Synchronous detection method for litchi fruits and picking points of a litchi-picking robot based on improved YOLOv8-pose

Hongxing Peng[1,2,3], Qijun Liang[1], Xiangjun Zou[2,4], Hongjun Wang[2,5], Juntao Xiong[1,2*], Yanlin Luo[1], Shangkun Guo[1], Guanjia Shen[1]

(1. *College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China*;
2. *Foshan-Zhongke Innovation Research Institute of Intelligent Agriculture and Robotics, Foshan528251, Guangdong, China*;
3. *Key Laboratory of Smart Agricultural Technology in Tropical South China, Ministry of Agriculture and Rural Affairs, Guangzhou 510642, China*;
4. *College of Intelligent Manufacturing and Modern Industry, Xinjiang University, Urumqi 830046, Xinjiang, China*;
5. *College of Engineering, South China Agricultural University, Guangzhou 510642, China*)

Abstract: In the unstructured litchi orchard, precise identification and localization of litchi fruits and picking points are crucial for litchi-picking robots. Most studies adopt multi-step methods to detect fruit and locate picking points, which are slow and struggle to cope with complex environments. This study proposes a YOLOv8-iGR model based on YOLOv8n-pose improvement, integrating end-to-end network for both object detection and key point detection. Specifically, this study considers the influence of auxiliary points on picking point and designs four litchi key point strategies. Secondly, the architecture named iSaE is proposed, which combines the capabilities of CNN and attention mechanism. Subsequently, C2f is replaced by Generalized Efficient Layer Aggregation Network (GELAN) to reduce model redundancy and improve detection accuracy. Finally, based on RFAConv, RFAPoseHead is designed to address the issue of parameter sharing in large convolutional kernels, thereby more effectively extracting feature information. Experimental results demonstrate that YOLOv8-iGR achieves an AP of 95.7% in litchi fruit detection, and the Euclidean distance error of picking points is less than 8 pixels across different scenes, meeting the requirements of litchi picking. Additionally, the GFLOPs of the model are reduced by 10.71%. The accuracy of the model's localization for picking points was tested through field picking experiments. In conclusion, YOLOv8-iGR exhibits outstanding detection performance along with lower model complexity, making it more feasible for implementation on robots. This will provide technical support for the vision system of the litchi-picking robot.

Keywords: litchi, object detection, picking point detection, YOLOv8-pose, picking robot
DOI: 10.25165/j.ijabe.20251804.9303

Citation: Peng H X, Liang Q J, Zou X J, Wang H J, Xiong J T, Luo Y L, et al. Synchronous detection method for litchi fruits and picking points of a litchi-picking robot based on improved YOLOv8-pose. Int J Agric & Biol Eng, 2025; 18(4): 266–274.

1 Introduction

China is the leading producer of litchi, accounting for over half of the world's total production[1]. However, litchi harvesting is the most time-consuming and labor-intensive part of the entire fruit production cycle, due to its seasonality, high labor intensity, and significant costs. Currently, litchi harvesting relies on manual labor, resulting in high labor intensity and low picking efficiency[2]. Therefore, to reduce fruit production costs and increase farmers' income, there is an urgent need to develop intelligent harvesting robots suitable for litchi.

When harvesting litchi, it is necessary to first identify the picking point on the main fruit-bearing branch and then proceed with cutting to prevent damage to the fruit[3]. Therefore, accurate identification and localization of the picking point are crucial issues for achieving intelligent operation of litchi harvesting robots. In natural environments, the challenges of accurately detecting the picking point on the litchi include: 1) complex backgrounds in outdoor orchards; 2) occlusion; and 3) lighting conditions[4].

With the rapid development of deep convolutional neural networks in object recognition, some researchers have applied deep learning to litchi picking recognition. Peng et al.[5] constructed a ResDense-focal-DeepLabV3+ network for accurate segmentation of litchi branches in orchard environments, achieving improved mIoU in simple, medium, and complex images compared to other models. Qi et al.[2] employed YOLOv5 to detect stems in litchi images, extracting region of interest (ROI) for the main stem and segmenting them using PSPNet. Post-processing operations on the segmented images provided pixel coordinates of picking points on the main stem, with an accuracy rate of 92.50%. While these methods ensure the accuracy of picking point localization, they face challenges. The multi-step operations (e.g., ROI extraction+ segmentation+post-processing) result in significantly lower detection speeds (e.g., 15 FPS as reported in Qi et al.[2]) compared to end-to-end approaches. Such delays hinder real-time robotic operations in dynamic orchard environments.

Received date: 2024-08-18 **Accepted date:** 2025-04-20
Biographies: Hongxing Peng, Associate professor, research interest:machine vision, picking robot, Email: xyphx@scau.edu.cn; **Qijun Liang**, Master candidate, research interest: machine vision, Email: 1453468711@qq.com; **Xiangjun Zou**, Professor, research interest: picking robot, Email: xzjou1@163.com; **Hongjun Wang**, Professor, research interest: picking robot, Email: xtwhj@scau.edu.cn; **Yanlin Luo**, Master candidate, research interest: machine vision, Email: 1044425585@qq.com; **Shangkun Guo**, Master candidate, research interest: machine vision, Email: gsk_scott@163.com; **Guanjia Shen**, Master candidate, research interest: machine vision, Email: 1934936086@qq.com.
***Corresponding author: Juntao Xiong**, Professor, research interest: machine vision, picking robot, College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China. Tel: +86-13560164695, Email: xiongjt2340@163.com.

