needs to recognize the "fruit" target, allowing it to focus more on the position and shape features of the target, thereby reducing

misjudgments and omissions caused by category confusion. The classification model, on the other hand, can make finer category distinctions on the detected fruit images, improving recognition accuracy. Finally, for classification results with lower confidence, feature matching is used for calibration to enhance classification accuracy. Additionally, feature matching requires only a small number of samples to effectively address classification issues when new fruit categories appear. The contributions of this paper can be summarized as follows:

(1) This paper proposes a multi-model collaborative method for fruit recognition and evaluation, which separates the detection and classification tasks and optimizes the classification results through a feature matching network. The proposed method can achieve more accurate fruit recognition and is also suitable for fruit ripeness assessment.
(2) An attention-based fusion module is designed to achieve interactive fusion and enhancement of input features, which helps improve the performance and generalization capability of the YOLOv8 detection model in complex tasks.
(3) A classification prediction network is designed, where a Swin Transformer-based classification network works in collaboration with a cosine similarity-based feature matching network. This approach enhances classification accuracy while effectively addressing the classification issues of new categories.
(4) Ablation experiments and experimental results demonstrate the effectiveness of the proposed method, which can be applied to both indoor and outdoor fruit recognition and ripeness assessment.

2. Materials and Methods

2.1. Data Preparation

In the fruit recognition model, images were annotated using X-AnyLabeling, with 3000 images selected from the Fruit Recognition public dataset and 2000 images collected from orchard datasets, resulting in a total of 5000 manually annotated images. Each annotation set includes bounding box data and category information, where the bounding box data consists of the center coordinates, width, and height, and all annotated categories are labeled as "fruit". For the fruit classification model, fruits of the same type were placed in their respective folders, with each folder named after the fruit category. This paper constructs a fruit classification dataset containing 27 categories of fruits, with a total of 4832 images. The distribution of data for each fruit category is shown in Table 1. In addition, a tomato ripeness dataset with 898 images, an apple ripeness dataset with 1110 images, and an apple nondestructive testing dataset with 713 images have also been created. To achieve feature matching in images, a small sample dataset was constructed, containing 10 fruit categories with 5 images per category.

Table 1. The quantity distribution of each type of fruit.

Number	Category	Amount	Number	Category	Amount	Number	Category	Amount
1	Banana	181	10	Watermelon	186	19	Hami melon	188
2	Bayberry	186	11	Mango	165	20	Pineapple	189
3	Strawberry	191	12	Lemon	148	21	Durian	184
4	Coconut	192	13	Sugar oranges	177	22	Orange	188
5	Mangosteen	178	14	Green grapes	155	23	Red apples	177
6	Pomegranate	187	15	Pitaya	182	24	Lichee	194
7	Tomato	184	16	Longan	191	25	Grape	194
8	Pear	188	17	Green apples	188	26	Grapefruit	185
9	Cherries	161	18	Cherry tomatoes	191	27	Kiwi	187

2.2. Proposed Method

This study proposes a fruit recognition and evaluation model that separates the tasks of detection and classification and incorporates a feature matching network for post-processing. Initially, the YOLOv8 model is improved by adding an attention mechanism to the backbone network and using a self-developed attention-based fusion module in the neck network to accurately locate fruit positions and generate bounding boxes. Subsequently, the SwinT model is employed to classify the detected fruits and calculate classification confidence scores. To further calibrate the classification results and address potential errors introduced by new fruit categories, a feature matching method based on cosine similarity is adopted. For classification results with low confidence, calibration is performed through the feature matching network. Finally, the detection and classification results are integrated to produce the final output. The overall framework is shown in Figure 1.

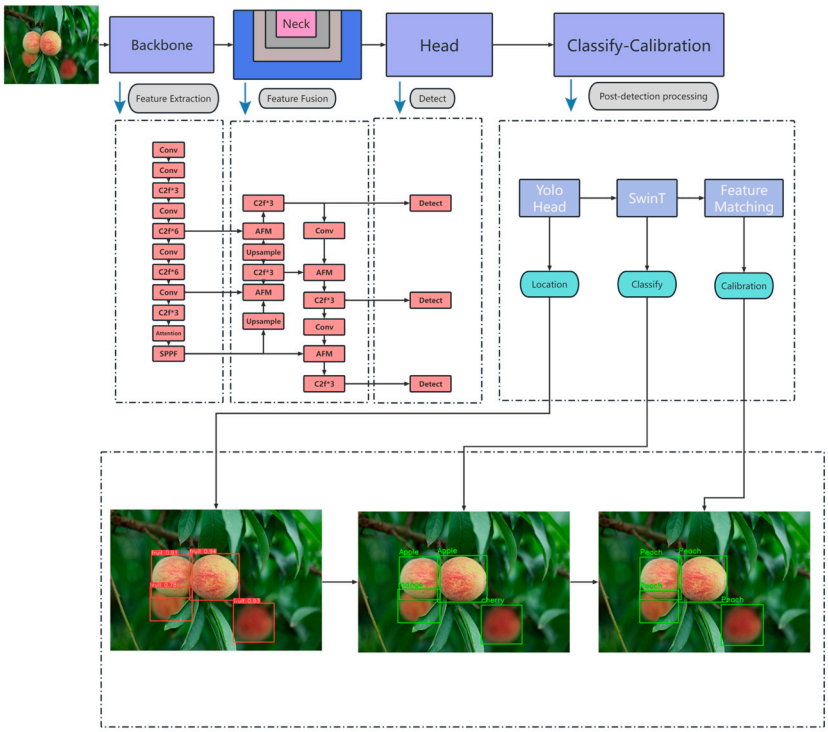


Figure 1. Overall framework.

