Engenharia Agrícola

ISSN 1809-4430
www.engenhariaagricola.org.br

http://dx.doi.org/10.1590/1809-4430-Eng.Agric.v46e20250139/2026

Scientific Paper

Orchard trunk recognition model based on improved YOLOv9

Jianzheng Liu[1], Jiangyi Han[1*], Minghao Huang[1]

[1] School of Automotive and Traffic Engineering, Jiangsu University/Zhenjiang, China.

* Corresponding author: hjy0306@ujs.edu.cn

How to cite: Liu, J., Han, J., & Huang, M. (2026). Orchard trunk recognition model based on improved YOLOv9. *Engenharia Agrícola, 46,* e20250139. http://dx.doi.org/10.1590/1809-4430-Eng.Agric.v46e20250139/2026

Abstract

The ability to recognise fruit tree trunks is essential to allow orchard operation robots to perform various intelligent tasks. However, dynamic changes in natural lighting, mutual occlusion of branches and leaves, and various weather-related factors pose severe challenges for target detection algorithms. To address these problems, this paper proposes a recognition method based on YOLOv9c for target objects such as fruit tree trunks, people, and obstacles in complex orchard environments. Orchard images were collected on site and augmented using weather-related factors such as fog, rain, and strong light to enrich the dataset. The YOLOv9c model was improved by integrating a spatial-channel synergistic attention (SCSA) mechanism into the backbone network to enhance its anti-interference capability in complex environments. The Slim-Neck-by-GSConv network structure was introduced to modify the neck network, and to reduce the number of model parameters. The improved YOLOv9c target recognition model was trained and validated on the experimental dataset, and the results showed that it achieved a precision of 97%, a recall rate of 95.4%, and an mAP@0.5 of 97.3%, thus demonstrating that the model exhibits superior detection precision and speed in complex orchard environments.

Keywords: fruit tree trunk, deep learning, SCSA, GSConv, loss function.

Introduction

China's fruit industry boasts a wide variety and large total output, and is consistently ranked first globally in terms of both its total orchard area and fruit production (Ghazal et al., 2024). With the continuous increases in China's fruit yield and cultivation area, issues such as labour shortages, high labour intensity, and low work efficiency in orchards have become increasingly prominent. In recent years, smart agriculture has emerged as a major future direction for agricultural development. In the context of vigorous development of smart agricultural technologies, intelligent orchard management is essential in regard to improving the yield and quality of fruit while reducing labour costs. One of the core tasks in intelligent orchard management is the accurate and rapid identification of targets such as fruit trees and fruits, meaning that in an object detection model, a balance must be struck between lightweight design, high precision, and strong robustness in order to meet the practical demands of complex orchard environments.

As fruit trees are the primary target objects in orchards, contour recognition and positioning are key

Area Editor: Gizele Ingrid Gadotti
Received in: 9-15-2025
Accepted in: 1-27-2026

Edited by
sbea

technologies for environmental perception and dynamic path planning for orchard operation robots and autonomous orchard tractors (Zhang et al., 2024). Due to the continuous developments in the area of deep learning, target detection algorithms have undergone significant advancements (Vijayakumar & Vairavasundaram, 2024; Wang et al., 2023). The development and application of object detection algorithms means that the identification and localisation of targets using these algorithms in agricultural production processes has become a key area of research (Crespo et al., 2025; Wang et al., 2025).

The YOLO series of algorithms not only have fast detection speeds and high precision but also possess strong model generalisation and adaptability, thereby satisfying the requirements for real-time detection. These algorithms have been widely applied to the task of fruit tree detection in orchards. Akdoğan et al. (2025) proposed two different methods, called UP-YOLO and PP-YOLO, for detecting cherry and apple trees. Their results showed that PP-YOLO achieved values for the F1-score and mAP@0.5 that were 1.5% and 1.4% higher than for UP-YOLO, respectively. Ling et al. (2024) proposed an improved lightweight precise detection algorithm for jujube tree trunks based on YOLOv8; compared to the original model, the precision of the improved model increased by 2.4%, the recall by 1.4%, and the mAP@0.5 by 1.8%. Xu & Rai (2024) designed a vision-based autonomous navigation system and proposed an improved algorithm based on the YOLOP network that was capable of simultaneously detecting trunks, obstacles, and traversable areas from RGB images with high precision (with an mAP@0.5 of 96.7%). The YOLO algorithm has also been widely applied in the domain of fruit recognition and detection in orchards. Liang et al. (2025) proposed a novel recognition method based on the YOLOv7-MSRSF model to detect litchi fruits, and their experimental results showed that the mAP for YOLOv7-MSRSF reached 96.1%. Li et al. (2024b) studied the detection of elevated strawberries and their picking points with a strawberry harvesting robot based on the YOLOv7 target detection algorithm and RGB-D sensing technology. The mAP@0.5 for the improved model reached 96.8%. Freitas et al. (2025) presented BananaRipe, a web application with mobile-friendly computer vision features that was designed to determine the ripeness level of bananas using a YOLO-v8-based classification model. Their test results showed that BananaRipe achieved values of 95.96% of accuracy and 97.59% for recall. Feng et al. (2024) proposed an improved YOLOv9c network model for identifying and detecting blueberry fruits at different stages of maturity. Their experimental results showed that the mAP@0.5 for the improved YOLOv9c network was increased by 0.7%.

Current schemes for the detection of orchard trees face challenges such as complex lighting variations, foliage occlusion, and similarity in trunk textures, which affect the generalisation capability and robustness of these models. Systems for fruit detection are also confronted with issues such as mutual occlusion among fruits and changes in colour and shape with ripening, which impose stringent demands in terms of the robustness and feature discrimination ability of the model. Although some studies have attempted to address the challenges associated with orchard object detection, most existing algorithms suffer from drawbacks such as high computational resource consumption and weak generalisation capability. Moreover, they are significantly affected by the complex orchard environment, and fail to meet the practical needs of orchard production. To address these issues, this paper proposes an orchard target detection algorithm based on YOLOv9c, which provides a high-precision, real-time solution for environmental perception and dynamic path planning in intelligent orchard agricultural machinery.

Material and Methods

YOLOv9 Target Detection Network

The YOLO series of algorithms are highly regarded in the field of target detection, and are favoured in numerous domains for their high speeds and respectable precision balance (Jiao & Abdullah, 2024). YOLOv9 (Wang et al., 2024), which is built upon the structure of YOLOv7, yields enhanced efficiency and precision of target detection through the use of the ELAN (GELAN) architecture and programmable gradient information. Compared to other versions, YOLOv9 can more accurately recognise and locate various target objects in complex scenes, including small objects, occluded objects, and overlapping objects, thus providing more reliable operation for practical applications.