Key point detection was initially applied in human pose estimation. In recent years, some researchers have attempted to apply key point detection methods to the detection of fruit picking points. Zheng et al.[6] integrated pixel-level instance segmentation of mangoes and picking point detection into an end-to-end network. This network demonstrated strong robustness to various lighting conditions and complex backgrounds, achieving good segmentation and picking point detection performance for medium and large mangoes. Du et al.[7] proposed the YOLO-lmk model based on YOLOv5s, which combined tomato bounding box detection and key point detection. YOLO-lmk achieved a detection accuracy of 93.4% for tomato bounding boxes, with a Euclidean distance of 7.9 between the ground truth and predicted key points. The above methods integrate two tasks into an end-to-end network, improving detection speed. Existing key point detection methods primarily focus on medium-to-large fruits (mangoes, tomatoes) where high-level semantic features dominate detection performance. However, litchi fruits are smaller in size (typically 3-5 cm in diameter) and often occluded by dense branches or leaves in natural environments. Directly applying feature pyramid-based approaches (e.g., FPN) to litchi detection may lead to two issues: 1) High-level features generated by deep layers lack sufficient spatial resolution to capture fine-grained details of small litchi fruits; and 2) Multi-scale feature fusion introduces computational redundancy, which conflicts with the lightweight requirements of robotic systems. For a litchi-picking robot, the vision system needs to meet the following two requirements: Firstly, due to limited computational resources, the model needs to be lightweight without sacrificing performance. Secondly, multi-step operations are cumbersome for robots and difficult to adapt to changing environments, thus requiring fast end-to-end models. To meet the robot's needs, this paper proposes the YOLOv8-iGR detection model based on improvements to YOLOv8n-pose, achieving synchronous recognition of litchi fruits and picking points. The main contributions of this study are as follows:

1) Expand the application scenarios of key point detection by establishing a key point dataset containing multiple litchi varieties, angles, and environmental factors, and design four key point distribution strategies based on the spatial relationship between litchi fruits and key points.

2) Propose the iSaE architecture, and combine the GELAN and RFAPoseHead modules to implement a lightweight and high-performance YOLOv8-iGR algorithm.

3) Compare YOLOv8-iGR with mainstream detection algorithms in detection tasks involving differences in lighting, background factors, and branches occlusion. The results demonstrate that YOLOv8-iGR outperforms other models in terms of detection performance and efficiency.

2 Datasets

2.1 Acquisition of datasets

The image data collected for this study were captured between May 26, 2023 and June 1, 2023, in the litchi orchard at South China Agricultural University. Images of ripe litchi fruits were taken using Realsense D435i camera for training and testing purposes. The cameras were positioned 30-100 cm away from the targets. The litchi varieties included "Feizixiao", "Guiwei", and "Nuomici". The obtained images were saved as pixel RGB images. To ensure that the datasets reflected the orchard characteristics in natural environments, a total of 1281 images of different litchi tree varieties were captured within the orchard at different times (9:00 am, 2:00 pm, 6:00 pm). Representative images depicting various environmental factors are shown in Figure 1.

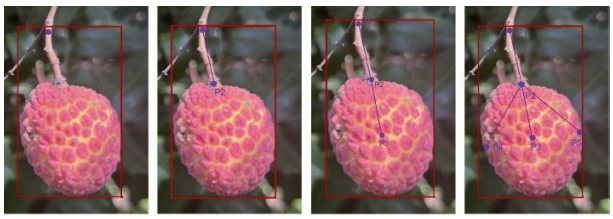
a. Adequate lighting b. Upward angle shot c. Hard light

d. Low light e. Tree leaves obstruction f. Branches obstruction

Figure 1 Representative sample datasets in different states

2.2 Annotation of datasets

Labelme was used to annotate litchi and key points. The bounding box of the mature litchi target was labeled as "mature". To further analyze the influence of the number of key points and their spatial relationships on the detection performance of picking key points, four key point strategies were designed during litchi key point annotation. Strategy 1 (1P): P1 represents the picking point on the main stem of the litchi. Strategy 2 (2P): Two key points are set, with auxiliary point P2 located at the junction of the litchi fruit and the main stem. Strategy 3 (3P): Three key points are set, with P3 defined as the centroid of the litchi fruit's bounding box. Strategy 4 (5P): Five key points are set, with P4 and P5 located on either side of the litchi fruit, symmetrically about line segment P2-P3. The four key point strategies are shown in Figure 2.



a. 1P b. 2P c. 3P d. 5P

Figure 2 Four key point strategies

The datasets are randomly partitioned into a training set (1025 images), a validation set (128 images), and a test set (128 images), following an 8:1:1 ratio.

3 Methodologies

3.1 YOLOv8n-pose model improvement strategy

The design intention of YOLOv8n-pose is based on a single-stage human key point detection model. Compared to human poses, litchi present variations in the number of key points and significant differences in features, so improvements are needed for YOLOv8n-pose to enhance the model's performance in litchi and picking point detection. In this study, the iSaE structure is proposed, which integrates the capabilities of both convolution and attention mechanisms. Compared to using a convolution or an attention mechanism alone, the iSaE module significantly improves the model's performance. Additionally, to deploy the model on mobile devices, the introduction of GELAN achieves a reduction in computational overhead without sacrificing performance. The RFAPoseHead detection head is used to simultaneously detect the key points of both the litchi fruit and the main branch. It is built

upon the foundation of RFAConv to enhance the efficiency of feature extraction. The network architecture of YOLOv8-iGR is depicted in Figure 3, with theoretical analysis and implementation details of each module elaborated in Sections 3.2-3.4.