2.2.1. Detection Model Based on Improved YOLOv8

The YOLOv8 network is primarily composed of three parts: the backbone network, the neck network, and the detection head. The backbone network extracts features from the input images and includes structures such as Conv, C2f, and SPPF. The Conv structure is used to extract basic features from the images; the C2f structure reduces feature redundancy and enhances feature extraction capabilities; and the SPPF structure reduces redundant feature extraction within the network, thereby speeding up the generation of candidate boxes. The neck network comprises structures like Upsample, Concat, C2f, and Conv, and it utilizes the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN). The FPN performs feature fusion through top-down upsampling techniques, while the PAN transmits spatial information in a bottom-up manner. The detection head uses loss functions and Non-Maximum Suppression (NMS) to output the target's category and confidence score. It also employs regression techniques to handle the candidate boxes, determining the precise location and size of the targets.

To enhance the feature extraction capability of the model, TripletAttention was added before the SPPF structure in the backbone network, thereby improving the network's

feature extraction ability and achieving structural improvements. For better feature fusion, an attention-based fusion module (AFM) was designed and implemented, replacing the original Concat module with AFM, which further improved the model. The framework of the improved YOLOv8 is shown in Figure 2.

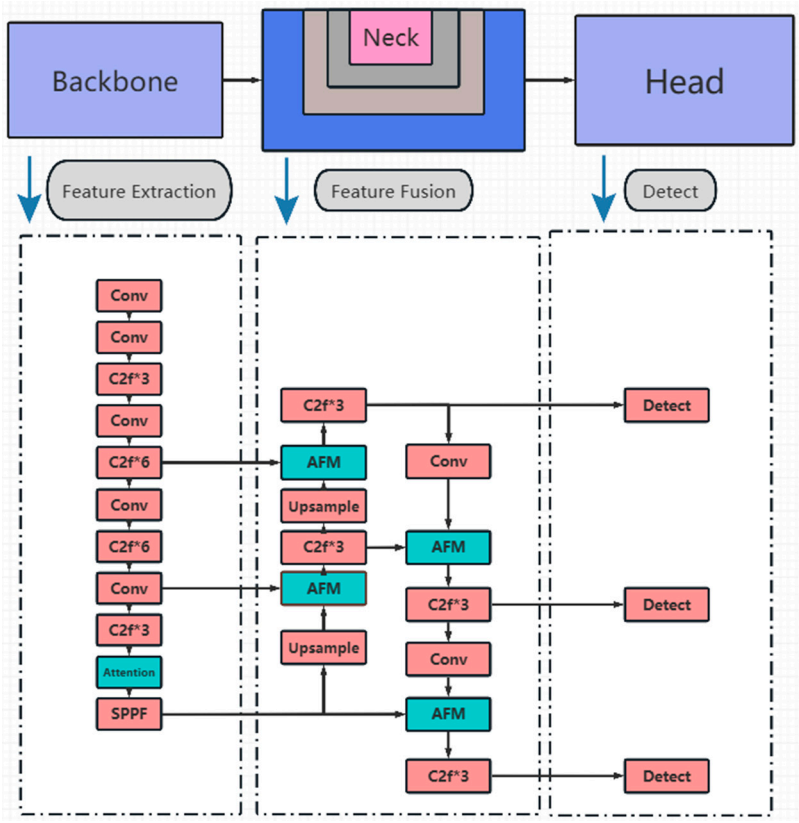


Figure 2. Framework diagram of improved YOLOv8.

TripletAttention is a lightweight and effective attention mechanism that calculates attention weights by capturing cross-dimensional interactions through a three-branch structure. TripletAttention establishes interactions from the width and height, dimension and width, and dimension and height branches of an image, thereby capturing cross-dimensional relationships. In each branch, the input tensor undergoes rotation operations (e.g., 90° counterclockwise rotation) and residual transformations to establish dependencies between different dimensions. When computing attention weights, TripletAttention uses lightweight methods such as Z-Pool and smaller convolutional kernels, resulting in low computational overhead. The TripletAttention structure is shown in Figure 3.

The Concat structure in the neck network directly concatenates extracted features, treating all features as equally important and failing to achieve information interac-tion between different dimensions. In this paper, we propose an Attention Fusion Module (AFM) based on attention mechanisms, and the specific structure is shown in Figure 4. The AFM structure first adjusts the channel number of the input feature ten-sor and concatenates features along the channel dimension. Next, it adds an attention mechanism to enhance the network's focus on different parts of the features, improv-ing the model's perceptual and expressive abilities. Finally, through feature separation and fusion, the weights of useful features are enhanced, the weights of unimportant features are reduced, and information exchange between different features is achieved. This effectively filters out noise and redundant information, thereby enhancing the network's feature representation and generalization capabilities.