In orchards, the primary recognition targets are fruit trees and the various obstacles that can hinder orchard operation robots. These targets collectively form a complex orchard environment. Due to the significant interference to the operation of a robot caused by obstacles and the mutual occlusion between tree trunks and branches/leaves, there are high demands on the precision and real-time performance of network models for detecting various orchard targets. The YOLOv9c version offers a higher inference speed and better detection performance compared to the base YOLOv9 model, and is therefore selected in this work as the foundational model. Based on this model, improvements are made to the network structure to enhance its performance in regard to recognising and detecting various target objects in a complex orchard environment. The main improvements to the model are threefold: (i) in order to enhance the network's resistance to interference in complex environments, a spatial-channel synergistic attention (SCSA) module is introduced to refine the RepNCSPELAN4 module in the backbone network; (ii) to increase the network's inference speed, a GSConv module is incorporated to

replace the ADown module in the neck network, while the RepNCSPELAN4 module in the neck network is replaced with the more lightweight VoVGSCSP module; (iii) by introducing the Shape-IoU loss function, the performance of the bounding box regression algorithm is improved, thereby enhancing the network's robustness against interference. The structure of the improved YOLOv9c model is shown in Figure 1.

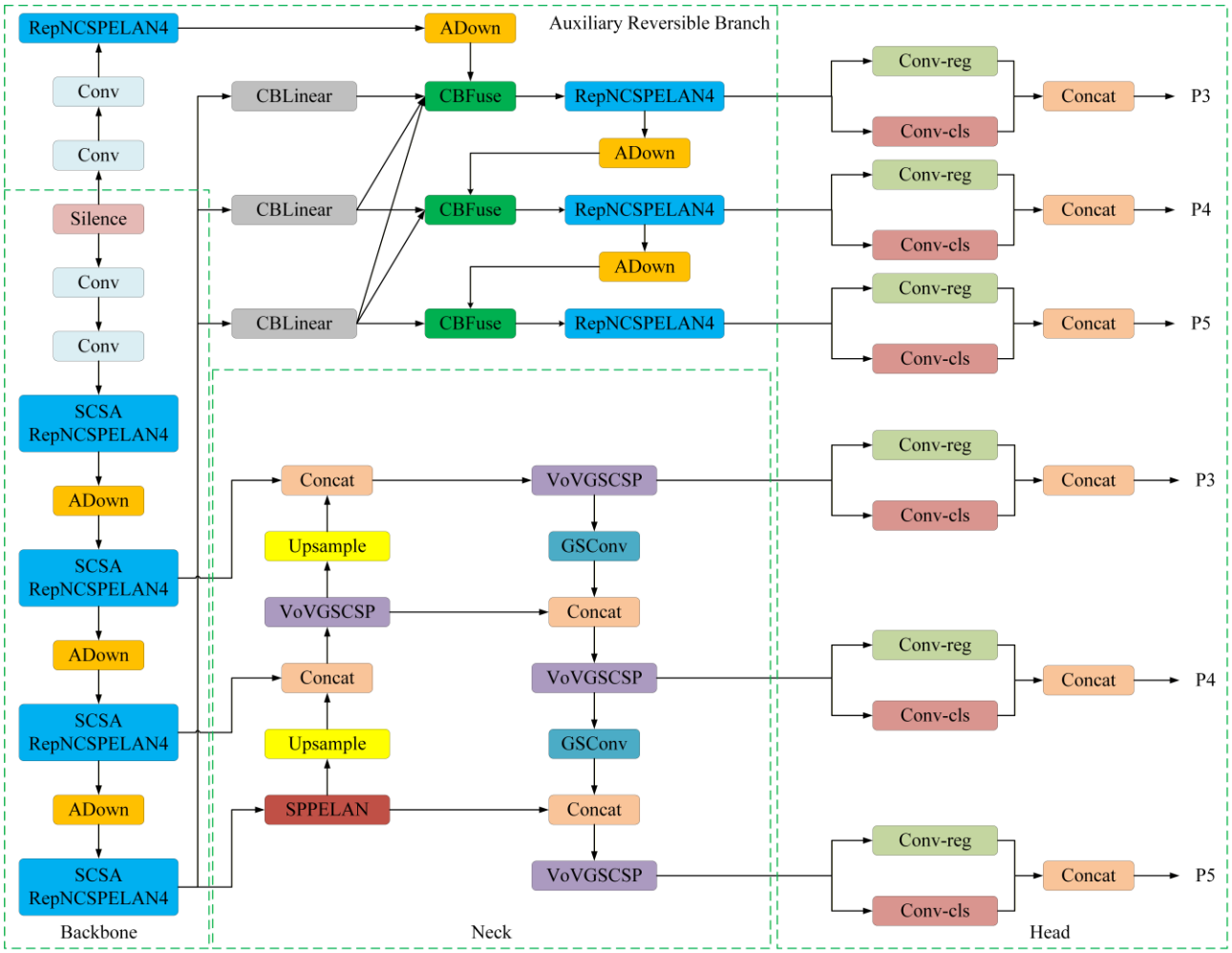

Figure 1. Structure of the improved YOLOv9c model.

SCSA Attention Module

The SCSA module (Si et al., 2025) is a lightweight feature enhancement module that combines multi-scale spatial attention with self-attention-driven channel attention. The core objective of this approach is to enable dynamic optimisation of the importance weights of the spatial and channel dimensions in the feature maps while reducing the computational cost. The structure of the SCSA module is shown in Figure 2, and consists of two parts: shareable multi-semantic spatial attention (SMSA) and progressive channel-wise self-attention (PCSA). SMSA fuses multi-semantic information and applies a progressive compression strategy to inject discriminative spatial prior information into the channel self-attention of PCSA, thereby effectively guiding channel recalibration. Robust feature interaction based on channel-wise single-head self-attention in PCSA further mitigates the differences in multi-semantic information between the different sub-features in SMSA. Simultaneously, channel compression is avoided to prevent the loss of key features. Through the self-attention mechanism, each channel in SCSA is allowed to associate information dynamically with other channels, which significantly enhances the channel interaction capabilities.