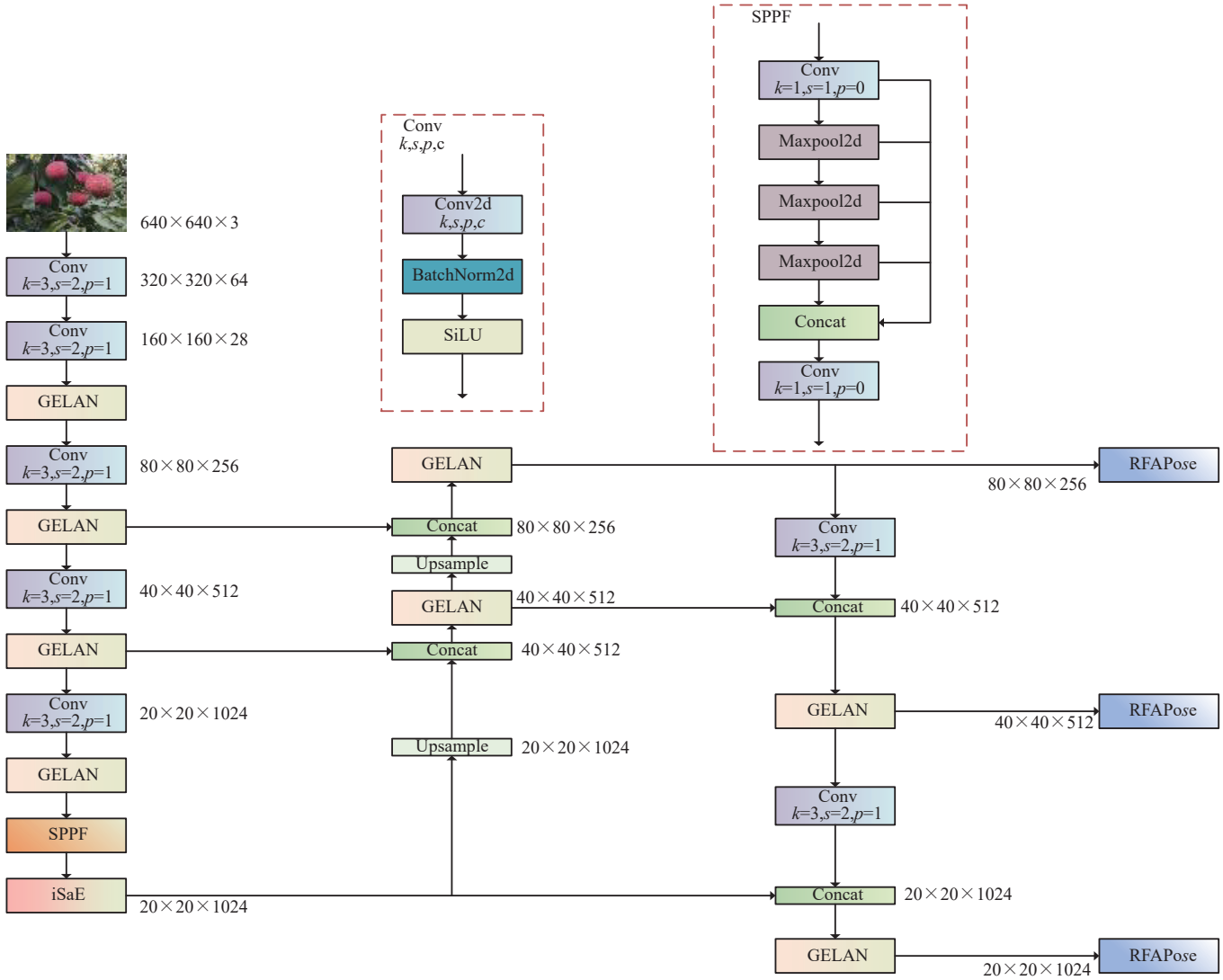


Figure 3 YOLOv8-iGR overall framework and its constituent modules

3.2 iSaE Block

In recent years, the effectiveness of attention mechanisms in enhancing model expressiveness and precise object localization has been well-validated in the realm of computer vision. The Squeeze aggregated Excitation (SaE) module, proposed by SENetv2[8], is a novel aggregated multilayer perceptron. Compared to SE[9], this fusion enhances the network's ability to capture channel information and global knowledge.

The working principle of SaE is illustrated in Figure 4. Input features are adjusted in size and channel count through the convolutional layer. These features are then aggregated using a global average pooling layer to capture channel information. The aggregated information is passed through FC layers for squeezing. Subsequently, the outputs of all FC branches are concatenated and subjected to excitation operations to restore the size to its initial shape. Finally, the squeezed and excited results are concatenated with the input of the residual module. The formula for the SaE module is articulated as follows:

$$\mathrm{SaE} = x + F\left(x \cdot E_x\left(\sum S_q(x)\right)\right) \tag{1}$$

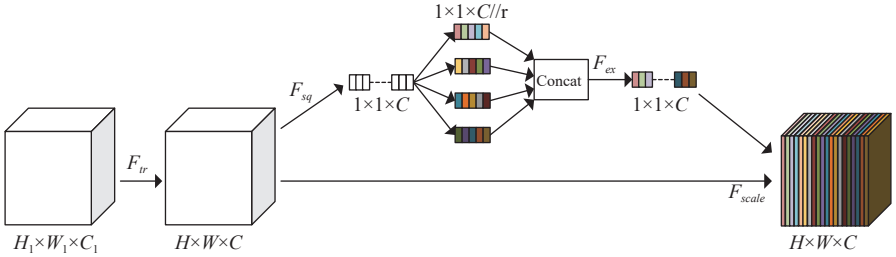


Figure 4 Working principle of SaE

In Equation (1), the 'S_q' function represents the squeezing operation, which includes the FC layers. The 'E_x' function represents the excitation operation.

Inspired by Zhang et al.[10], this study focuses on designing a module that combines lightweight CNN with attention mechanisms, called iSaE. It absorbs the efficient feature extraction capabilities of the Inverted Residual Block (IRB) from the CNN architecture[11] to model local features, as well as the comprehensive information capture capabilities of the SaE architecture to model global features. The structure of iSaE is depicted in Figure 5. Figure 6 illustrates the attention heat map for picking points generated by IRB, SaE, and iSaE. The influence on picking point localization correlates with the intensity of the red hue in its heat map. It can be observed that the picking point localization by IRB deviates from the branch. Due to the fixed weight parameters used in computing attention weights, SaE is susceptible to interference from background information, severely affecting the localization of the picking point. In contrast, iSaE combines the advantages of both approaches and accurately predicts the picking point location. The introduction of the iSaE module effectively addresses the issues of accuracy degradation caused by lightweight CNN and the deficiencies in the SaE attention mechanism.