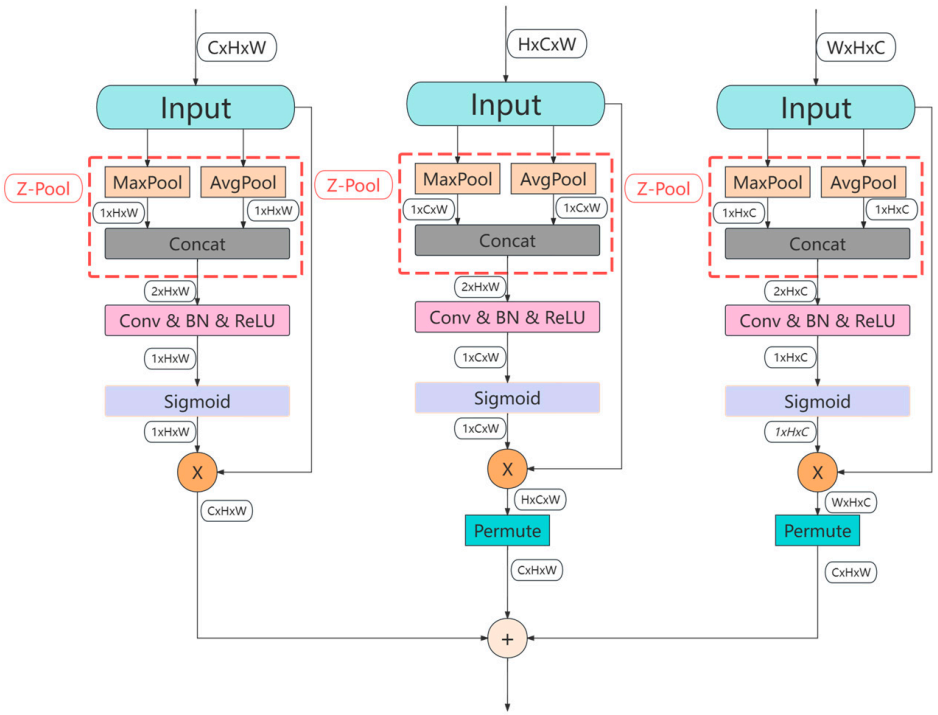


Figure 3. TripletAttention structure.

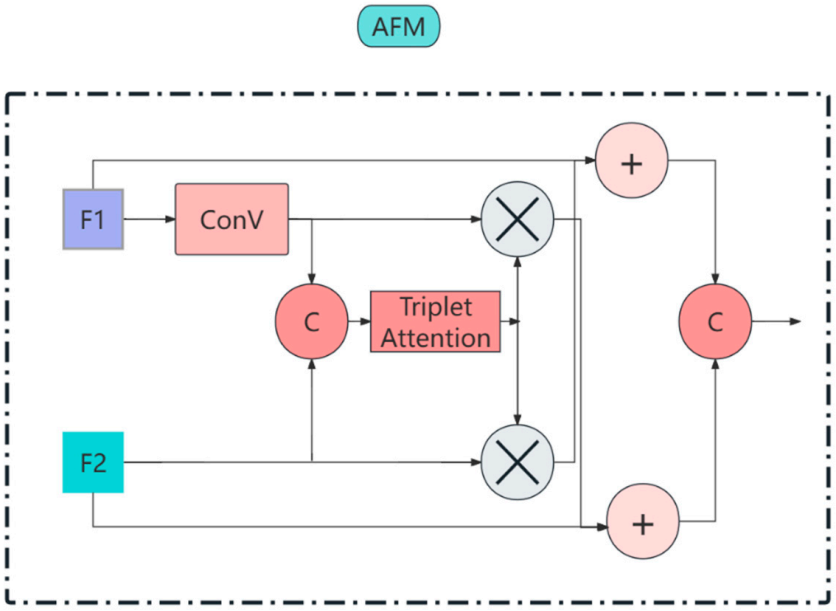


Figure 4. Attention-based fusion module (AFM).

2.2.2. Classification Model with Swin Transformer

The Swin Transformer classification model adopts a hierarchical design, consist-ing of 4 stages. First, the Swin Transformer splits the image through Patch Partition and then adjusts the channel number via Linear Embedding. Except for the first stage, each stage reduces the resolution of the input feature map through the Patch Merging layer, performing downsampling to gradually expand the receptive field for acquiring global information. The Swin Transformer block mainly consists of LayerNorm, Win-dow Attention, Shifted Window Attention, and MLP. The introduction of Shifted Win-dow Attention allows better interaction in each layer, efficiently capturing both local and global features, as shown in Figure 5 illustrating the Swin Transformer architec-ture and Figure 6 depicting the Swin Transformer block.

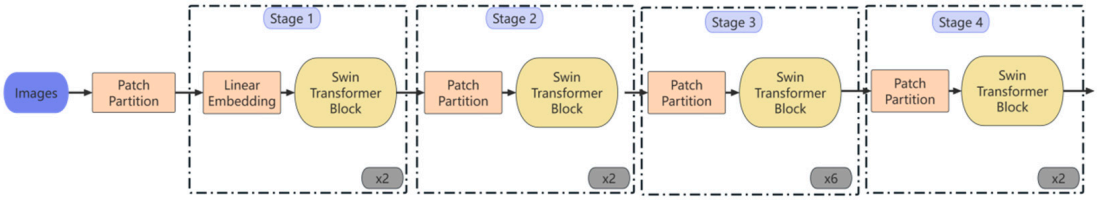


Figure 5. Swin Transformer architecture.

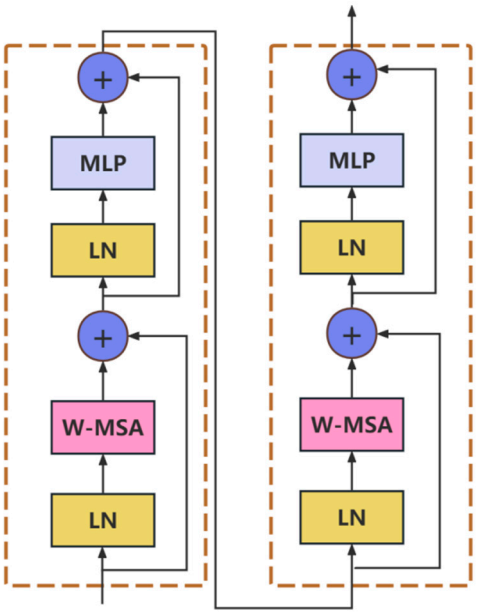


Figure 6. Swin Transformer block.

2.2.3. Cosine Similarity-Based Feature Matching Network

When using a classification model to recognize images, a low confidence score often indicates a higher risk of misclassification. In such cases, even images that are known and explicitly labeled as a specific fruit category may be misclassified by the model as belonging to another category. Additionally, for categories that the model has not encountered or learned during training, referred to as "unknown categories", the accuracy and confidence of classification are even harder to guarantee, further increasing the likelihood of misclassification. Therefore, it is crucial to focus on the model's low-confidence outputs and calibrate the classification results to improve overall performance and reduce misclassifications.