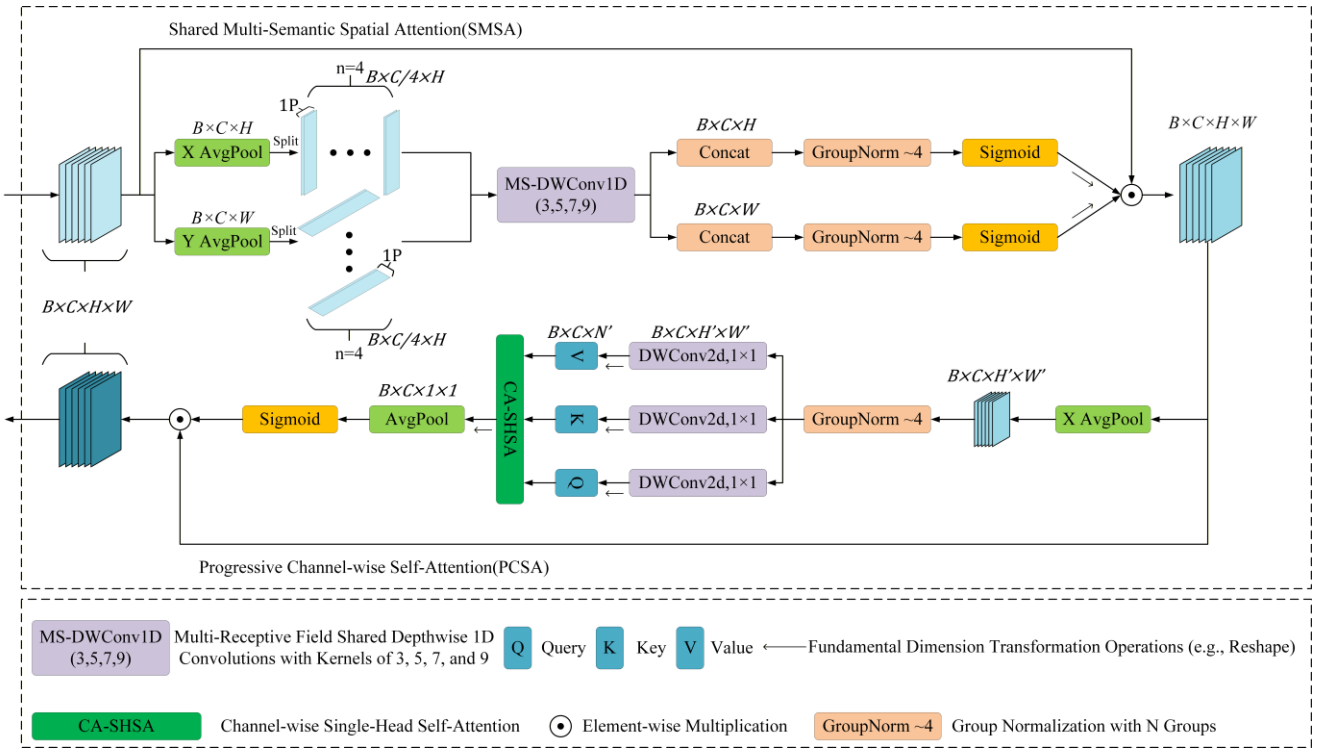

Figure 2. Structure diagram of the SCSA module, which uses multi-semantic spatial information to guide the learning of channel-wise self-attention. B denotes the batch size, C signifies the number of channels, and H and W correspond to the height and width of the feature maps, respectively. The variable n represents the number of groups into which sub-features are divided, and 1P denotes a single pixel.

In this paper, the SCSA module is not simply added to the backbone network of the YOLOv9c model; instead, it is used to modify the RepNCSPELAN4 module within the backbone. Specifically, it is introduced to replace the Conv block in the third branch of the RepNCSPELAN4 module, as shown in Figure 3. In this process, we maintained the same numbers of input and output channels as in the original convolutional block, to allow the SCSA module to be directly integrated into the existing architecture without altering the dimensions of the feature maps. This approach embeds the attention mechanism directly into the module's feature extraction workflow, without adding extra layers, thereby preserving the compactness of the module.

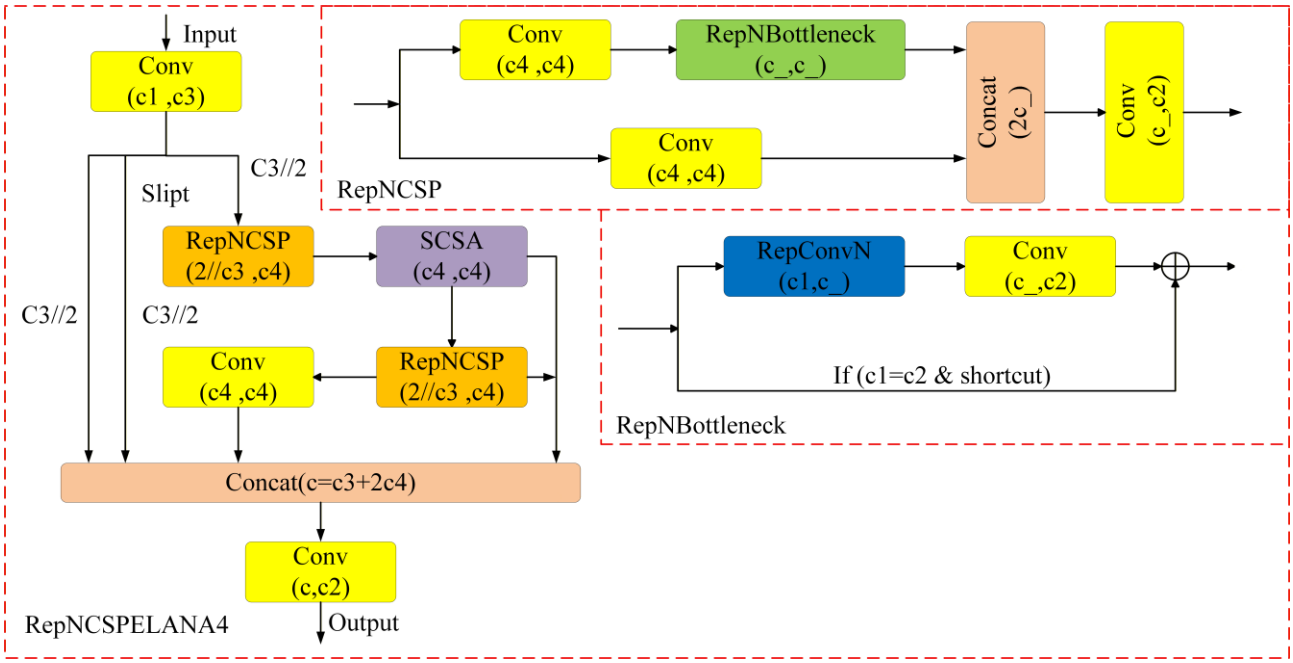

Figure 3. Structure of the SCSARepNCSPELAN4 module, in which Conv is replaced with SCSA in branch 3 to improve attention to occluded features.

