

# Research on the Performance Improvement of YOLO Algorithm Based on C3 Module Optimization in Agricultural Harvesting

<https://doi.org/10.63174/xdi.BWSN5805>

Received: 26 Apr 2025

Accepted: 11 May 2025

Published: 16 May 2025

Open Access



Liyang Mu<sup>1,Δ</sup>, Chenfeng Wang<sup>1,Δ</sup>, Hao Wang<sup>1</sup>, Kecheng Shan<sup>1</sup>, Yongqi Lian<sup>1</sup>, Xin Liu<sup>1, 2,\*</sup>

**Abstract:** With the development of computer vision and deep learning, agricultural automation has been significantly advanced. The YOLO series algorithms are widely applied in agricultural fields such as robotic fruit picking, yet they still confront challenges like occlusion and light variations. This study is based on YOLOv5 6.1, creating the lightweight C3-L module by processing the C3 module of the 5s model. Experiments replaced the C3 module with C3-L at Backbone, Head, and Backbone+Head, and introduced CBAM and CA attention mechanisms to compare model performances. Results show that the C3-L module reduces resource and GPU memory usage during training. Replacing C3 in Head is stable, and adding CBAM increases overall accuracy by 5%. When high accuracy isn't required, partial replacement in Backbone cuts memory usage by 17.4%, facilitating mobile hardware operation. This research provides references for optimizing YOLO algorithms in agricultural picking and porting them to devices like microcontrollers.

## 1. Introduction

Recent advancements in computer vision and deep learning have significantly enhanced the efficiency of agricultural harvesting robots. Sun et al. <sup>[1]</sup> proposed YOLO-P, an efficient pear detection method tailored for complex orchard environments, demonstrating high accuracy and speed. Similarly, Zhaoxin et al. <sup>[2]</sup> designed a tomato-picking robot system based on YOLOv5, optimizing real-time detection for robotic applications. Yang et al. <sup>[3]</sup> developed a pumpkin pick-and-place robot utilizing RGB-D cameras and YOLO-based detection, improving automation in unstructured environments. A comprehensive review by Badgujar et al. <sup>[4,8]</sup> highlighted the widespread adoption of YOLO algorithms in agricultural object detection. Ji et al. <sup>[5]</sup> introduced ShufflenetV2-YOLOX for real-time apple detection, balancing speed and precision. Yu et al. <sup>[6]</sup> enhanced strawberry harvesting by localizing picking points in ridge-planting systems using visual localization. Gai et al. <sup>[7]</sup> improved YOLOv4 for cherry detection, achieving robust performance in varying conditions. Kumar et al. <sup>[9]</sup> implemented a vision-guided pick-and-place system using Raspberry Pi and YOLO, emphasizing cost-effectiveness. Huang et al. <sup>[10]</sup> introduced Pepper-YOLO for green pepper detection and picking point localization in cluttered scenes. Ning et al. <sup>[11]</sup> optimized sweet pepper recognition and picking sequence planning in high-density orchards. Nan et al. <sup>[12]</sup> proposed WGB-YOLO for multi-class pitaya detection, enhancing accuracy in target rows. Zhang et al. <sup>[13]</sup> employed YOLOMS for mango recognition and stem picking point localization. Zhang et al. <sup>[14]</sup> applied improved YOLOv4 to detect impurities in machine-picked cotton. Ma et al. <sup>[15]</sup> and Du et al. <sup>[16]</sup> developed STRAW-YOLO and DSW-YOLO, respectively, for strawberry

detection under occlusion. Qin et al. <sup>[17]</sup> presented Ag-YOLO for precise palm spraying, while Zhang et al. <sup>[18]</sup> introduced AgriPest-YOLO for rapid pest detection. Liang et al. <sup>[19]</sup> improved YOLOv8 for citrus picking-point localization, and Xu et al. <sup>[20]</sup> proposed a real-time algorithm for trellis grape harvesting. These studies collectively advance robotic harvesting through optimized YOLO-based detection and picking strategies.

## 2. Data set

The training dataset in this paper uses online images. The number of online images that meet the requirements is relatively small. Therefore, the rotation method is used to increase the number of datasets (**Figure 1**). Increasing the number of datasets through rotation can enhance the diversity of the datasets. During the training process, it can improve the recognition of apple-picking environments from different angles and is more in line with the multi-angle environment of the robotic arm picking process.

