Journal of Electronic Research and Application, 2025, Volume 9, Issue 5
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Persimmon Fruit Quality Grading Detection Based on an Improved YOLOv8s Lightweight Model

Haogang Wang[1], Yunge Jing[2]*

[1]School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, Shanxi, China
[2]School of Mathematics and Information Technology, Yuncheng University, Yuncheng 044000, Shanxi, China

**Author to whom correspondence should be addressed.*

Abstract: Addressing challenges in accurately detecting persimmon fruit quality in orchards—such as reliance on manual grading, low efficiency, severe foliage obstruction, and subtle differences between quality grades—this paper proposes a lightweight persimmon detection model based on an improved YOLOv8s architecture. First, the Conv layer in the backbone network is replaced with an ADown module to reduce model complexity. Second, MSFAN is introduced in the Neck layer to fully extract texture features from persimmon images, highlighting differences between quality grades. Finally, the Wise-IoU loss function mitigates the impact of low-quality sample data on grading accuracy. The improved model accurately identifies and separates persimmons of varying quality, effectively addressing quality grading detection in complex backgrounds. This provides a viable technical approach for achieving persimmon quality grading detection.

Keywords: Persimmon quality grading; YOLOv8; Deep learning; Lightweight; Image detection

Online publication:

1. Introduction

Persimmon (*Diospyros kaki*) is a deciduous fruit tree belonging to the genus *Diospyros* within the family Ebenaceae. Native to China, it has been cultivated for over 2,000 years. As one of the world's four ancient fruit trees, persimmon holds significant economic and cultural value in East Asia [1–3]. Currently, persimmon is widely distributed across China, Japan, the Republic of Korea, and other countries, with China leading globally in both cultivation area and production volume.

Machine vision has demonstrated unprecedented development prospects in agricultural applications [4–6]. For a long time, fruit quality grading has primarily relied on manual visual inspection. This traditional method has significant limitations: judgment criteria vary among individuals, detection accuracy is low, and the workload is excessively high [7]. Advancements in image processing and machine learning have enabled researchers to integrate traditional image processing with machine learning for fruit quality assessment [8–12]. However, in natural environments, severe occlusion and overlapping among multiple fruits, coupled with partial coverage by

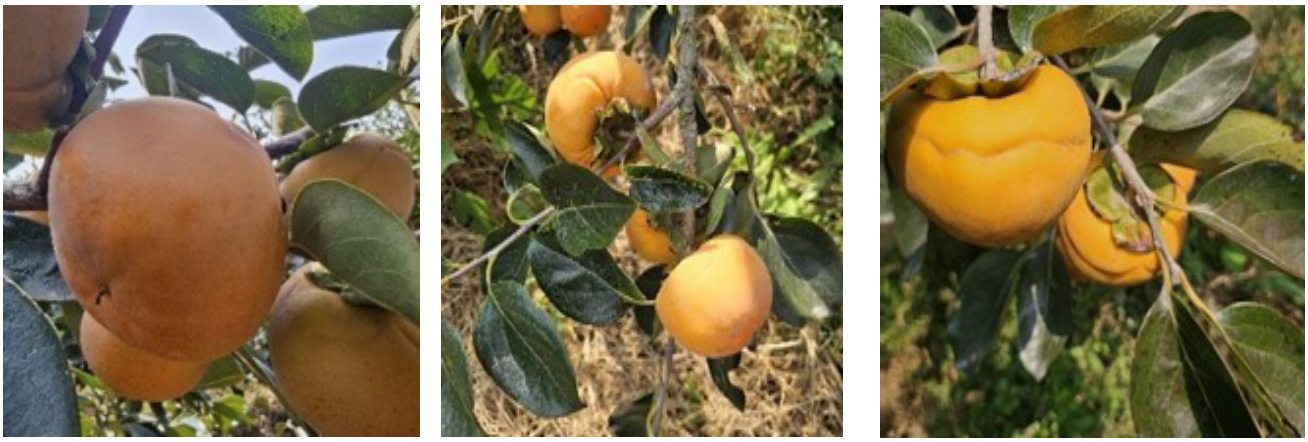
branches and foliage, pose significant challenges for target fruit detection, recognition, and spatial localization. Deep learning technology, with its powerful capability for autonomous image feature extraction, has been widely applied across various branches of computer vision [13]. Integrating object detection algorithms with agricultural product sorting has become a growing trend. Liu [14] proposed a Faster-YOLO-AP algorithm for apple detection, realized the lightweight of the model, and improved the model's mAP@50:95 to 84.12%. Zhang *et al.* [15] proposed a lightweight apple recognition method based on an improved YOLOv7 model. By introducing an ECA module to enhance detection performance, they reduced the number of parameters and computational complexity by 22.93% and 27.41%, respectively, compared to the original model. Qiu *et al.* [16] employed CNN and image segmentation to achieve automated detection and identification of surface damage on fragrant pears, achieving an accuracy rate of 91%. Sun [17] proposed an optimized residual neural network-based grading method for apple appearance quality. Using ResNet-34 as the base network, the standard convolutions within residual units were replaced with grouped convolutions to reduce the model's parameter count and computational complexity.