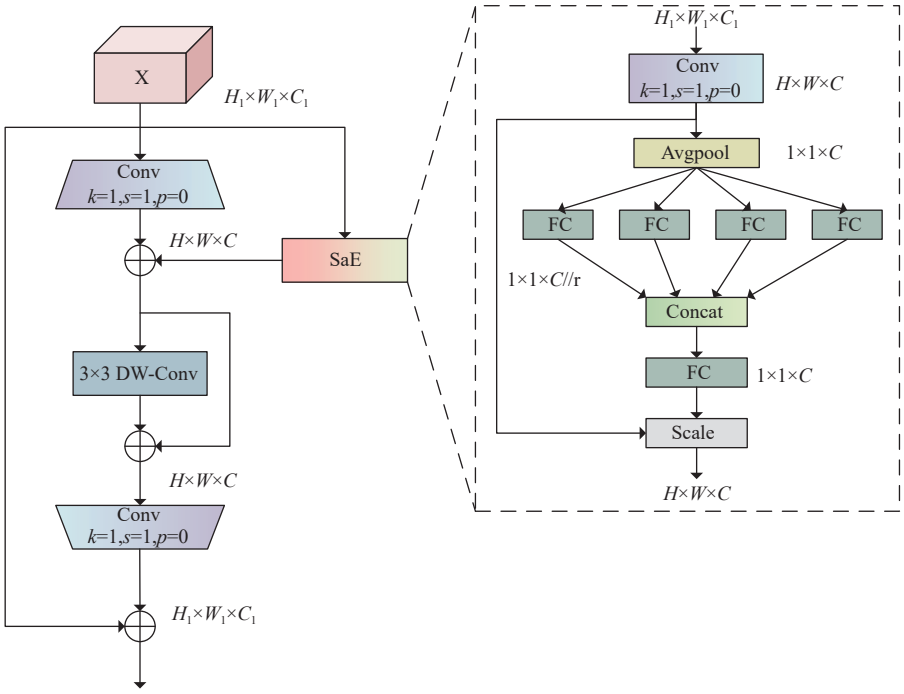
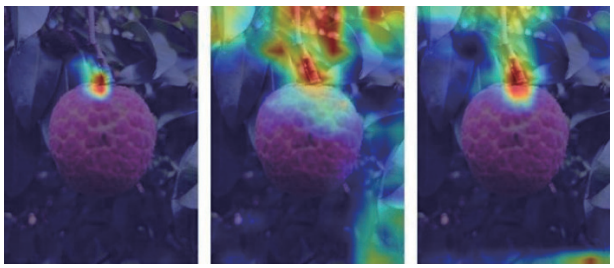


Figure 5 iSaE Block



a. IRB b. SaE c. iSaE

Figure 6 Visual heat maps of picking point attention of different modules

3.3 GELAN Block

Existing methods often suffer from information bottleneck issues during feature extraction across layers, resulting in the loss of crucial information. Therefore, there is a need to design a structure capable of capturing sufficient information. Wang et al.[12] proposed a Generalized Efficient Layer Aggregation Network (GELAN) by combining CSPNet[13] and ELAN[14].

The main concept of CSPNet is to segment gradient flows, allowing gradient flow information to propagate through different paths. The primary objective of this design is to reduce model computation while achieving richer gradient combinations, enabling the model to be deployed on mobile devices without sacrificing performance. The implementation details of CSPNet are as follows:

$$y = C(x_1, T(B(x_2))) \tag{2}$$

where, the input features x are split along the channels into two parts, represented as $[x_1, x_2]$. T is the transition function for inter-stage gradient flow, C is the function for merging two parts, and B is the function of the bottleneck module. This is achieved by dividing the feature map of the input layer into two parts and then merging them through a cross-stage hierarchical structure. However, as the number of stacked modules in CSPNet increases, the performance of the model tends to decrease because adding each module only increases the longest path for gradient flow propagation. To address this issue, ELAN was designed. This architecture is based on the design of the network architecture according to the gradient propagation path. The main idea behind ELAN is to increase the shortest gradient path of the model for faster convergence, which helps to reduce feature redundancy and enhance feature representation capability. GELAN is a new architecture derived from the capabilities of ELAN, which can employ any computational block. In this study, the BottleNeck in C2f is applied to GELAN, and the architecture of GELAN is illustrated in Figure 7.

3.4 RFAPoseHead

For the task of identifying and locating litchi fruits and key points, the shapes and distributions of targets in images can vary. In

convolution operations, convolutional kernels utilize the same weights to extract features within different receptive fields, ignoring differential information from various locations. Additionally, the spatial attention mechanism cannot fully address the parameter sharing issue posed by large convolutional kernels [Equation (3)]. To address these challenges, Zhang et al.[15] proposed a Receptive Field Attention (RFA) approach. The core idea is to integrate spatial attention mechanisms with convolution operation.

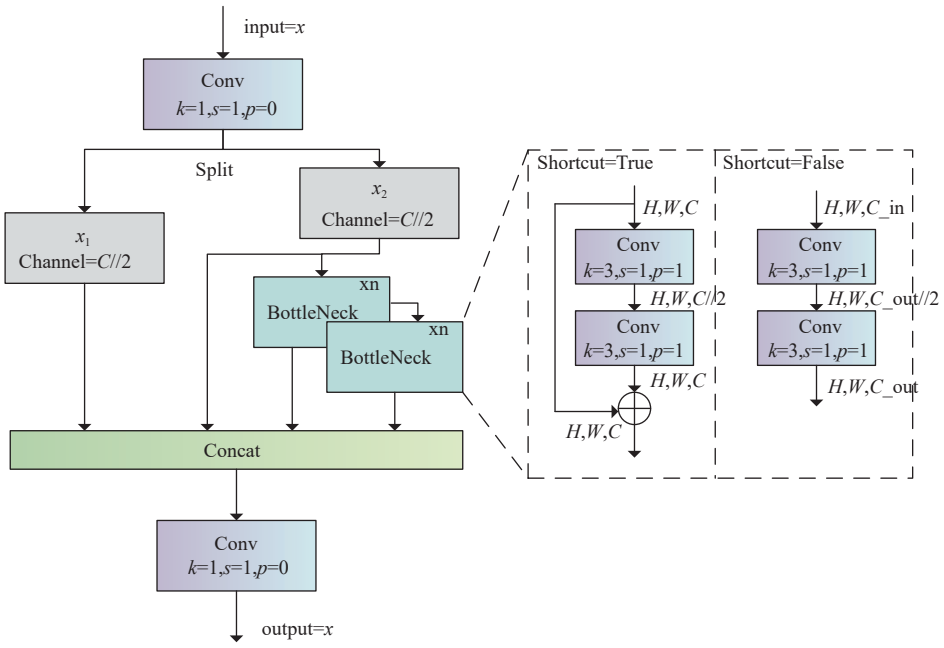


Figure 7 GELAN Block

$$F_N = X_{N1} \times A_{11} \times K_1 + ... + X_{19} \times A_{19} \times K_9 \quad (3)$$

where, 'K' represents the parameter values of the 1×1 convolutional kernel, and 'A' represents the values at different positions of the attention map. When the convolutional kernel extracts features from each receptive field, each sliding window shares the weights of the spatial attention map.