During the actual picking process, the robotic arm needs to make logical judgments on the picking priority due to the problem of apple occlusion. Therefore, the labels of this dataset are divided into three categories as shown in the **Figure 2**: no occlusion, branch occlusion, and apple occlusion. The priority order of the three is no occlusion, apple occlusion, and branch occlusion respectively. When encountering tree branches blocking, the processing logic of the robotic arm is more inclined to change the picking Angle.

## 3. Experiment set

<sup>1</sup> School of Mechanical and Electronic Engineering, Shandong Agriculture and Engineering University, Jinan 250100, Shandong Province, PR China

<sup>2</sup> School of physics and electronics, Shandong Normal University, Jinan 250358, Shandong Province, PR China

<sup>Δ</sup> Made the same contribution

\* Corresponding Author: liux951127@163.com



**Figure 1.** Data rotation result

### 3.1. Method

This experiment is lightweight based on the YOLOv5 6.1 version. This version further expands the support scope of the inference engine and deeply integrates advanced inference frameworks such as TensorRT, TensorFlow EdgeTPU, and OpenVINO. Achieve efficient adaptation to heterogeneous hardware platforms such as NVIDIA Gpus, mobile devices, Intel cpus and integrated Gpus. The experiment used pycharm and anaconda to build the experimental platform. The hardware used was a HP Z series notebook. The main hardware parameters are shown in **Table 1**, and the software parameters and training parameters are shown in **Table 2**.

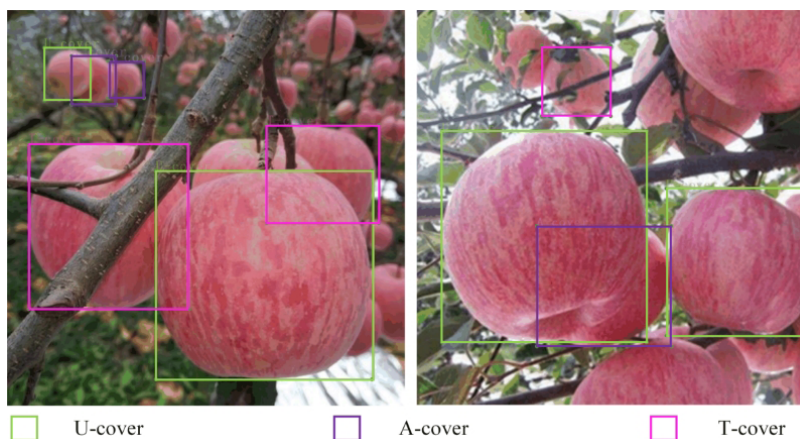
YOLOv5 has different variants (as shown in the **Figure 3**, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x and YOLOv5n) representing models of different sizes and complexities. These variants offer different trade-offs between speed and accuracy to accommodate varying computing power and real-time requirements. The optimization of this algorithm is applied to the implementation detection in the picking scenario. After the training is completed, it will be installed in lightweight devices such as single-chip microcomputers. YOLOv5s: This is the smallest model in the YOLOv5 series. “s” stands for “small”. This model performs best on devices with limited computing resources,

such as mobile devices or edge devices, and has the fastest detection speed. Therefore, this optimization will be carried out based on the 5s model.

### 3.2. Proposed method

In YOLOv5 6.1, the convolution layer adopts an improved convolution module, which optimizes the structure of the standard convolution and introduces more efficient convolution operations. As shown in the **Figure 4** is its schematic diagram. In the C3 module, lightweight processing is carried out. Batch normalization and activation functions are used to replace the convolution module and addition operation in the Bottleneck to reduce the number of algorithm operation layers.

In the YOLO algorithm, there are many calling positions for the C3 module. Replacing it at different positions using the improved C3 module (C3-L) will produce different results. In this experiment, C3 was replaced with C3-L at the positions of Backbone and Head and Backbone and Head respectively. Meanwhile, in order to verify the stability of C3-L, the attention mechanism is added in YOLO while the C3 module is replaced to verify the performance of C3-L under different algorithm complexities. The group parameters of this experiment are shown in the **Table 3**.