Although significant progress has been made in fruit quality grading algorithms, practical applications still face challenges. Some fruits exhibit insufficiently distinct grading characteristics, leading to misclassification. Conventional deep learning models struggle to extract shallow-level details, limiting overall detection accuracy. The negative impact of low-quality training samples weakens the model's generalization performance. To address these issues, this paper proposes a fruit persimmon quality grading detection model based on an improved YOLOv8s (YOLOv8s-MSFAN).

2. Materials and methods

2.1. Dataset construction

This study selected persimmons as its research subject, with images captured at orchards in Linyi County, Yuncheng City, Shanxi Province. The collection device was an OPPO smartphone, saving images in JPG format at a resolution of 4000×1800. During data collection, particular emphasis was placed on scene diversity and representativeness, capturing the dataset under varying lighting conditions and shooting angles. A total of 1,200 images were collected. After removing unqualified samples—such as blurred or disordered images—the final dataset comprised 950 images. A portion of the dataset is illustrated in **Figure 1**.



upward shot aerial view front lighting

Figure 1. Example of persimmon fruit image acquisition

To enhance the model's noise resistance, random adjustments were made to contrast and brightness, and Gaussian noise was added, resulting in a total of 1,054 images. The image samples were categorized into two classes: normal and sunken. Manual dataset annotation was performed using the open-source annotation software LabelImg, generating a text file. The dataset was then split in a 7:1:2 ratio, yielding a training set of 730 images, a validation set of 105 images, and a test set of 209 images.

2.2. YOLOv8 network model improvements

To address issues such as weak interference resistance and poor information extraction in the YOLOv8s model, this paper enhances the model's perception and capture capabilities for positional information through three aspects: the backbone network, neck network, and bounding box regression loss function. It also optimizes the multi-scale feature fusion strategy to improve the model's ability to extract deep feature information and reduce redundant feature information. The improved YOLOv8s-MSFAN model is illustrated in **Figure 2**, with the specific enhancement approach outlined below:

(1) Replacing the Conv convolutions in the trunk and neck networks with the ADown module maximizes the retention of important image features while reducing parameters. This effectively addresses feature selection and preservation in occlusion scenarios, preventing significant loss of spatial information.

(2) To address information loss and redundancy during shallow feature extraction in persimmon quality grading, we incorporate a Multi-Scale Feature Aggregation Network (MSFAN).

(3) To overcome the limitations of CIoU loss function in fruit grading tasks—such as insufficient detection capability and interference from low-quality samples—we replace CIoU with Wise-IoU featuring a dynamic scaling mechanism, thereby increasing the loss for difficult-to-classify samples.