Given that image feature vectors are high-dimensional, this study employs cosine similarity for the rapid calibration of classification results. Cosine similarity is effective in handling high-dimensional and sparse data and is computationally efficient. Thus, we use cosine similarity to measure the similarity between two images. Cosine similarity is calculated by measuring the cosine of the angle between two vectors, with the formula

$$Cosine\ similarity\ = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

The CLIP [23] training dataset consists of 400 million text-image pairs, endowing it with powerful image feature extraction capabilities and Few-Shot and Zero-Shot learning abilities. To extract image features, CLIP is employed to extract features from the image to be calibrated, and these features are normalized. By converting the image into tensor format, PyTorch's computational and model inference capabilities can be fully utilized. Secondly, features are extracted from each reference category's set of reference images, normalized, and stored. Establishing the reference feature library only requires storing

images in folders named after their respective categories. Finally, cosine similarity is computed through dot product between the feature of the image to be calibrated and all reference category features to identify the category with the highest similarity score, thereby predicting its class.

3. Experiments and Results

3.1. Experimental Environment and Training Parameter Settings

The experiment is trained under the pytorch framework, and the experimental environment configuration is shown in the Table 2 below.

Table 2. Experimental environment configuration.

Configuration Name	Enviromental Parameter
CPU	13th Gen Intel(R) Core(TM) i9-13900KF
GPU	NVIDIA GeForce RTX 4080, 16375MiB
Memory	128 G
Python	3.7.16
Torch	1.13.1
CUDA	11.6

The training parameters for the improved YOLOv8 detection model are detailed in Table 3, and the training parameters for the Swin Transformer classification model are provided in Table 4.

Table 3. YOLOv8 detection model training parameters.

Parameter	Setting
Batch Size	64
Learning Rate	0.002
Epochs	100
Pretrained Weights	No (Training from scratch)
Dataset Split	80% training, 20% test
Momentum	0.9
Data Caching	Yes
Optimizer	AdamW
Device	CUDA
Workers	8

Table 4. Swin Transformer detection model training parameters.

Parameter	Setting
Batch Size	32
Learning Rate	0.0001
Epochs	100
Pretrained Weights	Yes
Dataset Split	80% training, 20% test
Optimizer	AdamW
Weight Decay	5E-2
Device	CUDA
Workers	8
Fine-tuning Layers	Fine-tuned all layers

3.2. Evaluation Index

3.2.1. Evaluation Index of YOLOv8 Detection Model

To validate the effectiveness of the improved models, Precision (P), Recall (R), mAP50 (mean Average Precision with IoU = 0.5), and mAP50-95 (mAP with IoU thresholds from

0.5 to 0.95) were utilized. IoU is used to measure the overlap between the candidate boxes generated by the model and the ground truth boxes. A higher IoU value indicates greater similarity between the boxes. Precision measures the accuracy of fruit detections, ensuring that identified objects are indeed fruits. Recall assesses how well the model detects all instances of fruits present in the images.

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{2}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{3}$$

where TP represents the number of correctly predicted positive samples; FP is the number of false positive samples. FN represents the number of mispredicted negative class samples.

$$\mathrm{AP} = \frac{1}{\mathrm{m}} \sum_{r=1}^{m} (P(r)\Delta R(r)) = \int_0^1 P(R)dR \tag{4}$$

where m is the number of positive samples, P(r) is the proportion of positive samples in the first r retrieval processes, $\Delta \mathrm{R}(r)$ is the change in Recall with respect to r in the first r retrieval results, and P(R) represents the Precision (P) under feature recall (R).

$$\mathrm{m}AP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{5}$$

Here, n is the total number of categories and AP_i represents the AP value of the ith class.

mAP50 evaluates the Average Precision (AP) of each class based on candidate boxes generated by the model that have an IoU (Intersection over Union) of 0.5 or greater with the ground truth boxes. It calculates the AP for each class and then averages these values to obtain mAP50.

mAP50-95 extends this evaluation by considering a series of confidence thresholds (typically from 0.5 to 0.95, with a 0.05 interval). For each confidence threshold, it computes the AP for each class, where predictions with IoU greater than or equal to that threshold are considered correct predictions. Finally, it averages the AP values across all classes and thresholds to derive mAP50-95.

3.2.2. Evaluation Metrics for Swin Transformer Classification Models

Using accuracy and average loss as the primary metrics to evaluate the model. Accuracy refers to the proportion of samples correctly classified by the model out of the total number of samples, calculated as:

$$\text{Accuracy} = \frac{\mathrm{CS}}{\mathrm{TS}} \tag{6}$$

Here, CS represents the number of correctly predicted samples and TS represents the total number of predicted samples.

Loss calculation is typically performed using the Cross-Entropy Loss function, which is computed and accumulated over the entire training dataset, then averaged by the total number of iterations to yield the average loss. Cross-Entropy Loss is a widely used loss function that effectively measures the inconsistency or discrepancy between the model's predicted distribution and the true data distribution. The formula for Cross-Entropy Loss is

$$\mathrm{L}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{c} y_{ic} \log(\hat{y_{ic}}) \tag{7}$$

where C represents the number of categories classified, N is the number of samples, $y_{ic} = 0$ if sample i does not belong to class c, and equal to 1 otherwise. $\log(\hat{y_{ic}})$ is the probability distribution that the model predicts that sample i belongs to each class and is a vector of length C, where all elements of the vector sum to one.

3.3. Ablation Experiment of Improved YOLOv8 Detection Model

To further validate the effectiveness of the improved models, ablation experiments were conducted. The experiments compared the performance of models a, b, c, and d. Model a represents the YOLOv8 baseline algorithm. Model b incorporates TripletAttention into the backbone network of the baseline algorithm. Model c replaces the Concat structure in the neck network of the baseline algorithm with the AFM structure. Model d combines the additions of TripletAttention in the backbone network and the replacement of Concat with AFM in the neck network of the baseline algorithm. Performance evaluation metrics include Precision (P), Recall (R), mean Average Precision at IoU = 0.5 (mAP50), and mean Average Precision from IoU = 0.5 to 0.95 (mAP50-95). Detailed results of the ablation experiments are shown in the Table 5 below:

Table 5. Ablation experiments.