**Figure 2.** Annotated examples<sup>[21]</sup>

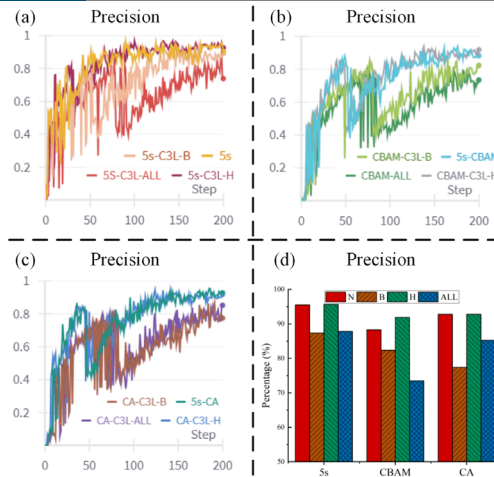
**Table 1.** Hardware parameters

Parameter	Value
System	Windows 11
CPU	Intel i5 12500H
GPU	NVIDIA GeForce RTX 3050 Ti Laptop GPU
RAM	16G
SSD	512G+1T

**Table 2.** Software version

Parameter	Value
Python version	3.8.1.8
PyTorch version	2.1.0
Wandb version	0.16.3
CUDA version of NVIDIA	12.0
CUDA version of PyTorch	11.7
Epochs	200
Input image size	640
Batch size	4





**Figure 5.** The detection precision of different models

variants extract and process features, thereby affecting the model’s accurate recognition ability of the target. It is found that the optimization in the Head part can maintain good stability.

**Figure 5 (b-c)** focuses on the model variants that introduce the CBAM (Convolutional Block Attention Module) and the CA module. Although the Precision of all variants shows a fluctuating upward trend with the increase of the number of training steps, However, there are obvious differences among different variants in terms of growth rate and the final achieved Precision values. It was found that after adding the attention mechanism, the stability of the optimization of the Head part was more obvious, and the accuracy was improved under the CBAM attention mechanism.

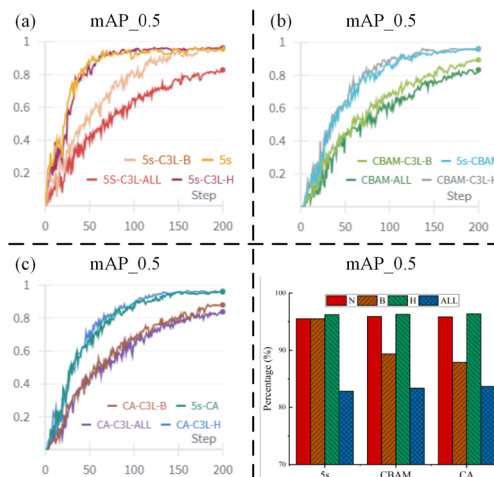
**Figure 5 (d)** shows the final accuracy under different conditions. It is found that the optimization at different positions has different improvements under different attention mechanisms. However, the optimization of the Head part

is stable in all cases, with an accuracy improvement of 5% under the CBAM attention mechanism and maintaining the original performance in other cases. However, the optimization of the Backbone and All parts reduces the accuracy. In the task of object detection, the mean average accuracy (mAP) when the intersection and union ratio (IoU) threshold is 0.5 (mAP\_0.5) is one of the key indicators for evaluating the performance of the model, which reflects the model’s ability to recognize and locate the object under relatively loose detection standards.

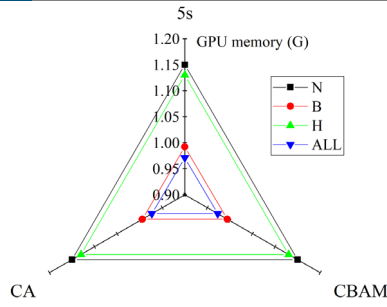
**Figure 6 (a)** shows various model variants based on 5s, including 5S-C3L-B, 5s, 5S-C3L-ALL and 5S-C3L-H. In the early stage of training, the growth trend of mAP\_0.5 of each model variant showed obvious differences. Among them, although the growth of 5s-C3L-B is slow, it can still reach a relatively high percentage with the increase of training rounds. However, as the number of training steps continues to increase, each curve gradually separates. This

**Table 4.** Model summary of different model

No.	Layers	GFLOPs	Paparameters
1	270	16	7027720
2	215	13.2	5970376
3	234	13.9	6001416
4	179	11.1	4944072
5	281	16	7060586
6	226	13.3	6003242
7	245	13.9	6034282
8	190	11.1	4976938
9	280	16	7053368
10	225	13.2	5996024
11	244	13.9	6027064
12	189	11.1	4969720