GSConv Convolution Module

GSConv (Li et al., 2024a) is an efficient convolution module based on an improved version of depthwise separable convolution (DSC). The core idea is to optimise the computational efficiency of a convolutional neural network (CNN) through channel separation, feature mixing, and lightweight design while maintaining the feature representation capability. GSConv consists of a hybrid operation combining a standard convolution (Conv) with DSC. In the module, a 1×1 convolution is first used to compress the number of input channels to half the number of output channels. Then, multi-scale features are extracted from the halved feature map via 5×5 depthwise convolution. The features extracted in the above two steps are concatenated along the channel dimension (dim=1). Finally, feature mixing is achieved through channel concatenation and feature shuffling. Compared to a standard convolution, the GSConv module has reduced numbers of parameters and computations through the use of depthwise convolution and channel shuffling.

The VoVGSCSP module is a lightweight CSP module based on GSConv, in which the gradient optimisation of the CSP structure is combined with the lightweight multi-scale feature extraction of GSConv, thus enhancing the multi-scale feature fusion capabilities of the model while reducing the computational load. During the experiment, images with a size of 640×640 were input into the model. After the features had been extracted by the backbone network, they were passed to the VoVGSCSP module. This module processed the input feature maps at multiple scales, including 80×80 (for small object detection), 40×40 (for medium object detection), and 20×20 (for large object detection). This design significantly reduced the number of parameters and computational cost across feature maps of different scales while enhancing the multi-scale feature fusion capability of the network through the use of efficient cross-layer connections. As a result, the overall computational efficiency of the model was optimised while maintaining high detection accuracy. The structures of the GSConv and VoVGSCSP modules are shown in Figures 4 and 5, respectively.

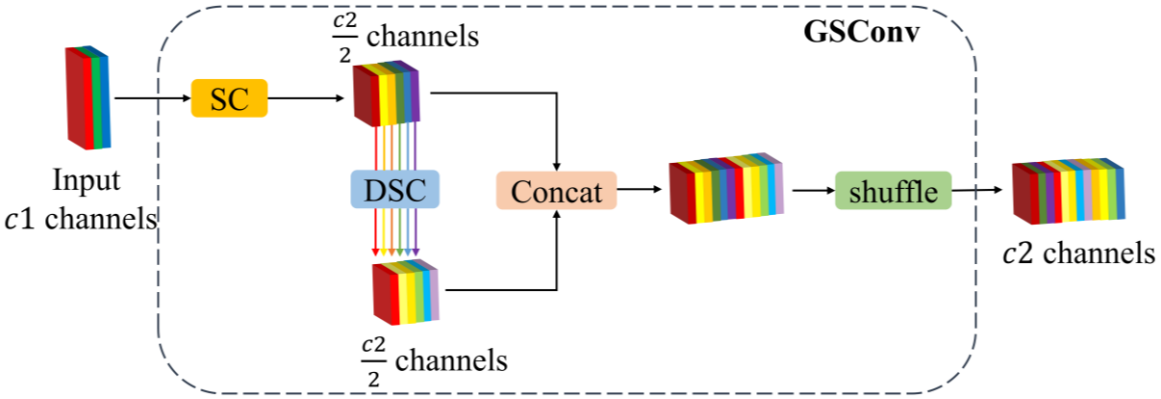

Figure 4. Structure of the GSConv. SC represents a standard 2D convolutional layer with a 2D batch normalisation layer and an activation layer. DSC represents a depth-wise 2D convolutional layer with 2D batch normalisation and an activation layer.

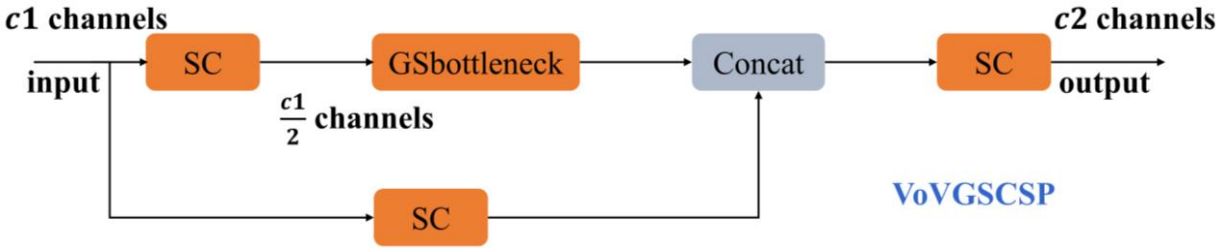

Figure 5. Structure of the VoVGSCSP module.

Shape-IoU Loss Function

The original YOLOv9 model uses the CIoU loss function, which takes into account only the difference in the aspect ratio of the bounding box, rather than the true difference between the aspect ratio and its confidence, thereby hindering the optimisation of the model for similarity. Furthermore, existing bounding box regression methods typically consider the geometric relationship between the ground truth (GT) box and the predicted box, using their relative positions and shapes to calculate the loss, while ignoring the influence of inherent attributes such as the shapes and scales of the bounding boxes themselves on the regression process.

To address the shortcomings of existing schemes, a bounding box regression method was adopted in this study that was based on the shape and scale of the bounding box itself. Yang et al. (2024) proposed the Shape-IoU method, in which the loss is calculated by focusing on the shape and scale of the bounding box itself, leading to more accurate bounding box regression. Hence, to effectively solve the problem of detection box distortion caused by overlap between fruit trees and to reduce the occurrence of missed detections, CIoU was replaced with the Shape-IoU loss function in our model.