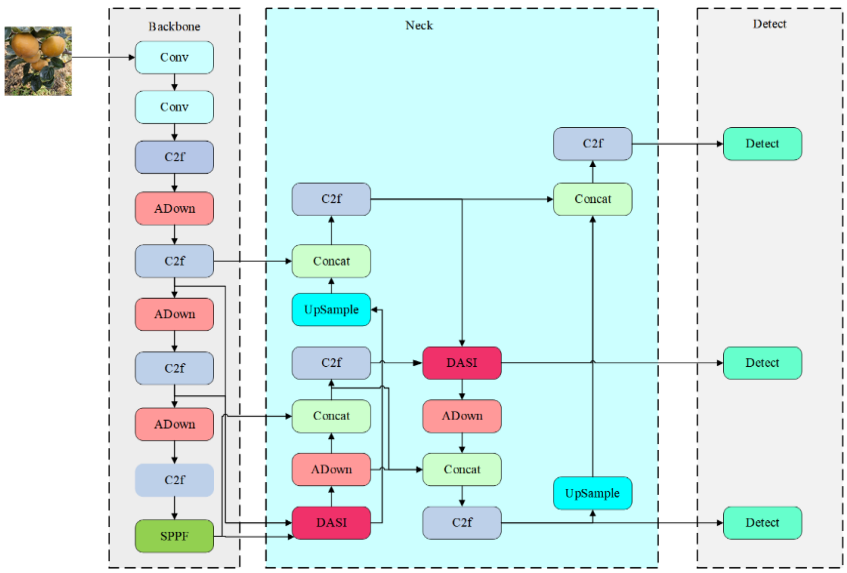


Figure 2. YOLOv8-MSFAN network model architecture diagram

2.2.1. ADown module

The YOLOv8 model extracts multi-scale features through convolutional operations with a stride of 2, but its computational burden is heavy due to the lack of an efficient downsampling mechanism. To address this issue, this study introduces the ADown [18] subsampling module. By combining average pooling and max pooling, it effectively balances feature compression with information loss while significantly reducing computational

complexity. The method first applies 2D average pooling to reduce the spatial sensitivity of feature maps with C input channels. The processed images are then split into two feature maps, each with C/2 channels. Next, one set of feature maps undergoes 3×3 convolution before output, while the other set undergoes 2D max pooling followed by 1×1 convolution before output. Finally, these two outputs are combined to produce a feature map with C channels. The architecture is illustrated in **Figure 3**.

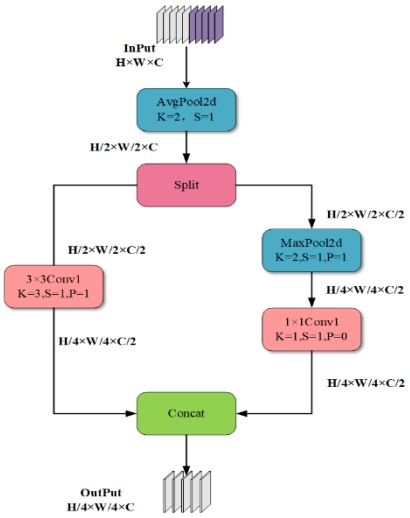


Figure 3. ADown module diagram

H, W, and C represent the height, width, and number of channels of the feature map, respectively; K denotes the size of the convolution kernel; S is the stride; P is the padding. AvgPool2d is the average pooling layer; MaxPool2d is the max pooling layer.

2.2.2. MSFAN module

Traditional methods lack adaptive mechanisms, failing to dynamically adjust features across scales based on different scenarios and target characteristics, resulting in suboptimal fusion performance. Moreover, in complex environments, leaf occlusion often obscures fine details of targets, compromising the detection accuracy of small objects. To address these issues, this paper proposes the MSFAN module. Its network architecture is illustrated in **Figure 4**.

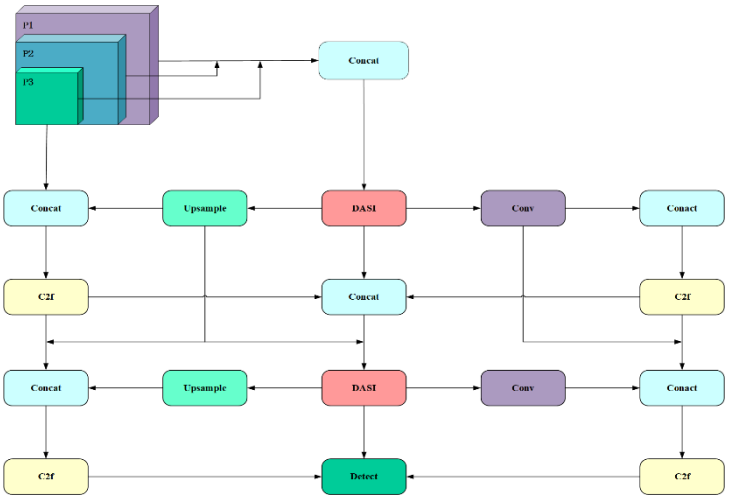


Figure 4. MSFAN - Multiscale Feature Aggregation Propagation Network architecture diagram