**Figure 6.** The detection mAP\_0.5 of different models



**Figure 7.** GPU information of different models

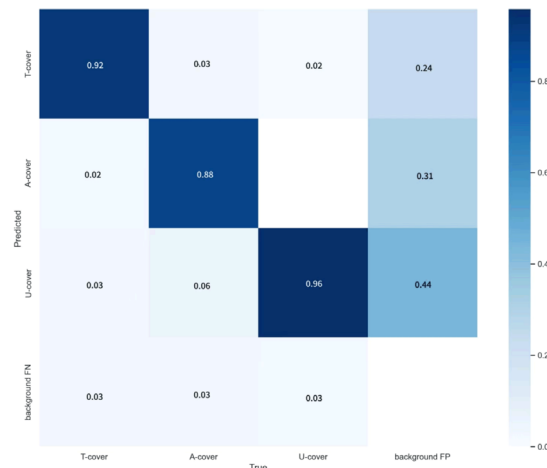
indicates that different structural adjustment methods, even based on the same basic model 5s, perform differently at different stages of the training process. The mechanisms of action of different variants on feature extraction, fusion and the overall optimization of the model are different, which in turn affects the growth rhythm and final value of mAP\_0.5. It was found that all except 5S-C3L-ALL could achieve better performance.

In the YOLO model, the introduction of CBAM (and CA, two different variants of the two attention mechanisms, presents their respective characteristics in the variation of the mAP\_0.5 index with the number of training steps. The model variants introducing CBAM and CA, such as CBAM-C3L-H, CA-C3L-H, etc., although mAP\_0.5 has steadily increased, there are significant differences from other variants in terms of growth rate and final values, indicating that different embedding positions and methods have different effects on

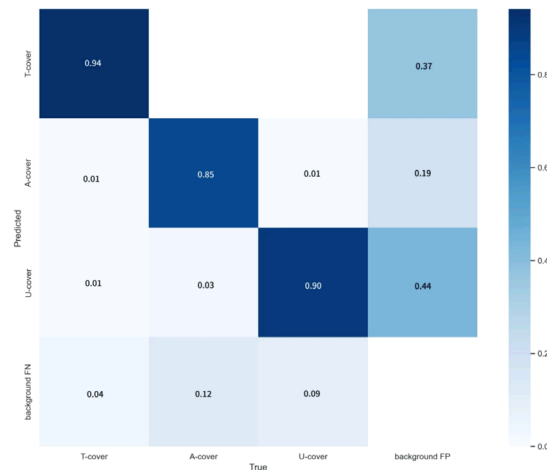
the improvement of model performance. The degree of effect varies in focusing on key features and suppressing background interference. It was found that the changes in the Head section of C3L could maintain or improve minor performance. It can be seen from **Figure 6 (d)** that both the common model and the model with C3L replaced in the Head part can maintain high performance and have good stability in various situations.

### 4.3. GPU information

From an overall perspective in **Figure 7**, the memory usage of all unoptimized models is relatively high, and the fluctuation range is approximately around 1.15G. Among them, the peak memory usage of component N reached 1.15G, while the memory usage of component ALL was relatively low, at 0.95G. The



(a) 5s-CBAM-H



(b) 5s

**Figure 8.** Confusion Matrix of different model

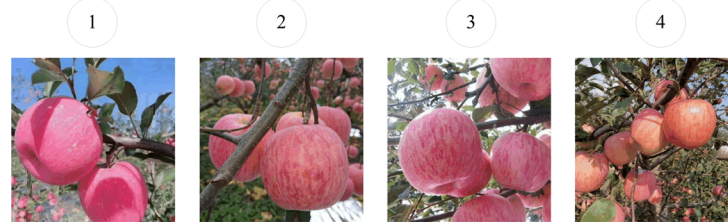


Figure 9. Original picture

use of the C3L model can reduce the invocation of GPU memory and alleviate the burden at runtime, which is crucial for hardware such as single-chip micro-computers.

#### 4.4. Result of detection

As can be seen from Figure 8, after introducing the attention mechanism CBAM, the detection accuracy of the model for each category has been optimized. Although the improvement is slight, it is sufficient to reflect the positive promoting effect of this mechanism on the performance of the model. Compared with the original 5s model, the improved model shows remarkable optimization effects when dealing with categories that had a higher probability of missed detection or false detection. The probability of incorrect detection has steadily decreased. For the remaining categories, although there is a certain degree of fluctuation in the probabilities of missed detection and false detection, this fluctuation range is relatively small and does not have a substantial impact on the overall detection accuracy, fully demonstrating the stability and reliability of the improved model under different scenarios and categories.