The specific formulas used for this calculation are as follows:

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}} \tag{1}$$

$$\omega\omega = \frac{2 \times (\omega^{gt})^{scale}}{(\omega^{gt})^{scale} + (h^{gt})^{scale}} \tag{2}$$

$$hh = \frac{2 \times (h^{gt})^{scale}}{(\omega^{gt})^{scale} + (h^{gt})^{scale}} \tag{3}$$

$$distance^{shape} = hh \times \frac{(x_c - {x_c}^{gt})^2}{c^2} + \omega\omega \times \frac{(y_c - {y_c}^{gt})^2}{c^2} \tag{4}$$

$$\Omega^{shape} = \sum_{t=\omega,h} (1 - e^{-\omega t})^{\theta}, \theta = 4 \tag{5}$$

$$\omega_{\omega} = hh \times \frac{|\omega - \omega^{gt}|}{\max(\omega, \omega^{gt})} \tag{6}$$

$$\omega_{h} = \omega\omega \times \frac{|h - h^{gt}|}{\max(h, h^{gt})} \tag{7}$$

where:

B and B^{gt} are the predicted bounding box and the ground truth (GT) box;

c is the diagonal length of the smallest enclosing bounding box covering both B and B^{gt};

ω^{gt} and h^{gt} are the width and height of the GT box;

ω and h represent the width and height of the anchor box;

$scale$ is a scaling factor related to the target size distribution in the dataset (and is set to 0.5);

$\omega\omega$, hh are the weighting coefficients for the horizontal and vertical directions, respectively, whose values depend on the aspect ratio of the GT box. The weighting coefficients are used to adjust the intensity of the penalty based on the aspect ratio of the ground truth box. When the shape of the predicted box deviates significantly from that of the ground truth box in a certain direction, the penalty in that direction is increased; for example, when detecting tall tree trunks, deviations in the height of the predicted box incur a more severe penalty than deviations in its width. This mechanism helps to balance the influence of predicted boxes with different aspect ratios in subsequent loss calculations, enabling the model to handle various complex environments.

The corresponding bounding box regression loss is given by:

$$L_{Shape-IoU} = 1 - IoU + distance^{shape} + 0.5 \times \Omega^{shape} \tag{8}$$

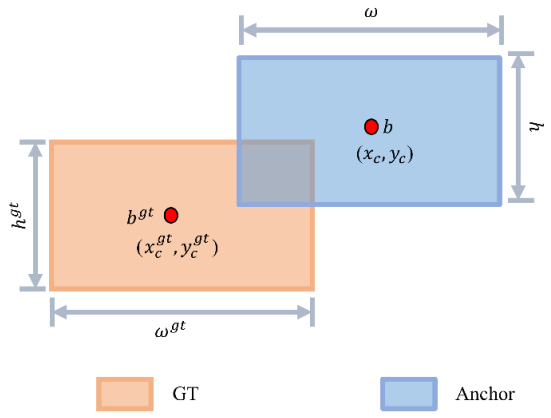

Figure 6. Illustration of the process used in Shape-IoU.

Results and Discussion

Data Collection and Processing

The images used for this experiment were collected from a peach orchard in Zhenjiang City, Jiangsu Province. A camera was used to collect environmental data from the orchard. During shooting, the camera was positioned approximately 60 cm above the ground (equivalent to the height of installation on an orchard operating robot). The images captured in the experiment had two sizes, 1280×720 pixels and 1920×1440 pixels, to enable the model to handle images of different resolutions and aspect ratios. Some

images were also captured to enrich the data for the "people" category. A total of 1,213 images were collected, and Figure 7(a) shows some examples of these original images. To enhance the generalisation ability of the model, image augmentation was performed on these images to expand the dataset. Figure 7(b) shows some of these augmented images. After augmentation, a total of 2,626 images were obtained, and based on the natural conditions of the orchard, the targets were categorised into three classes: tree trunks, people, and obstacles. The distribution was as follows: 2,001 images in the trees category, 689 images in the people category, and 1,495 images in the obstacles category. The environmental conditions represented in the images included sunny (50%), cloudy (25%), overcast (15%), and rainy (10%) weather.

The essential detection targets were tree trunks. However, in early-stage intelligent orchards, orchard robots require human assistance during their operation. There are also obstacles in the environment, primarily fruit boxes, tools, and low-growing vegetation, which are commonly found in orchard environments. Hence, the detection of both humans and obstacles is indispensable in orchard settings. The three target categories (trees, people, and obstacles) were annotated using the LabelImg annotation tool to generate corresponding label files. After labelling, the images and their corresponding label files were divided into training, validation, and test sets in a ratio of 7:2:1. During the dataset partitioning process, data needed to be randomly allocated, while ensuring that the proportion of each category in every subset matched that of the entire dataset, to avoid any sequential or selection bias. In addition, each subset needed to contain a certain proportion of both original and augmented images. These datasets were then used for training of the model, parameter optimisation, and comparison of prediction results, to evaluate the model's performance.

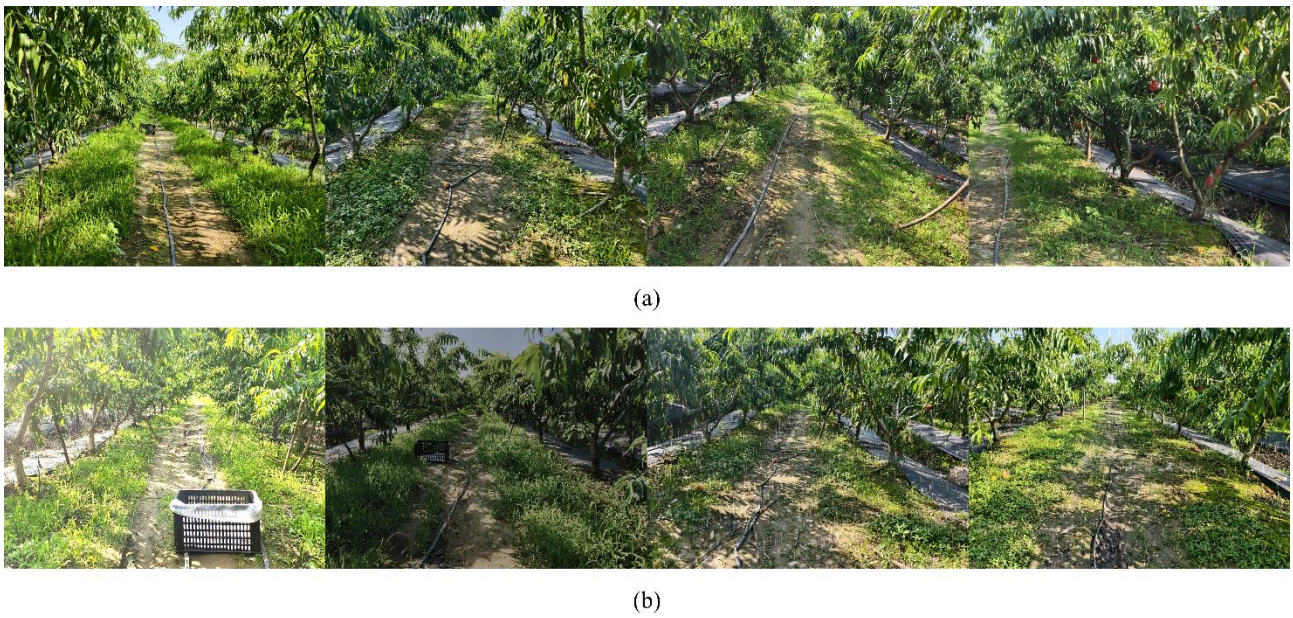

Figure 7. Photographs of the orchard: (a) original images; (b) augmented images.