RFAConv (Figure 8) divides the input features into two pathways. One pathway utilizes grouped convolutions of corresponding sizes to dynamically generate feature information based on the receptive field. The other pathway aggregates global information for each receptive field feature using Avgpool and interacts with the feature information using 1×1 convolution. Finally, the feature information from the two pathways is combined to generate spatial feature information at different positions within the receptive field. The spatial feature information of the receptive field is dynamically generated based on the size of the convolutional kernel, thus addressing the parameter-sharing issue of convolutional kernels. In this study, RFAConv is incorporated into the detection head of YOLOv8n-pose (Figure 9), replacing standard convolution. RFAPoseHead enables the model to adapt to changes in the natural environment, thereby enhancing the robustness of the model.

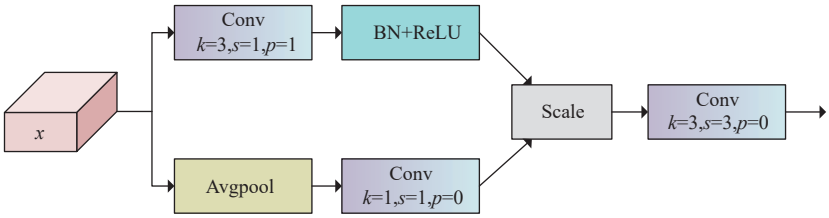


Figure 8 RFAConv structure diagram

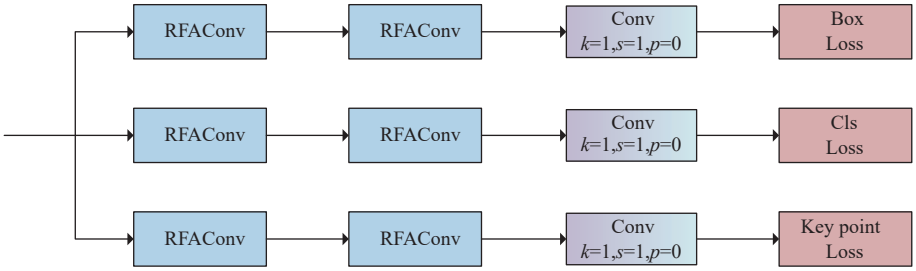


Figure 9 RFAPoseHead structure diagram

4 Experiments

4.1 Training details

The network training was based on the Ubuntu system and the PyTorch framework. The main hardware configuration included an i9-10900k CPU, GeForce RTX 3090 GPU, CUDA 12.3, and Python 3.8. The training environment for different algorithms was the same, and the training parameters were set as follows: the batch size

of the model was set to 32, the maximum number of iterations was 300 epochs, and training stopped if the model performance did not improve in 50 consecutive epochs. The input image size was set to 640×640 pixels, the initial learning rate was 0.01, the decay rate was 0.2, and the weight decay coefficient was 0.0005.

4.2 Evaluation metrics

4.2.1 Evaluation metrics of key point detection

The key point detection metric was inspired by object detection. Precision and Recall were calculated using the quantities of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). The correct detection of harvesting points (TP) should meet the following criteria: 1) Litchi and picking point must be manually labeled, meaning that litchi in the foreground should be medium- or large-sized, and either all or part of the litchi stem should be visible. Litchi that is too small or completely obscured is not considered for manual annotation or automatic detection. 2) The detected picking point is on the stem of the litchi fruit. In the above two cases, incorrect picking point detection (FP) may occur (for example, when the picking point falls on the fruit or non-fruit main stem), and there may also be cases of missed detection (FN). The above criteria can be evaluated using OKS (Object Key point Similarity) as the evaluation metric for key point detection algorithms[16]. The equation for OKS is as follows:

$$\mathrm{OKS}=\frac{\sum_i \exp(-d_{\mathrm{pi}}^2/2S^2\sigma_i^2)\delta}{\sum_i \delta},\ \delta=\begin{cases}1, & (V_{\mathrm{pi}}>0)\\ 0, & (V_{\mathrm{pi}}<0)\end{cases} \tag{4}$$

The evaluation metrics for key point detection include precision (P_{kp}), recall (R_{kp}), average precision ($\mathrm{AP}_{\mathrm{kp}}$), and mean average precision ($\mathrm{mAP}_{\mathrm{kp}}$). Their formulas are as follows:

$$P_{\mathrm{kp}}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{5}$$

$$R_{\mathrm{kp}}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{6}$$

$$\mathrm{AP}_{\mathrm{kp}}=\frac{\sum_n\sum_i \beta}{\sum_n\sum_i 1},\ \beta=\begin{cases}\mathrm{OKS}_i, & (\mathrm{OKS}_i>T)\\ 0, & (\mathrm{OKS}_i\le T)\end{cases} \tag{7}$$

$$\mathrm{mAP}_{\mathrm{kp}}=\frac{\sum_i^C \mathrm{AP}_{\mathrm{kp}}}{C} \tag{8}$$

where, d_{pi} represents the Euclidean distance between the i^{th} detected key point and the corresponding key point in the target; S is the scale factor of the point; V_{pi} denotes the visibility of the point, where "0" indicates unannotated, "1" denotes annotated points obscured, and "2" indicates annotated points visible; σ_i represents the normalization factor of the i^{th} key point; β is the visibility indicator for each key point; T is the OKS threshold; and C is the number of key point categories.

4.2.2 Evaluation metrics of key point position error

In the experiment evaluating the prediction of picking point location, the evaluation metric was the pixel Euclidean distance error between the predicted point (Pre) and the ground truth (GT). Assuming the pixel coordinates of the predicted picking point are (x, y) and the pixel coordinates of the ground truth picking point are (x_1, y_1), and the input image resolution is W×H, the distance (d_x) in the X-axis direction between the predicted point and the ground truth point can be calculated using Equation (9), and the distance (d_y) in the Y-axis direction can be calculated using Equation (10). The pixel Euclidean distance (E) between two points can be calculated using Equation (11).

$$d_x=W|x_1-x| \tag{9}$$

$$d_y=H|y_1-y| \tag{10}$$

$$E=\sqrt{d_x^2+d_y^2} \tag{11}$$

4.3 Experiments with different key point strategies

To select the most suitable key point skeleton for litchi picking, the average distance error between the predicted picking point and the ground truth was used as the evaluation metric. Error analysis was performed using the coordinates of 100 picking points detected by the YOLOv8n-pose model under different key point strategies.