MSFAN achieves spatially adaptive feature fusion. The network calculates the optimal fusion weights for each spatial location within an image. This fine-grained adaptability significantly enhances the network's processing capability for complex scenes. By deploying DASI modules at key nodes of the feature pyramid, as shown in **Figure 5**, MSFAN substantially improves performance in multi-scale object detection. Particularly for small object detection, the adaptive fusion mechanism better preserves detailed information, thereby increasing detection accuracy. For large object detection, the effective utilization of semantic information enhances classification accuracy. The DASI module is the core component of MSFAN. In object detection tasks, different regions of an image exhibit significant differences in deep semantic features and shallow detail features. Based on this understanding, the DASI module designs an adaptive weight generation mechanism that calculates the most suitable feature fusion weight for each spatial location.

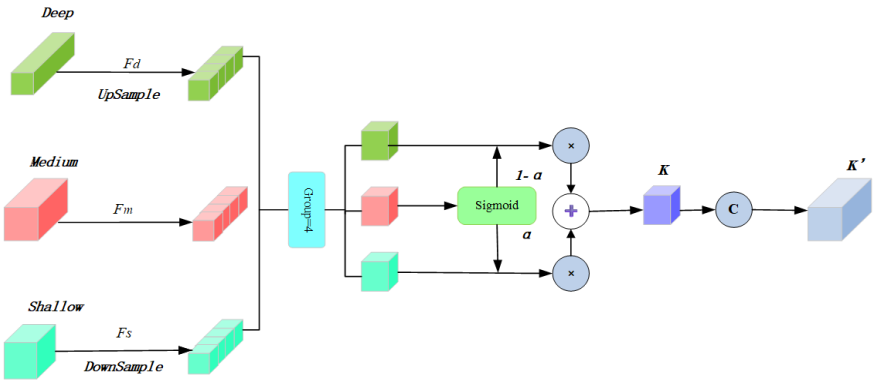


Figure 5. DASI module structure diagram

(1) Feature input and preprocessing

The DASI module receives feature inputs across three levels. To ensure effective fusion, the module first standardizes the input features as shown in Equations (1) and (2):

$$F_d^{norm} = Align(F_d, H \times W, C) \tag{1}$$

$$F_s^{norm} = Align(F_s, H \times W, C) \tag{2}$$

In the formula, represents deep features; H, W, and C denote the height, width, and number of channels of the feature map, respectively; F_s represents shallow features; Align represents aligned deep features, where the Align operation unifies all features to the same spatial and channel dimensions via bilinear interpolation and 1×1 convolution.

(2) Adaptive weight generation

Deep and shallow feature spatial dimensions are aligned with intermediate layer features through upsampling and downsampling operations, followed by 1×1 convolutions to unify channel counts. Subsequently, the three aligned features are subdivided into four equal parts along the channel dimension, with each subblock containing channels.

For each group *i*, the DASI module generates adaptive weights α based on the intermediate layer feature sub-blocks. These weights guide the fusion process between deep and shallow feature sub-blocks, enabling pixel-level adaptive feature integration. The specific formula is expressed as:

$$\alpha = Sigmoid(m_i) \tag{3}$$

The intermediate layer feature sub-block generates weights ∈ [0,1] through the Sigmoid activation function. These weights serve as adaptive fusion coefficients between shallow and deep features, as shown in Equation (4):

$$F_m^1 = \alpha \times F_s^{norm} + (1 - \alpha) \times F_d^{norm} \tag{4}$$

An adaptive weighting combination representing deep and shallow features: when α approaches 1, it primarily preserves the detailed information of shallow features; when α approaches 0, it primarily preserves the detailed information of deep features; when α is set to 0.5, both types of features achieve relatively balanced fusion.

(3) Feature optimization and output

To further enhance the expressive capability of the fused features, all adaptive fusion results are merged along the channel dimension, as shown in Equation (5):

$$K = Conact[F_m^1, F_m^2, F_m^3, F_m^4] \tag{5}$$

The result obtained by concatenating the selected aggregated outputs from the four partitions. Finally, the merged features undergo convolution, batch normalization, and ReLU activation in sequence, as shown in Equation (6):

$$K^{'} = ReLU(BN(Conv(K))) \tag{6}$$

Represents the final output of the fused features.