Analysis of Results and Evaluation

Testing Environment

The software testing environment consisted of a deep learning framework based on Windows 11, Python 3.12, PyTorch 2.6.0, and CUDA 12.9. The hardware testing environment included an Intel i5-13500HX CPU and an NVIDIA GeForce RTX 4060 12G graphics card. The size of the input images was set to the standard dimensions of 640×640 pixels, the maximum number of iterations was set to 200, and the batch size was set to four. The learning rate scheduler employed the OneCycleLR strategy, with an initial learning rate of 0.01.

Ablation Experiments

Ablation experiments were conducted on the experimental dataset to analyse and validate the precision contributions of the SCSA module, GSConv module, and Shape-IoU loss function. The improvements to YOLOv9c were added incrementally, to create seven different models, with the initial parameters kept consistent for all models during the training phase. Table 1 shows the overall results from YOLOv9c after adding the different modules.

From Table 1, it can be seen that the addition of the SCSA module, GSConv module, or Shape-IoU loss function individually did not significantly improve the mAP@0.5. Adding the GSConv module increased the mAP@0.5 by 0.5%, while reducing the model size (Params) and FLOPs by 7.2% and 8.1%, respectively. However, a combination of these modules improved the mAP@0.5 while reducing the number of parameters. As shown in Table 1, adding both the SCSA and GSConv modules to YOLOv9c increased the mAP@0.5 and F1-score by 1.2% and 0.7%, respectively. Finally, combining the original YOLOv9c model with the SCSA

module, GSConv module, and Shape-IoU loss function resulted in a 2% increase in mAP@0.5, a 1.2% increase in F1-score, and reductions of 9.7% in parameters and 10.5% in FLOPs. The inference speed was evaluated on an NVIDIA RTX 4060 graphics processor. The improved YOLOv9-c model achieved a frame rate of 37 FPS, while the value for the standard YOLOv9-c model was 32 FPS. This result confirms that the proposed model is suitable for real-time orchard robotic applications. In addition, the confidence interval for the performance metrics of the YOLOv9-c model was [0.8991, 0.9343] at a 95% confidence level, while the improved YOLOv9-c model showed a significant performance enhancement, with a confidence interval of [0.9261, 0.9573], indicating that the enhanced model has a stronger detection capability and stability. Based on these data, it is evident that the improved model yields significantly enhanced detection performance while also becoming more lightweight.

Table 1. Results of the ablation experiments.

SCSA	GSConv	Shape-IoU	P	R	mAP@0.5	F1	Params	FLOPs(G)
			0.96	0.941	0.953	0.95	50702898	236.6
√			0.958	0.946	0.954	0.952	49349490	230.5
	√		0.962	0.943	0.958	0.953	45773426	211.7
		√	0.962	0.943	0.958	0.952	50702898	236.6
√	√		0.964	0.951	0.965	0.957	45773426	211.7
√		√	0.961	0.946	0.963	0.953	49349490	230.5
	√	√	0.959	0.952	0.961	0.955	47126834	217.8
√	√	√	0.97	0.954	0.973	0.962	45773426	211.7

The context in which an orchard robot moves during its work typically includes inter-row areas and orchard edge areas. In practical operations, adverse weather conditions such as rain and fog are inevitable. Figure 8 illustrates the detection performance of the improved YOLOv9c model in various scenarios. From Figure 8(a), it can be observed that when the orchard robot moves normally between tree rows, the tree trunks are accurately detected. Figure 8(b) shows the detection results when a person and weeds are present in the captured image, and demonstrates that the algorithm can accurately identify these targets. In Figure 8(c), it can be seen that when obstacles are present within the robot's field of view, both tree trunks and obstacles are accurately detected, which aids in operations such as route changes. As shown in Figure 8(d), even when the image is partially overexposed, the algorithm can still detect tree trunks within the overexposed region. From Figures 8(e) and (f), we see that even under cloudy and rainy conditions, there are no missed detections. Overall, from Figure 8, it can be concluded that the improved model presented in this paper accurately detects tree trunks, obstacles, and people under different scenarios and weather conditions, demonstrating good performance.