The test set selected fruit pictures of different complexity levels for detection in Figure 9. It was found in Figure 10 that adding the attention mechanism in simple scenarios would instead reduce the detection accuracy, but as the complexity increased, adding the attention mechanism would have a slight improvement on the accuracy. After replacing the C3 module with the optimized C3L, the accuracy will also decrease in the single-room scenario. However,

as the complexity of the Apple environment increases, the optimized C3L can effectively improve the originally very low-accuracy Apple, but there will be a slight reduction for the high-accuracy Apple. In the most complex scenarios, the accuracy of different apples increases and decreases, while the overall accuracy remains unchanged.

It is not difficult to find from Figure 11 that, compared with the models that have not replaced the C3 module with the C3L module, most of the improved training models show a significant improvement in the training speed. This phenomenon strongly indicates that when the C3 module in the model is replaced by the C3L module, the performance of most training models has been optimized, highlighting the positive role of the C3L module in improving the training efficiency and performance of the model.

#### 5. Conclusion

The improved C3L module can effectively reduce the invocation of computer resources during the training process. It was found that replacing the C3 module at different positions could reduce the usage of graphics card memory, but the accuracy rate varied in different situations.

1. It was found that replacing the C3 module in the Head section has excellent stability. After adding different attention mechanisms, it can achieve high positional accuracy. Under the CBAM attention mechanism, the overall accuracy rate has increased by 5%.
2. When the accuracy rate requirement is not high, the C3 module can be re-

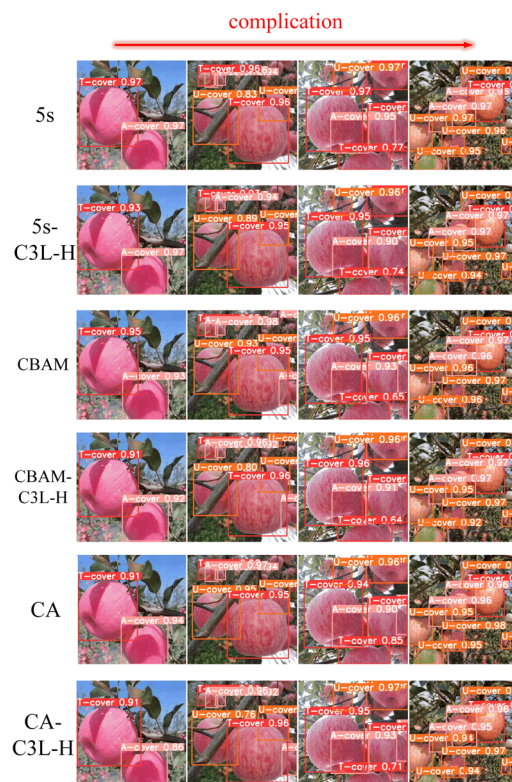


Figure 10. Visual inspection map

	The first one	The second one	The third one	The fourth one	Sum
5s	0.009	0.009	0.014	0.014	0.046
CA	0.02	0.009	0.009	0.009	0.047
CBAM	0.012	0.009	0.01	0.011	0.042
5s-B	0.012	0.008	0.008	0.008	0.036
CA-B	0.012	0.008	0.014	0.008	0.042
CBAM-B	0.008	0.012	0.008	0.008	0.036
5s-H	0.008	0.008	0.008	0.008	0.032
CA-H	0.019	0.008	0.015	0.008	0.05
CBAMH	0.008	0.008	0.009	0.008	0.033
5s-ALL	0.007	0.007	0.007	0.007	0.028
CA-ALL	0.01	0.008	0.007	0.007	0.032
CBAM-ALL	0.007	0.01	0.012	0.011	0.04

**Figure 11.** The processing time of the same dataset for different models

placed in the Backbone part, which can effectively reduce the call to hardware resources, and the video memory can be reduced by 17.4%, which can improve the operation status in mobile hardware.

#### Future work

Set up the operating environment on the mobile device to test the detection effect and running speed of the modified code.

#### Acknowledgements

This work was supported by the Funding for Visiting Scholar project of ordinary undergraduate universities in Shandong Province in 2024, Youth Fund of Shandong Agriculture and Engineering University (QNKJZ202301), and Shandong Agriculture and Engineering University Start-Up Fund for Talented Scholars (BSQJ-202301). There is no conflict of interest in this paper. All the achievements of others have been marked.