2.2.3. Improved loss function

Wise-IoU employs a dynamic focusing mechanism to adaptively mitigate the impact of low-quality sample data on the accuracy of persimmon quality grading detection. Furthermore, the relative area weighting strategy introduced during Wise-IoU computation assigns more appropriate loss weights to small-scale targets. This alleviates the reduction in extractable features caused by leaf and branch occlusion. Therefore, Wise-IoU is selected as the bounding box regression loss function. The calculation formulas for the Wise-IoU loss function (7), (8), and (9) are as follows:

$$L_{IoU} = 1 - IoU \tag{7}$$

$$R_{WIoU} = \exp\left(\frac{\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2}{\left(W_i^2 + H_i^2\right)^*}\right) \tag{8}$$

$$L_{WIoU} = rR_{WIoU}L_{IoU} \tag{9}$$

In the formula, IoU denotes the intersection-over-union ratio between the target box and anchor box, RWIoU represents the distance loss, and the superscript * denotes the gradient separation operation, which isolates the width W_i and height H_i of the ground truth box from the backpropagation computation graph. When RWIoU falls within the interval [1, e], it indicates a medium-quality anchor box, leading to an increased LIOU loss value. Conversely, when LIOU resides within the interval [0, 1], it corresponds to a high-quality anchor box, prompting

the system to significantly reduce its RWIoU weight.

2.3. Experimental environment and parameter configuration

The training and testing processes of this research model were implemented on the Windows 11 operating system. The relevant configuration comprised a 16 vCPU Intel® Xeon® Platinum 8481C processor equipped with an NVIDIA GeForce RTX 4090 GPU. The development environment consisted of PyTorch (2024.3.1.1) + Python 3.8 + CUDA 11.8. During training, the following parameters were set: 100 training epochs, a batch size of 32, input image dimensions of 640×640 pixels, SGD as the optimizer, and an initial learning rate of 0.01.

2.4. Evaluation indicators

In object detection tasks, different evaluation metrics reflect the performance of detection algorithms from various perspectives. To comprehensively assess the detection capabilities of the model developed in this study, a comprehensive evaluation is conducted using metrics such as precision (P), recall (R), mean average precision (mAP), and others.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{12}$$

In terms of detection speed, computational load is evaluated using the number of floating-point operations (GFLOPs), while model complexity is measured by the number of parameters (params).

3. Results and analysis

3.1. Ablation experiment

To quantitatively evaluate the specific contributions of each proposed improvement method to the performance of the YOLOv8s model, a systematic set of ablation experiments was designed. Detailed results of the ablation experiments are shown in **Table 1**. The experiments used the original YOLOv8s as the baseline model and progressively integrated the proposed improvement modules into the baseline model. All experiments were conducted under identical hardware environments, datasets, training parameters, and evaluation metrics. By comparing key performance indicators—including detection accuracy, recall, and mAP—under both individual and combined effects of each improvement module, we thoroughly analyzed the specific impact of each enhancement on persimmon grading detection performance, thereby validating the effectiveness of each module.

Table 1. Ablation experiments

ADown	MSFAN	WIoU	*P*/%	*R*/%	mAP/%	Para/M	GFLOPs
—	—	—	78.7	82.1	86.7	11.1	28.4
√	—	—	79.5	83.3	86.9	9.4	25.7
—	√	—	73.9	83.4	87.3	10.9	29.9
—	—	√	79.2	80.8	86.9	11.1	28.4
√	√	—	80.3	83.0	88.2	8.6	27.4
√	—	√	79.0	83.5	88.8	9.4	25.7
—	√	√	75.8	78.7	86.4	10.2	29.9
√	√	√	82.8	83.7	89.4	8.6	27.4

The fully improved model (ADown+MSFAN+Wise-IoU) achieved the best performance across all evaluation metrics. Accuracy reached 82.8%, an improvement of 4.1 percentage points over the baseline model; recall was 83.7%, an increase of 1.6 percentage points; mAP@0.5 reached 89.4%, an increase of 2.7 percentage points. Simultaneously, the model's parameter count was reduced to 8.6 million, a decrease of 22.5%, with GFLOPs at 27.4, showing a reduction compared to the baseline model.