The vertical and horizontal lines of different colors in Figure 10 visualize the average distance errors between all predicted points and ground truth points in the X-axis and Y-axis directions under different strategies of the YOLOv8n-pose model. It visually illustrates the performance gap between YOLOv8n-pose-5p and the other strategies in terms of localization accuracy. Therefore, selecting the 5P key point strategy as the key point skeleton for litchi picking was appropriate, and all subsequent experiments on picking point detection were based on the 5P key point skeleton.

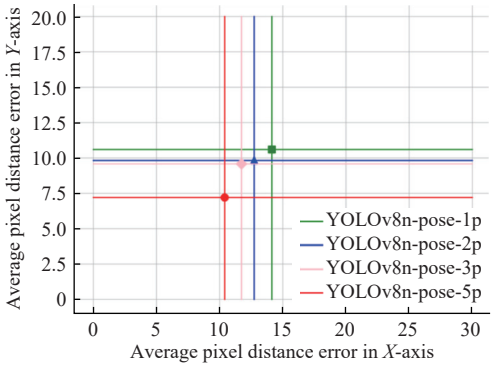


Figure 10 Comparison of average pixel error of different strategies

4.4 Ablation experiment

The YOLOv8-iGR model proposed in this study is based on the YOLOv8n-pose with three improvements: U: integrating the iSaE module into the feature extraction network; V: replacing the C2f with the GELAN; and W: replacing the original detection head with the RFAPoseHead. To validate the effectiveness of the integrated modules, ablation experiments were conducted. The results are listed in Table 1.

Table 1 Comparison results of different models for ablation experiments

U	V	W	P_{box}/%	$\mathrm{AP}_{\mathrm{box}}$/%	P_{kp}/%	$\mathrm{mAP}_{\mathrm{kp}}$/%	GFLOPs/%	FPS
×	×	×	87.2	93.1	87.4	90.3	8.4	99.3
√	×	×	91.0	94.1	90.2	94.4	8.5	93.1
×	√	×	91.7	95.5	91.7	94.6	7.1	93.4
×	×	√	87.4	94.6	88.9	94.2	8.8	91.7
√	√	√	92.0	95.7	92.3	95.6	7.5	90.9

Integrating the iSaE module designed in this study into the base model resulted in improvements of 4.36% and 3.20% in P_{box} and

P_{kp}, respectively. This is because the inclusion of the iSaE module enhanced the network's ability to capture feature information while suppressing irrelevant interference. Replacing the C2f module in the base model with GELAN led to a decrease of 15.48% in GFLOPs, accompanied by improvements of 2.58% and 4.76% in AP_{box} and mAP_{kp}, demonstrating the effective reduction in model complexity achieved by the GELAN module. Substituting the detection head with RFAPoseHead resulted in improvements of 1.61% and 4.32% in AP_{box} and mAP_{kp}. The combination of iSaE, GELAN, and RFAPoseHead enhanced the model's detection performance for litchi fruits and picking points, reducing GFLOPs by 10.71% while increasing P_{box}, AP_{box}, P_{kp}, and mAP_{kp} by 5.51%, 2.79%, 5.61%, and 5.87%, respectively.

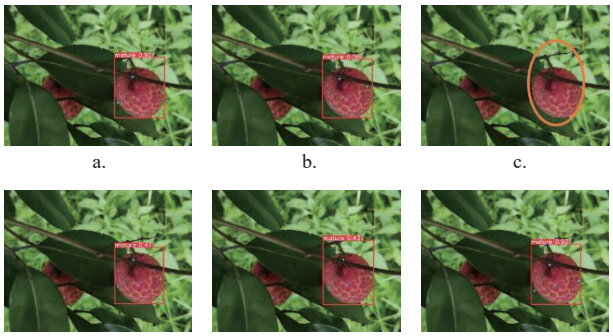
4.5 Comparison experiment

To demonstrate the comprehensive performance of the proposed YOLOv8-iGR algorithm in litchi fruit detection and key point detection, comparisons were made with several mainstream object detection and key point detection algorithms. Due to the inability of the RT-DETR model to directly identify key points, the detection head was replaced with the detection head of YOLOv8n-pose in this experiment. The comparison results for different algorithms are presented in Table 2. Except for the RT-DETR-pose, all models are lightweight detection models. YOLOv8n-pose achieved the highest detection speed of 109.3 fps, demonstrating relatively good detection performance, hence, YOLOv8n-pose was chosen as the baseline model. Compared to YOLOv3n-pose, YOLOv5n-pose, YOLOv6n-pose, and RT-DETR-pose, YOLOv8-iGR showed the highest AP_{box} in object detection, which was higher by 3.0%, 2.5%, 1.17%, and 19%, respectively. In terms of key point detection, the mAP_{kp} of YOLOv8-iGR was higher by 3.24%, 4.36%, 1.81%, and 3.24%, respectively. Although the detection speed of YOLOv8-iGR decreased by 8.5% compared to YOLOv8n-pose, the precision of object detection and key point detection increased by 5.5% and 5.6%, respectively. The decrease in the detection speed of YOLOv8-iGR is attributed to the addition of the attention mechanism, which requires additional computational operations and storage of extra attention weights leading to increased memory consumption. However, considering the robustness provided by the iSaE module and the improvement in detection performance, this decrease is acceptable. A detection speed of 90.9 fps still meets the requirements for real-time detection.