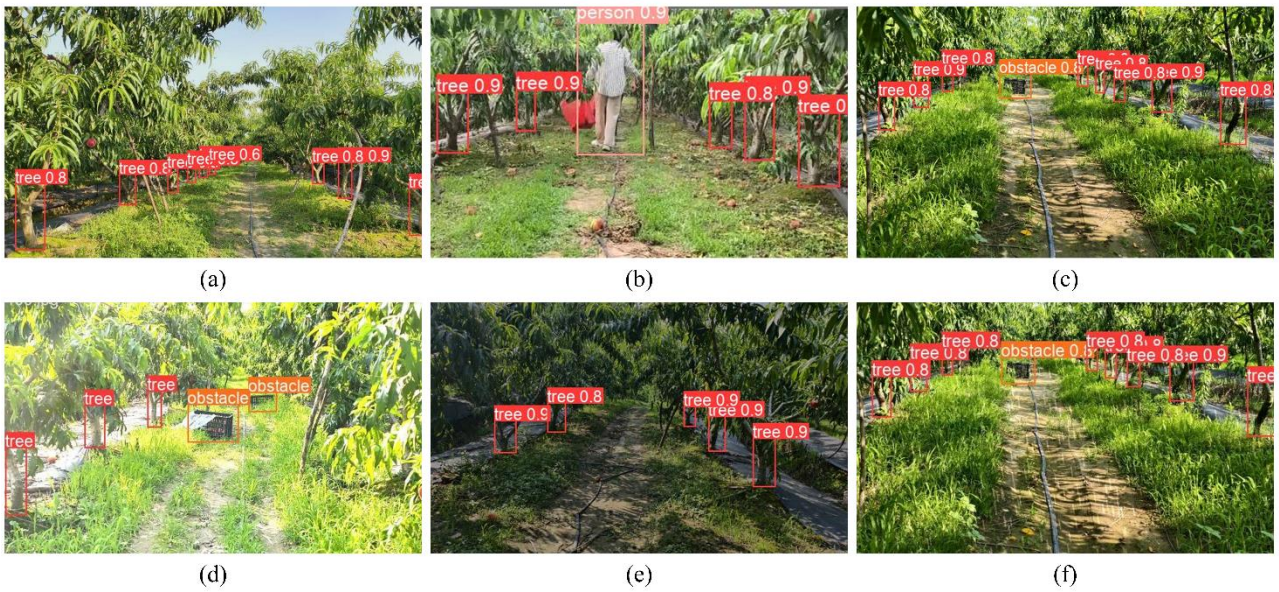

Figure 8. Detection results in different scenarios: (a) moving between tree rows; (b) with weeds and a person; (c) with obstacles; (d) local overexposure; (e) overcast; (f) rainy.

Comparison of training results from different models

To validate the detection effectiveness of the improved method, the enhanced YOLOv9c model was compared with mainstream real-time detection models such as YOLOv5n, YOLOv7, YOLOv8, YOLOv9c, and YOLOv11. The dataset and class definitions used in the experiment were identical, and all model weights were randomly initialised. Table 2 shows the overall results from different detection models.

From Table 2, it can be seen that compared to the original YOLOv9c model, the improved model achieved gains in the precision (P), recall (R), mAP@0.5, and F1-score. Embedding GSConv and SCSA into the improved YOLOv9c model reduced the number of parameters while enabling the model to learn key features more comprehensively, thereby enhancing its adaptability to the detection of tree trunks, people, and obstacles. The improved YOLOv9c model achieved improvements in the average precision (mAP@0.5) of 2.9%, 3.8%, 2.4%, and 2.5% over YOLOv5n, YOLOv7, YOLOv8, and YOLOv11, respectively, with an F1-score that was increased by 2.2%, 2.4%, 1.5%, and 2.3%, respectively. In summary, the improved YOLOv9c model outperformed other real-time detection models across all evaluation metrics, achieved the best balance between precision and recall, and excelled at detecting fruit tree trunks and other target objects in complex orchard environments.

Table 2. Results of comparative experiments.

Model	P	R	mAP@0.5	F1
YOLOv5n	95.7%	92.3%	94.4%	94%
YOLOv7	95.2%	92.4%	93.5%	93.8%
YOLOv8	95.5%	94%	94.9%	94.7%
YOLOv9c	96%	94.1%	95.3%	95%
YOLOV11	95%	92.8%	94.8%	93.9%
Improved YOLOv9c	97%	95.4%	97.3%	96.2%

Figure 9 shows a comparison of the mAP@0.5 results during the training process for the different network models. From the figure, it can be seen that the improvements in the mAP@0.5 for the proposed model are not as rapid as for the other models in the early stages of training; however, as the number of training epochs increases, the improved YOLOv9c model gradually surpasses the others and maintains this lead until the end of the training process. A comparison with other object detection models demonstrates that the improved YOLOv9c model is suitable for target detection tasks in complex orchard scenes and can meet the real-time requirements of mobile terminals.

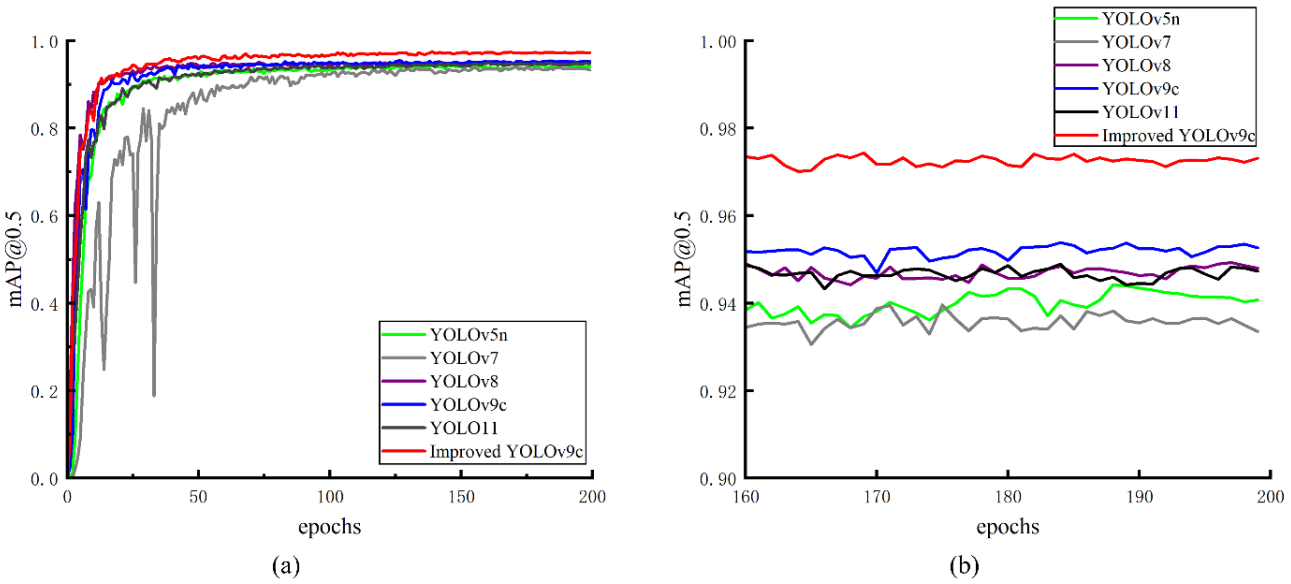

Figure 9. Comparison of results from different network models: (a) original comparison image; (b) expanded comparison image.

Conclusions

The ability to identify fruit tree trunks in orchards is essential to enable orchard robots to successfully complete various tasks. However, the complexity of an orchard environment severely impacts detection performance. With the aim of enhancing the detection efficiency under such conditions, this paper has presented a recognition model based on YOLOv9 for target objects including fruit tree trunks, people, and obstacles.

In summary, the research presented in this paper involved the following:

(1) With a focus on robots in orchard operation scenarios, images of an orchard were collected, augmented, and compiled into a dataset to ensure the precision of the experimental data.

(2) To improve the YOLOv9c model, the SCSA attention mechanism was introduced into the backbone network to enhance the channel interaction capabilities, particularly in semantically complex scenes. The GSConv and VoVGSCSP modules were also introduced to improve the neck network, which reduced the complexity and computational cost of the model.