#### References

- [1] H. Sun, B. Wang, J. Xue. "YOLO-P: An efficient method for pear fast detection in complex orchard picking environment." *Front. Plant Sci.* **2023**, *13*, 1089454.
- [2] G. Zhaoxin, L. Han, Z. Zhijiang, P. Libo. "Design a Robot System for Tomato Picking Based on YOLO v5." *IFAC-PapersOnLine* **2022**, *55*, 3, 166–71.
- [3] L. Yang, T. Noguchi, Y. Hoshino. "Development of a pumpkin fruits pick-and-place robot using an RGB-D camera and a YOLO based object detection AI model." *Computers and Electronics in Agriculture* **2024**, *227*, 109625.
- [4] C. M. Badgujar, A. Poullose, H. Gan. "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review." *Computers and Electronics in Agriculture* **2024**, *223*, 109090.
- [5] W. Ji, Y. Pan, B. Xu, J. Wang. "A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX." *Agriculture* **2022**, *12*, 6, 856.
- [6] Y. Yu, K. Zhang, H. Liu, L. Yang, D. Zhang. "Real-Time Visual Localization of the Picking Points for a Ridge-Planting Strawberry Harvesting Robot." *IEEE Access* **2020**, *8*, 116556–68.
- [7] R. Gai, N. Chen, H. Yuan. "A detection algorithm for cherry fruits based on the improved YOLO-v4 model." *Neural Comput & Applic* **2023**, *35*, 19, 13895–906.
- [8] C. M. Badgujar, A. Poullose, H. Gan. "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review." *Computers and Electronics in Agriculture* **2024**, *223*, 109090.
- [9] G. H. Kumar, D. R. Kumar Raja, S. Suresh, R. Kottamala, M. Harsith. "Vision-Guided Pick and Place Systems Using Raspberry Pi and YOLO." **2024**, 1–7.
- [10] Y. Huang, Y. Zhong, D. Zhong, C. Yang, L. Wei, Z. Zou, R. Chen. "Pepper-YOLO: an lightweight model for green pepper detection and picking point localization in complex environments." *Front. Plant Sci.* **2024**, *15*, 1508258.
- [11] Z. Ning, L. Luo, X. Ding, Z. Dong, B. Yang, J. Cai, W. Chen, Q. Lu. "Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards." *Computers and Electronics in Agriculture* **2022**, *196*, 106878.
- [12] Y. Nan, H. Zhang, Y. Zeng, J. Zheng, Y. Ge. "Intelligent detection

of Multi-Class pitaya fruits in target picking row based on WGB-YOLO network." *Computers and Electronics in Agriculture* **2023**, *208*, 107780.

[13] B. Zhang, Y. Xia, R. Wang, Y. Wang, C. Yin, M. Fu, W. Fu. "Recognition of mango and location of picking point on stem based on a multi-task CNN model named YOLOMS." *Precision Agric* **2024**, *25*, 3, 1454–76.

[14] C. Zhang, T. Li, W. Zhang. "The Detection of Impurity Content in Machine-Picked Seed Cotton Based on Image Processing and Improved YOLO V4." *Agronomy* **2021**, *12*, 1, 66.

[15] Z. Ma, N. Dong, J. Gu, H. Cheng, Z. Meng, X. Du. "STRAW-YOLO: A detection method for strawberry fruits targets and key points." *Computers and Electronics in Agriculture* **2025**, *230*, 109853.

[16] X. Du, H. Cheng, Z. Ma, W. Lu, M. Wang, Z. Meng, C. Jiang, F. Hong. "DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels." *Computers and Electronics in Agriculture* **2023**, *214*, 108304.

[17] Z. Qin, W. Wang, K.-H. Dammer, L. Guo, Z. Cao. "Ag-YOLO: A Real-Time Low-Cost Detector for Precise Spraying With Case Study of Palms." *Front. Plant Sci.* **2021**, *12*, 753603.

[18] W. Zhang, H. Huang, Y. Sun, X. Wu. "AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning." *Front. Plant Sci.* **2022**, *13*, 1079384.

[19] Y. Liang, W. Jiang, Y. Liu, Z. Wu, R. Zheng. "Picking-Point Localization Algorithm for Citrus Fruits Based on Improved YOLOv8 Model." *Agriculture* **2025**, *15*, 3, 237.

[20] Z. Xu, J. Liu, J. Wang, L. Cai, Y. Jin, S. Zhao, B. Xie. "Realtime Picking Point Decision Algorithm of Trellis Grape for High-Speed Robotic Cut-and-Catch Harvesting." *Agronomy* **2023**, *13*, 6, 1618.

[21] K. Shan, Q. Feng, X. Li, X. Meng, H. Lyu, C. Wang, L. Mu, X. Liu. "C3-Light Lightweight Algorithm Optimization under YOLOv5 Framework for Apple-Picking Recognition." *XDI* **2025**, *1*, 1.