Table 2 Comparison of detection performance among different network models

Model	P_{box}/%	R_{box}/%	AP_{box}/%	P_{kp}/%	P_{kp}/%	mAP_{kp}/%	GFLOPs/G	FPS
YOLOv8n-pose	87.2	91.8	93.1	87.4	91.4	90.3	8.4	99.3
YOLOv5n-pose	86.7	89.8	93.3	86.4	90.4	91.6	7.3	93.3
YOLOv3n-pose	88.9	89.7	92.9	87.9	87.5	92.6	11.4	98.1
YOLOv6n-pose	87.5	90.3	94.6	86.8	89.5	93.9	12.0	93.6
RT-DETR-pose	86.6	91.1	93.9	85.9	90.6	92.6	139.4	36.3
YOLOv8-iGR	92.0	93.9	95.7	92.3	93.5	95.6	7.5	90.9

Figures 11 and 12 visualize the detection results of litchi in scenes with occlusion and complex scenes. In these figures, the litchi missed by the model was marked with orange ellipses, while the litchi incorrectly detected by the model was marked with blue rectangles. In cases of occlusion, certain features of the litchi were obstructed, affecting the detection performance of the model. Figure 11 illustrates the performance of the model in a scene with branches occlusion, where YOLOv5n-pose missed the detection of litchi fruits and key points, while YOLOv6n-pose, YOLOv3n-pose, and RT-DETR-pose exhibited low confidence levels in their predictions. Figure 12 depicts a dense litchi orchard scene, where both YOLOv8n-pose and RT-DETR-pose encountered detection errors, and YOLOv3n-pose missed litchi farther from the camera. The proposed YOLOv8-iGR demonstrates robust performance in both simple and complex scenes. Even when litchi is occluded by leaves and branches, YOLOv8-iGR effectively completes the task of litchi detection and key point prediction.



Note: (a) YOLOv8n-pose (base model), (b) YOLOv6n-pose, (c) YOLOv5n-pose, (d) YOLOv3n-pose, (e) RT-DETR-pose, (f) YOLOv8-iGR (ours).

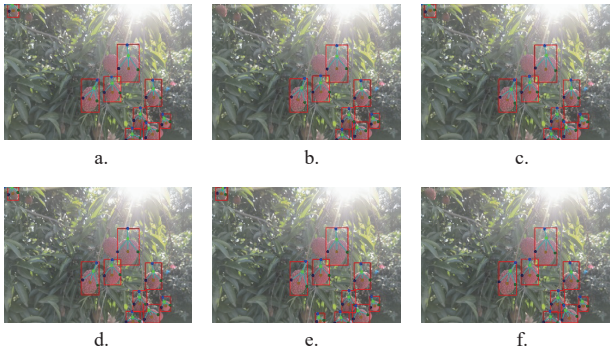
Figure 11 Detection results of branches occlusion

Note: (a) YOLOv8n-pose (base model), (b) YOLOv6n-pose, (c) YOLOv5n-pose, (d) YOLOv3n-pose, (e) RT-DETR-pose, (f) YOLOv8-iGR (ours).

Figure 12 Detection results of dense scene

4.6 Comparison experiment of picking point position error

To validate the localization performance of models in different natural environments, this section conducted experiments in two scenes: Scene 1: intense natural light conditions; Scene 2: low natural light conditions. Figures 13 and 14 illustrate the picking



Note: (a) YOLOv8n-pose (base model), (b) YOLOv6n-pose, (c) YOLOv5n-pose, (d) YOLOv3n-pose, (e) RT-DETR-pose, (f) YOLOv8-iGR (this study).

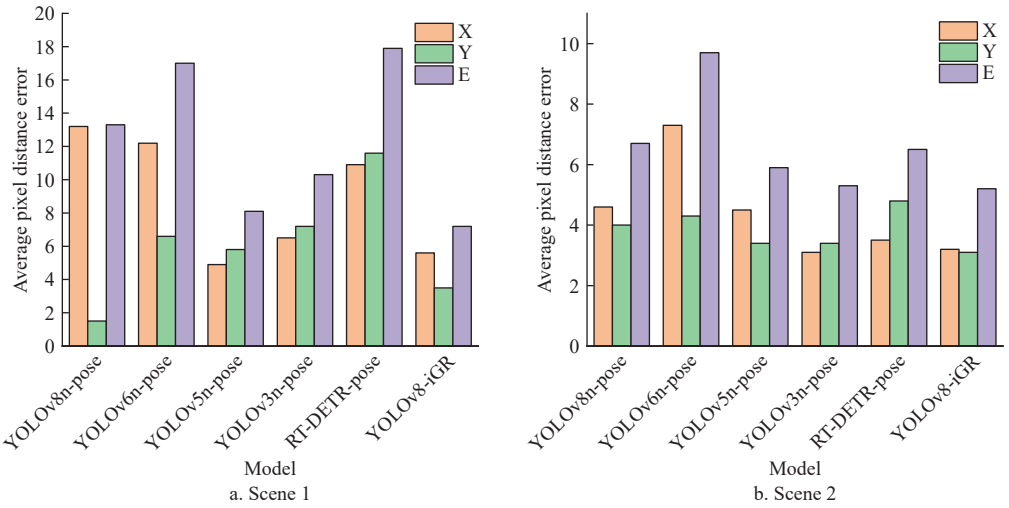
Figure 13 Detection results of picking point in Scene 2

Note: (a) YOLOv8n-pose (base model), (b) YOLOv6n-pose, (c) YOLOv5n-pose, (d) YOLOv3n-pose, (e) RT-DETR-pose, (f) YOLOv8-iGR (ours).

Figure 14 Detection results of picking point in Scene 3

point detection results. Each scene included 100 picking points detectable by the models, with the average pixel distance error between predicted and ground truth points used as the evaluation metric. Figure 15 presents the analysis of the average pixel distance error in the *X*-axis, *Y*-axis, and Euclidean directions for the 100 predicted picking points across different models and scenes. Other key points served only as auxiliaries for localization. To visually assess the detection performance of litchi key points, all key points were visualized and connected to form a litchi skeleton.

Figure 13 presents the recognition results of different models in Scene 1. Influenced by strong lighting, the image features of litchis became blurred. YOLOv8-iGR achieved picking point prediction with an average pixel error of 7.20. In Figure 14, under low light conditions, YOLOv8-iGR completed picking point prediction with an average error of less than 6 pixels. It is worth noting that other models exhibit instances of missed detection (marked with orange ellipses). In fact, assuming the operating range of the end effector is 60 mm, as demonstrated by Xiong et al.[17] when the camera is positioned 30-100 cm away from the target, the pixel error within a range of 60 mm is approximately 35-80 pixels[18]. Therefore, in the above scenes, YOLOv8-iGR's ability to locate picking points can meet the harvesting requirements.