(3) The precision, recall, and mean average precision (mAP@0.5) achieved by the improved model reached 97%, 95.4%, and 97.3%, respectively, representing improvements of 1%, 1.3%, and 2% compared to the original model. Extensive ablation studies and comparative experiments demonstrated that the improved model yielded a significantly enhanced detection capability for fruit tree trunks.

The improved model offers performance that is enhanced from multiple perspectives. It can effectively reduce the probability of robots colliding with fruit trees, thereby minimising damage to trees. It also enables more comprehensive identification of tree trunks and other obstacles along the navigation path, which can improve operational safety. The model has a lightweight design while maintaining a relatively fast detection speed, facilitating long-term stable operation on mobile robotic platforms and providing reliable support for prolonged, large-scale autonomous inspection and operational tasks.

Although the improved model performed well under moderate occlusion and lighting variations, severe occlusion caused by dense foliage may still reduce the recall rate. In future work, the incorporation of depth information or temporal fusion techniques could be considered to mitigate these limitations. In addition, multi-sensor fusion and experiments with binocular stereo cameras are planned to acquire richer environmental information and further enhance the model's performance in complex and dynamic orchard environments.

Data Availability Statement

The datasets generated during the current study are available from the corresponding author on reasonable request.

References

Akdoğan, C., Özer, T., & Oğuz, Y. (2025). PP-YOLO: Deep learning based detection model to detect apple and cherry trees in orchard based on histogram and wavelet preprocessing techniques. *Computers and Electronics in Agriculture*, *232*, 110052. https://doi.org/10.1016/j.compag.2025.110052

Crespo, A., Moncada, C., Crespo, F., & Morocho-Cayamcela, M. E. (2025). An efficient strawberry segmentation model based on Mask R-CNN and TensorRT. *Artificial Intelligence in Agriculture*, *15*(2), 327–337. https://doi.org/10.1016/j.aiia.2025.01.008

Feng, W., Liu, M., Sun, Y., Wang, S., & Wang, J. (2024). The use of a blueberry ripeness detection model in dense occlusion scenarios based on the improved YOLOv9. *Agronomy*, *14*(8), 1860. https://doi.org/10.3390/agronomy14081860

Freitas, E. D. de, Martins, J. L., Neves, J. P., & Gomes, D. G. (2025). Classification of banana ripeness degree using YOLO-V8. *Engenharia Agrícola, 45*(spe1), e20240193. https://doi.org/10.1590/1809-4430-Eng.Agric.v45nespe120240193/2025

Ghazal, S., Munir, A., & Qureshi, W. S. (2024). Computer vision in smart agriculture and precision farming: Techniques and applications. *Artificial Intelligence in Agriculture, 13*, 64 - 83. https://doi.org/10.1016/j.aiia.2024.06.004

Jiao, L., & Abdullah, M. I. (2024). YOLO series algorithms in object detection of unmanned aerial vehicles: A survey. *Service Oriented Computing and Applications, 18*(3), 269–298. https://doi.org/10.1007/s11761-024-00388-w

Li, H., Li, J., Wei, H., Liu, Z., Zhan, Z., & Ren, Q. (2024a). Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *Journal of Real-Time Image Processing, 21*(3), 62. https://doi.org/10.1007/s11554-024-01436-6

Li, Y., Wang, W., Guo, X., Wang, X., Liu, Y., & Wang, D. (2024b). Recognition and positioning of strawberries based on improved YOLOv7 and RGB-D sensing. *Agriculture*, *14*(4), 624. https://doi.org/10.3390/agriculture14040624

Liang, C., Liang, J., Yang, W., Ge, W., Zhao, J., Li, Z., Bai, S., Fan, J., Lan, Y., & Long, Y. (2025). Enhanced visual detection of litchi fruit in complex natural environments based on unmanned aerial vehicle (UAV) remote sensing. *Precision Agriculture, 26*(1), 23. https://doi.org/10.1007/s11119-025-10220-w

Ling, S., Wang, N., Li, J., & Ding, L. (2024). Accurate recognition of jujube tree trunks based on contrast limited adaptive histogram equalization image enhancement and improved YOLOv8. *Forests*, *15*(4), 625. https://doi.org/10.3390/f15040625

Si, Y., Xu, H., Zhu, X., Zhang, W., Dong, Y., Chen, Y., & Li, H. (2025). SCSA: Exploring the synergistic effects between spatial and channel attention. *Neurocomputing*, *634*, 129866. https://doi.org/10.1016/j.neucom.2025.129866

Vijayakumar, A., & Vairavasundaram, S. (2024). YOLO-based object detection models: A review and its applications. *Multimedia Tools and Applications, 83*(35), 83535–83574. https://doi.org/10.1007/s11042-024-18872-y

Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2024, September). YOLOv9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision* (pp. 1–21). Cham: Springer Nature Switzerland. https://doi.org/10.48550/arXiv.2402.13616

Wang, H., Xu, X., Liu, Y., Lu, D., Liang, B., & Tang, Y. (2023). Real-time defect detection for metal components: A fusion of enhanced Canny–Devernay and YOLOv6 algorithms. *Applied Sciences, 13*(12), 6898. https://doi.org/10.3390/app13126898

Wang, H., Zhang, G., Cao, H., Hu, K., Wang, Q., Deng, Y., Gao, J., & Tang, Y. (2025). Geometry-Aware 3D point cloud learning for precise cutting-point detection in unstructured field environments. *Journal of Field Robotics*, *42*(7), 3063 - 3076. https://doi.org/10.1002/rob.22567

Xu, S., & Rai, R. (2024). Vision-based autonomous navigation stack for tractors operating in peach orchards. *Computers and Electronics in Agriculture*, *217*, 108558. https://doi.org/10.1016/j.compag.2023.108558

Yang, X., Zhao, W., Wang, Y., Yan, W. Q., & Li, Y. (2024). Lightweight and efficient deep learning models for fruit detection in orchards. *Scientific Reports, 14*(1), 26086. https://doi.org/10.1038/s41598-024-76662-w

Zhang, L., Li, M., Zhu, X., Chen, Y., Huang, J., Wang, Z., Hu, X., Wang, T., Wang, Z., & Fang, K. (2024). Navigation path recognition between rows of fruit trees based on semantic segmentation. *Computers and Electronics in Agriculture, 216*, 108511. https://doi.org/10.1016/j.compag.2023.108511