3.2. Comparison results of mainstream models

To validate the superiority of the YOLOv8-MSFAN algorithm, six widely applied and representative network models from the YOLO series—YOLOv5, YOLOv8, YOLOv9, YOLOv10, YOLOv11, and YOLOv12—were selected for comparative testing against the proposed YOLOv8-AFDAN model. Experimental results are presented in **Table 2**. To ensure experimental accuracy, all network models were trained and tested using the same dataset and identical training parameters, with consistent evaluation metrics applied as the assessment standard.

Table 2. Comparative experimental data for different YOLO models

Models	*P*/%	*R*/%	mAP/%	P/MB	GFLOPs
YOLOv5	76.7	80.9	85.4	9.1	23.8
YOLOv8	78.7	82.1	86.7	11.1	28.4
YOLOv8-MSFAN	82.8	83.7	89.4	8.6	27.4
YOLOv9	74.1	76.8	80.9	7.2	26.7
YOLOv10	67.7	78.0	77.9	7.2	21.4
YOLO11	74.8	77.9	82.9	9.4	21.3
YOLO12	69.6	81.0	79.5	9.2	21.2

Comparative results across different models reveal that the YOLOv8-MSFAN model outperforms all others in every aspect, surpassing even the latest YOLO11 and YOLO12 versions. This fully demonstrates the effectiveness of the improved model proposed in this paper for fruit persimmon quality grading detection tasks.

3.3. Detection performance under different occlusion conditions

To validate the detection performance of the improved YOLOv8s model on persimmons under varying occlusion

conditions, the enhanced model was tested against the standard YOLOv8s model in identical scenarios. As shown in **Figure 6**, under mild occlusion, the YOLOv8s model produced false positives, misclassifying a single occluded fruit as two separate objects. Under severe occlusion, the YOLOv8s model incorrectly classified the "normal" category as "sunken." Thus, under occlusion conditions, the YOLOv8-MSFAN model demonstrates superior detection performance.

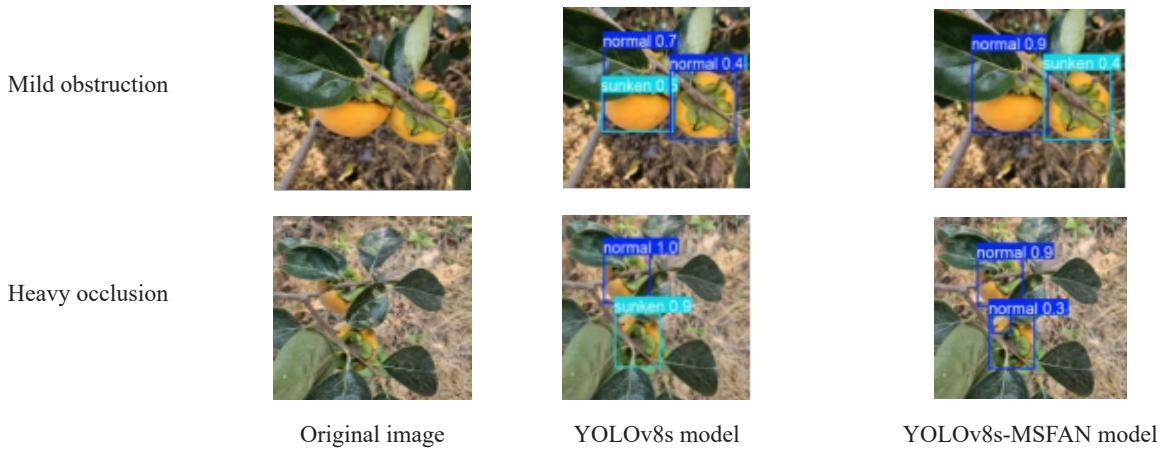


Figure 6. Performance of YOLOv8s and YOLOv8-MSFAN models under different occlusion conditions

4. Conclusion

Due to factors such as severe foliage obstruction and subtle differences in characteristics between fruits of varying quality, accurate detection in persimmon quality grading remains challenging. Furthermore, research on persimmon quality assessment is scarce. this study proposes a lightweight persimmon detection model based on an improved YOLOv8s. It replaces the Conv layer in the backbone network with an ADown module, incorporates a Multi-Scale Feature Aggregation Propagation Network (MSFAN) into the neck layer, and finally introduces the Wise-IoU loss function. Across various occlusion scenarios, the YOLOv8-MSFAN model demonstrated robust performance, accurately identifying persimmons under both mild and severe occlusion conditions. This effectively addresses the accuracy degradation issues faced by traditional methods in complex environments. The model provides a valuable technical solution for fruit quality detection in smart agriculture, holding significant theoretical and practical value for advancing agricultural modernization and intelligent development.