Model	Method	P	R	mAP50	mAP50-95	Parameters
a	YOLOv8 (baseline)	0.945	0.878	0.947	0.789	3,005,843
b	YOLOv8+TripletAttention	0.966	0.883	0.956	0.804	3,006,143
c	YOLOv8+AFM	0.957	0.884	0.956	0.800	3,110,019
d	YOLOv8+AFM+TripletAttention	0.969	0.899	0.957	0.802	3,110,319

The results indicate that incorporating attention mechanisms into the backbone network and replacing the Concat structure in the neck network with AFM structure improved the detection model compared to the baseline model by 2.4% in Precision (P), 2.1% in Recall (R), 1% in mAP50, and 1.3% in mAP50-95. Additionally, the number of parameters increased from 3,005,843 to 3,110,319. The specific training process is shown in Figure 7.

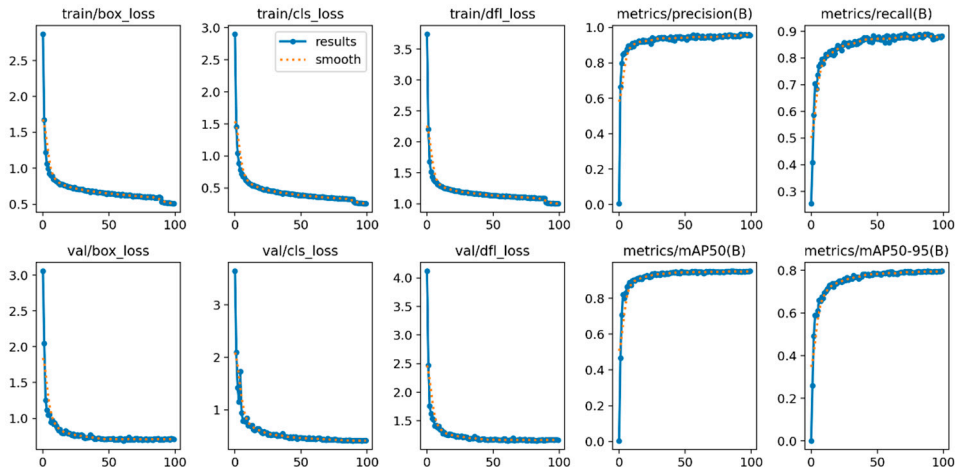
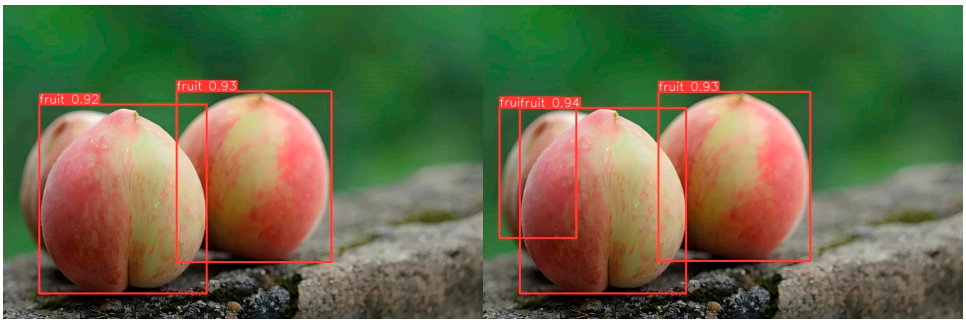


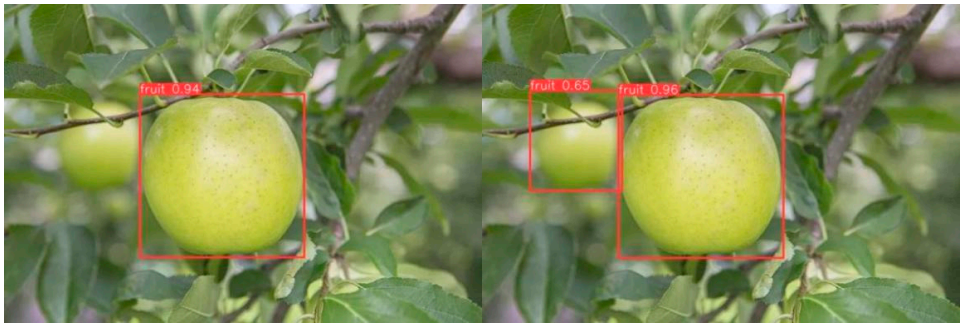
Figure 7. Training plot of improved YOLOv8 detection model.

The comparison shows that the improved model can reduce instances of missed detections and decrease the frequency of false positives, especially in scenarios with more occlusions and complex environments. Detailed experimental results validating this outcome are provided in Figures 8–10.



(**a**) YOLOv8 (baseline) (**b**) YOLOv8+AFM+TripletAttention

Figure 8. The improved model has fewer missed detections in the case of more occlusions.



(**a**) YOLOv8 (baseline) (**b**) YOLOv8+AFM+TripletAttention

Figure 9. The improved model has fewer missed detections in the fuzzy case.

(**a**) YOLOv8 (baseline) (**b**) YOLOv8+AFM+TripletAttention

Figure 10. The improved model has fewer missed detections and fewer false detections than the original mode.

3.4. Training the Swin Transformer Classification Model

To improve the model's classification accuracy, several steps were taken: first, data augmentation techniques such as random cropping and horizontal flipping were applied, and images were normalized to ensure they met the model's input requirements. Secondly, pre-trained weights were loaded and certain layers were frozen as needed to fine-tune the

model. Finally, iterative parameter optimization was conducted to gradually refine the model, assessing its generalization ability on the validation set.

The fruit classification model was trained on a total of 4362 images across 27 fruit categories, with 3501 images used for training and 861 images for validation. During training, the model achieved a loss of 0.088 and a classification accuracy of 97.5%. After 100 epochs, the loss on the validation set increased to 0.278, with a classification accuracy of 92.6%. The specific training process is illustrated in Figure 11. The model was further tested on indoor and outdoor images of pineapple, grape, orange, and lychee fruits, achieving high classification accuracy, as shown in Figure 12.

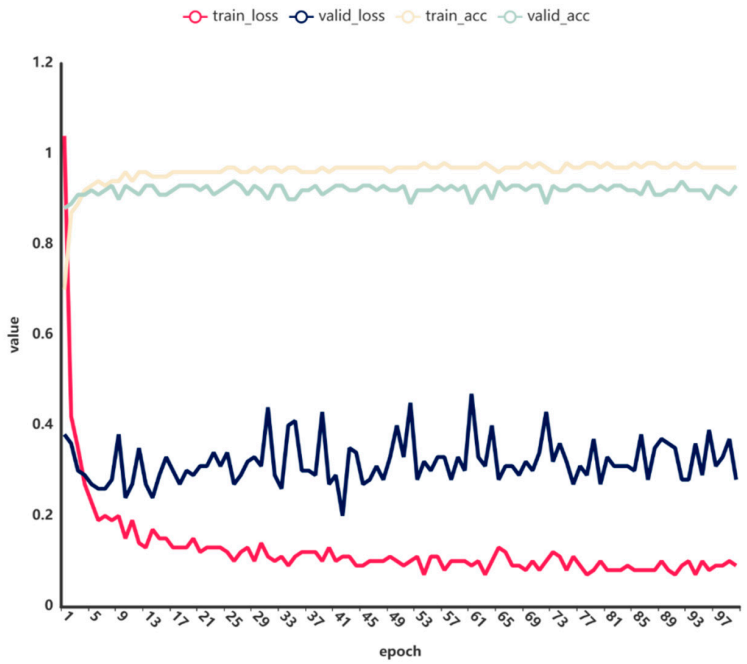


Figure 11. Swin Transformer classification model training.

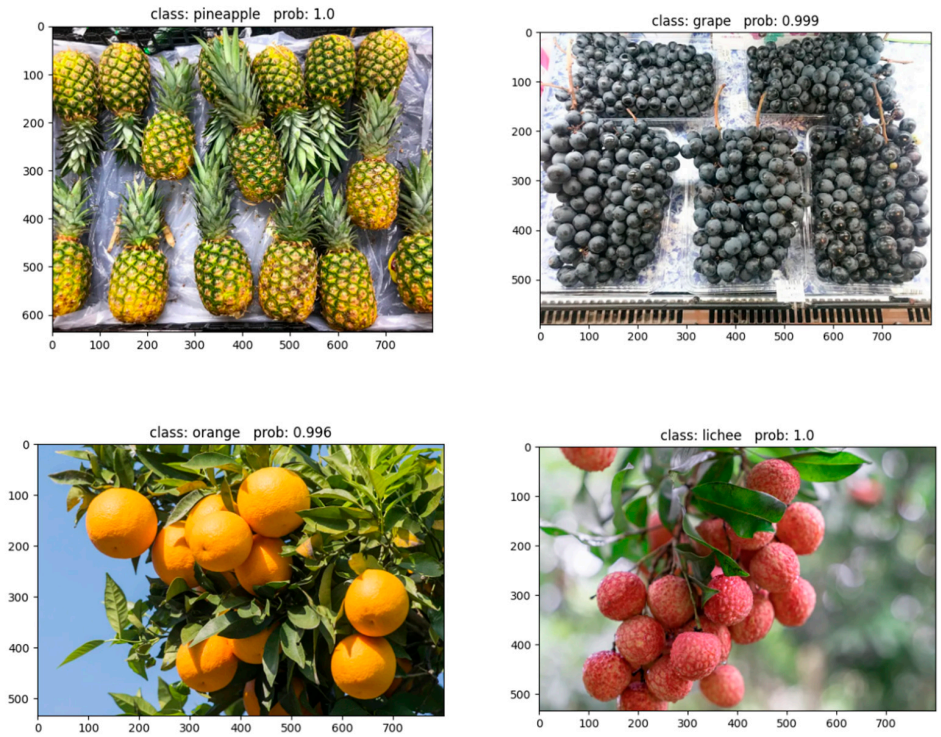


Figure 12. Swin Transformer classification model prediction.

3.5. Result and Analysis

The YOLO detection model excels in the field of object detection due to its advantages in real-time performance and accuracy. On the other hand, the Swin Transformer classification model, with its hierarchical structure, self-attention mechanism, and shifted window partitioning strategy, enhances its ability to comprehend complex images and capture long-range dependencies within them. By integrating these models, an efficient fruit recognition and evaluation system can be realized, accurately pinpointing fruit locations and identifying their categories. The results of the integrated model are shown in Figure 13.

Figure 13. Fruit recognition result.

To further validate the superiority of the proposed framework, this study examines three scenarios: detecting untrained fruit categories by the detection model, encountering undefined fruit categories by the classification model, and applying maturity detection. These scenarios assess the model's transferability and generalization capabilities effectively.

3.5.1. Experiment on Detecting Untrained Fruit Categories

Circular and elliptical fruit shapes constitute a significant proportion in fruit forms. To validate the model's generalization ability, this study selected passion fruit, which was not included in the target detection model, to verify the recognition of similar circular and elliptical fruits. Additionally, irregularly shaped bananas, which were not previously detected, were successfully identified, further demonstrating the model's strong generalization capability, as shown in Figure 14.