Funding

National Natural Science Foundation of China (61703363, 62272284); Shanxi Provincial Basic Research Program (201801D121148); Yuncheng University Research and Innovation Team for Data Mining and Industrial Intelligence Applications (YCXYTD-202402)

Disclosure statement

The authors declare no conflict of interest.

References

[1] Gao R, Huang Y, Zhao W, et al., 2023, Current Status and SWOT Analysis of Luotian Sweet Persimmon Industry Development in Luotian County. Southern China Fruit, 52(03): 212–218.

[2] Guo X, Chang Y, Huang J, et al., 2022, Rural Revitalization: A Promising Path for Persimmon Industry!—Development Strategies for Guangxi's Persimmon Industry in the Context of Rural Revitalization. Southern Horticulture, 33(3): 78–84.

[3] Huo P, 2023, Development Countermeasures for Guangxi Zhuang Autonomous Region's Fruit Industry Based on SWOT Analysis. Rural Science and Technology, 14 (11): 42–45.

[4] Li F, 2022, An Exploration of Intelligent Technology Applications in Modern Agricultural Machinery. Rural Practical Technology, (3): 110–111.

[5] Gao M, 2022, Research on the Application of New Agricultural Machinery Technologies in Rural Production. Farmer's Wealth Companion, (23): 87–89.

[6] Pan M, 2021, Application of Machine Vision Technology in Field Agricultural Machinery. Automation Technology and Application, 40(08): 163–166.

[7] He D, Yang Q, Xue S, et al., 1998, Research on Computer Vision Grading Technology for Fruit Surface Color. Transactions of the Chinese Society for Agricultural Engineering, (03): 207–210.

[8] Tao Y, Heinemann PH, Varghese Z, et al., 2013, Machine Vision for Color Inspection of Potatoes and Apples. Transactions of the ASAE, 38(5): 1555–1561.

[9] Sofu MM, Er O, Kayacan MC, et al., 2016, Design of an Automatic Apple Sorting System Using Machine Vision. Computers and Electronics in Agriculture, 2016: 395–405.

[10] Huang C, Fei J, 2017, An Online Grading Method for Apples Based on Image Feature Fusion. Transactions of the Chinese Society for Agricultural Engineering, 33(01): 285–291.

[11] Huang F, 2016, Research on Quality Detection and Variety Identification of Nectarines Based on Hyperspectral Imaging Technology, dissertation, Shanxi Agricultural University.

[12] Ma G, 2022, Research on Pear Quality Detection and Grading System Based on Machine Vision, dissertation, Hebei Normal University of Science and Technology.

[13] Huang Y, Li N, Liu Z, et al., 2023, Classification Method for Grape Disease-Affected Leaves Using an Improved Lightweight Convolutional Network. Journal of Harbin University of Science and Technology, 28(03): 1–9.

[14] Liu Z, Ding Y, 2020, Research on Recognition Methods for Occluded Fruits in Natural Environments. Research on Computer Applications, 37(S2): 333–335, 339.

[15] Zhang Z, Zhou J, Jiang Z, et al., 2024, Apple Recognition Method in Natural Orchard Environments Based on an Improved YOLOv7 Lightweight Model. Transactions of the Chinese Society for Agricultural Machinery, 55(3): 231–242.

[16] Qiu P, Su Z, Jia Y, 2023, Research on Surface Damage Detection of Fragrant Pears Using Machine Vision. Information Systems Engineering, (12): 133–136.

[17] Sun X, 2023, Research on Apple Appearance Quality Grading Using Residual Neural Networks, dissertation, Shandong Technology and Business University.

[18] Wang CY, Yeh IH, Mark Liao HY, 2024, YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, European Conference on Computer Vision, Springer Nature Switzerland, Cham, 1–21.