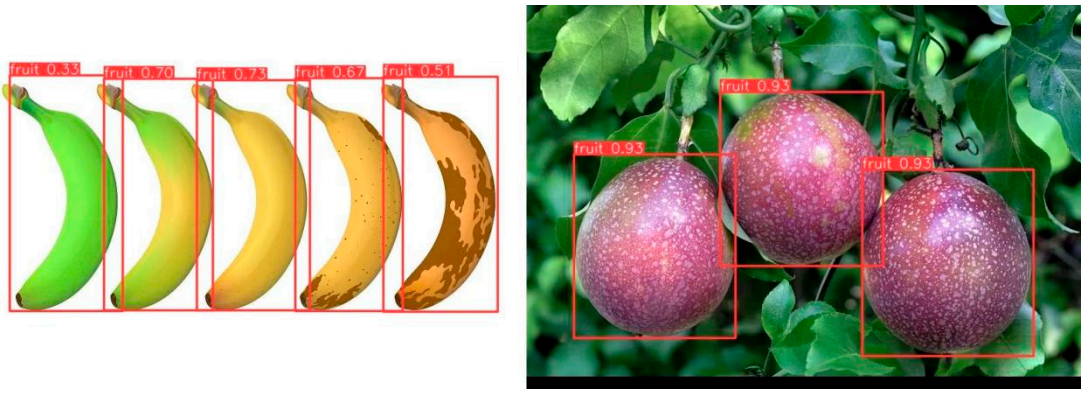


Figure 14. Results of an untrained fruit category detection experiment.

3.5.2. Experiment on Classifying Untrained Fruit Categories

When the detected image belongs to a fruit category unknown to the classification model, its confidence level is low. The classification model outputs the category most similar to those defined, leading to potential misclassifications. In such cases, a feature matching network based on cosine similarity adjusts the classification based on reference categories, as shown in Figure 15.

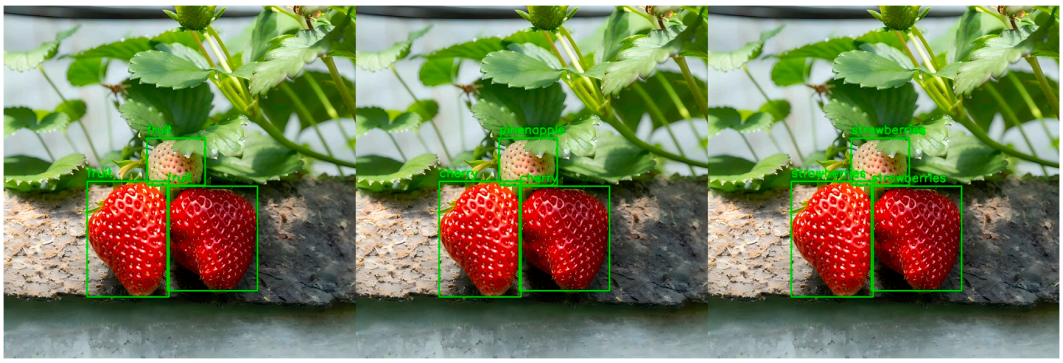


Figure 15. Results of an experiment on fruit classification without training.

3.5.3. Fruit Evaluation Experiment

Based on the apple skin color and market demand, maturity is divided into three categories: high maturity (mostly red skin), medium maturity (red and green mixed skin), and low maturity (mostly green skin). Following this classification, an apple maturity dataset was constructed for model training. By replacing the fruit classification model with an apple maturity classification model, fruit maturity detection can be performed, as shown in Figure 16.

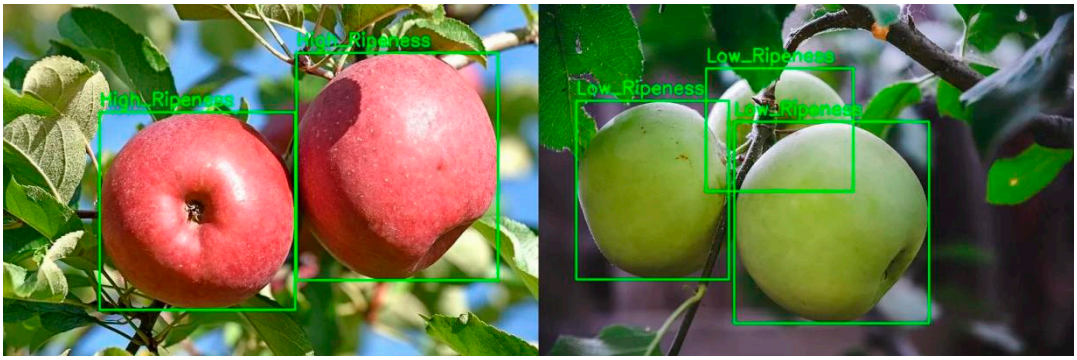


Figure 16. Apple maturity detection.

During the growth process of tomatoes, once the fruit has set, the skin is initially white-green during the unripe stage. It starts turning yellow during the semi-ripe stage and finally becomes red at the ripe stage. Tomato maturity can be classified into three stages based on skin color: unripe, semi-ripe, and ripe. A maturity classification dataset can be constructed based on these skin color distinctions. As shown in Figure 17, tomato maturity can be effectively detected.

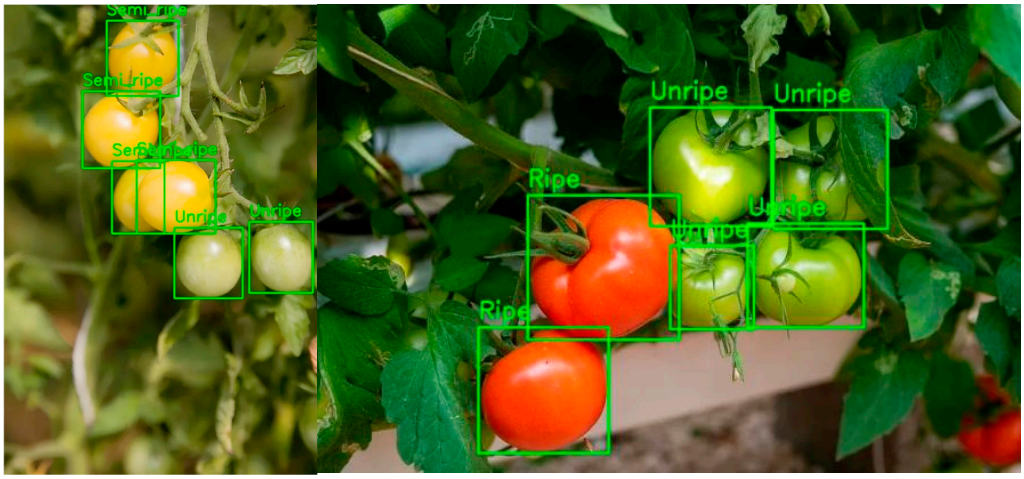


Figure 17. Tomato maturity test.

In order to further realize the evaluation of fruits, the images of apples are divided into two categories: damaged, diseased, or rotten apples and lossless apples. By training the Swin Transformer classification model, the model can classify whether there is damage, disease, or decay on the surface of apple images. As shown in Figure 18, nondestructive testing of apples is implemented.

Figure 18. Nondestructive apple testing.

4. Conclusions

In this paper, detection and classification tasks are handled separately and post-processed using a feature matching network. The model can choose different detection and classification models based on practical needs, allowing for independent optimization to achieve the best overall performance. Since the detection model only needs to identify the broad category of "fruit", it can focus more on the target's location and shape features, reducing misclassification and missed detection due to category confusion. The classification model can then make finer distinctions within the detected fruit areas, improving recognition accuracy. The feature matching network can correct classification results and expand the recognized categories when new ones appear.

When new fruit types need to be added, only the classification model needs to be updated, without retraining the detection model. This modular design makes system expansion easier and more efficient, and it can be extended to maturity detection applications.

The methods proposed in this paper exhibit good generalization and transferability in fruit detection, effectively reducing the time required for data annotation and improving model accuracy. However, recognition is limited when the fruit shape is irregular. When applied to maturity detection, the model performs well for categories with significant differences in skin color but is less effective when the skin color is unusual.

The improved YOLOv8 model P, R, mAP50, and MAP50-95 are 2.4%, 2.1%, 1%, and 1.3% higher than the baseline model, respectively. It can realize the location of most fruits with circular and oval shapes. The Swin Transformer classification model can realize the classification of 27 types of fruits. The ripeness test of two fruits and the nondestructive test of one fruit are realized.

Multi-model collaboration can significantly enhance the overall performance of the model, particularly in terms of accuracy and robustness. However, this process inevitably increases the computational overhead, leading to a decrease in processing speed. Specifically, this performance degradation is primarily due to the computation involved with multiple models and the complex feature fusion process. To address this challenge, future work will focus on optimizing algorithms and leveraging hardware acceleration to alleviate the computational bottlenecks introduced by multi-model collaboration, thereby improving the model's running efficiency while maintaining high performance.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: M.H. and D.F.; data collection: M.H. and D.C.; analysis and interpretation of results: M.H.; draft manuscript preparation: M.H. and 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 raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: I would like to thank all the teachers and students in the laboratory for your help and valuable suggestions during the research process. The cooperation and discussion with you greatly promoted the progress of the research, and thank you for your efforts and support.

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

References

1. Goel, N.; Sehgal, P. Fuzzy classification of pre-harvest tomatoes for ripeness estimation—An approach based on automatic rule learning using decision tree. *Appl. Soft Comput.* **2015**, *36*, 45–56. [CrossRef]
2. Yu, L.; Xiong, J.; Fang, X.; Yang, Z.; Chen, Y.; Lin, X.; Chen, S. A litchi fruit recognition method in a natural environment using RGB-D images. *Biosyst. Eng.* **2021**, *204*, 50–63. [CrossRef]
3. Lu, J.; Lee, W.S.; Gan, H.; Hu, X. Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis. *Biosyst. Eng.* **2018**, *171*, 78–90. [CrossRef]
4. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Lian, J.; Zheng, Y. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [CrossRef]
5. Zhang, C.; Kang, F.; Wang, Y. An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds. *Remote Sens.* **2022**, *14*, 4150. [CrossRef]
6. Gu, Z.; He, D.; Huang, J.; Chen, J.; Wu, X.; Huang, B.; Dong, T.; Yang, Q.; Li, H. Simultaneous detection of fruits and fruiting stems in mango using improved YOLOv8 model deployed by edge device. *Comput. Electron. Agric.* **2024**, *227*, 109512. [CrossRef]

7. Lu, S.; Chen, W.; Zhang, X.; Karkee, M. Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. *Comput. Electron. Agric.* **2022**, *193*, 106696. [CrossRef]
8. Ananthanarayana, T.; Ptucha, R.; Kelly, S.C. Deep Learning based Fruit Freshness Classification and Detection with CMOS Image sensors and Edge processors. *Electron. Imaging* **2020**, *32*, 2352. [CrossRef]
9. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105108.
10. Xue, G.; Liu, S.; Ma, Y. A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex Intell. Syst.* **2023**, *9*, 2209–2219. [CrossRef]
11. Chen, S.; Xiong, J.; Jiao, J.; Xie, Z.; Huo, Z.; Hu, W. Citrus fruits maturity detection in natural environments based on convolutional neural networks and visual saliency map. *Precis. Agric.* **2022**, *23*, 1515–1531. [CrossRef]
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
13. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
14. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
15. 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, Montreal, BC, Canada, 11–17 October 2021.
16. Redmon, J. 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.
17. 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.
18. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
19. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
20. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to attend: Convolutional triplet attention module. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021.
21. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
22. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
23. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.