Note: X: Average pixel distance error in the *X*-axis direction. Y: Average pixel distance error in the *Y*-axis direction. E: Average pixel Euclidean distance error.

Figure 15 Average pixel distance error predicted by different models for picking points

4.7 Field picking test

To validate the accuracy of this study's method, this study conducted field picking experiments in the litchi orchard from June 23 to July 3, 2024. Figure 16 illustrates the actual integration of the experimental environment for the litchi-picking robot. The picking times were from 8:00 am to 11:00 am and from 3:00 pm to 5:00 pm. The experiments were based on the Xarm 6-axis robotic arm platform, with a Realsense D435i camera mounted on the arm in an eye-in-hand configuration. After training the proposed YOLOv8-iGR model, it was deployed on the robotic arm for detection. In each experiment, to simulate real-world harvesting scenarios where litchi clusters vary in density and spatial distribution, 12 fruits were randomly placed from different tree heights (1.3-1.5 m) and arranged in randomized layouts (e.g., sparse, dense, occluded). This design ensures that the model is tested under diverse conditions reflective of actual orchard environments, conducting a total of 10 experiments with 120 litchi in total. The success rate of picking was 95.0%. The main reason for picking failures was the litchi's low position, which caused depth value deviations in the depth camera, which in turn led to coordinate analysis errors and resulted in incorrect movement paths for the robotic arm.

Figure 16 Field picking experiment

5 Conclusions

YOLOv8-iGR is a novel method for litchi-picking point identification using a key point detection model. Initially designed for human pose estimation, key point detection algorithms are expanded in this study to a new application scenario. By integrating the iSaE, GELAN, and RFAPoseHead architectures into YOLOv8n-pose, an enhanced detector capable of simultaneous recognition of litchi fruits and picking points is developed. Additionally, the Object Key point Similarity (OKS) metric is employed to evaluate key point detection performance, while pixel Euclidean distance is utilized to assess the prediction error of picking point position. Experimental results demonstrate that YOLOv8-iGR achieves a precision improvement in litchi picking point detection from 87.4% to 92.3%, with a reduction in computational complexity from 8.4 G to 7.5 G. The average pixel Euclidean distance error between predicted and ground true picking point positions is within 8 pixels. Compared to various mainstream detection algorithms, YOLOv8-iGR exhibits significant advantages in detection performance under complex and dynamic environmental conditions. These results underscore the potential of the proposed YOLOv8-iGR model to support the visual systems of picking robots.

Acknowledgements

This research was supported by Natural Science Foundation of Guangdong Province (Grant No. 2025A1515011771), Guangzhou Science and Technology Plan Project (Grant No. 2024E04J1242, 2023B01J0046), Guangdong Provincial Department of Science and Technology (Grant No. 2023A0505050130), Key Projects of Guangzhou Science and Technology Program (Grant No. 2024B03J1357), and Natural Science Foundation of China (Grant No. 61863011, 32071912).

[References]

[1] Xie J X, Peng J J, Wang J X, Chen B H, Jing T W, Sun D Z, et al. Litchi detection in a complex natural environment using the YOLOv5-litchi model. Agronomy, 2022; 12(12): 3054.

[2] Qi X K, Dong J S, Lan Y B, Zhu H. Method for identifying litchi picking position based on YOLOv5 and PSPNet. Remote Sensing, 2022; 14(9): 2004.

[3] Zhang G M, Cao H, Hu K W, Pan Y Q, Deng Y Q, Wang H J, et al. Accurate cutting-point estimation for robotic lychee harvesting through geometry-aware learning. arXiv: 2404.00364, 2024; In press. doi: 10.48550/arXiv.2404.00364.

[4] Tang Y C, Qiu J J, Zhang Y Q, Wu D X, Cao Y H, Zhao K X, et al. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review. Precision Agriculture, 2023; 24(4): 1183–1219.

[5] Peng H X, Zhong J R, Liu H, Li J, Yao M W, Zhang X. ResDense-focal-DeepLabV3+ enabled litchi branch semantic segmentation for robotic harvesting. Computers and Electronics in Agriculture, 2023; 206: 107691.

[6] Zheng C, Chen P F, Pang J, Yang X F, Chen C X, Tu S Q, et al. A mango picking vision algorithm on instance segmentation and key point detection from RGB images in an open orchard. Biosystems Engineering, 2021; 206: 32–54.

[7] Du X Q, Meng Z C, Ma Z H, Lu W W, Cheng H C. Tomato 3D pose detection algorithm based on keypoint detection and point cloud processing. Computers and Electronics in Agriculture, 2023; 212: 108056.

[8] Narayanan M. SENetV2: Aggregated dense layer for channelwise and global representations. arXiv: 2311.10807. 2023; In Press. doi: 10.48550/arXiv.2311.10807.

[9] Hu J, Shen L, Albanie S, Sun G, Wu E H. Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020; 42(8): 2021–2023.

[10] Zhang J N, Li X T, Li J, Liu L, Xue Z C, Zhang B S. Rethinking mobile block for efficient attention-based models. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France: IEEE, 2023; pp.1389–1400.

[11] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA: IEEE, 2018; pp.4510–4520.

[12] Wang C-Y, Yeh I-H, Liao H-Y M. YOLOv9: Learning what you want to learn using programmable gradient information. arXiv: 2402.13616, 2024; In press. doi: 10.48550/arXiv.2402.13616.

[13] Wang C-Y, Liao H-Y M, Wu Y-H, Chen P-Y, Hsieh J-W, Yeh I-H. CSPNet: A new backbone that can enhance learning capability of CNN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA: IEEE, 2020; 1571–1580.

[14] Wang C-Y, Liao H-Y M, Yeh I-H. Designing network design strategies through gradient path analysis. arXiv: 2211.04800, 2022; In press. doi: 10.48550/arXiv.2211.04800.

[15] Zhang X, Liu C, Yang D G, Song T T, Ye Y C, Li K, et al. Rfaconv: Innovating spatital attention and standard convolutional operation. arXiv: 2304.03198, 2023; In press. doi: 10.48550/arXiv.2304.03198.

[16] Maji D, Nagori S, Mathew M, Poddar D. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA: IEEE, 2022; pp.2636–2645. doi: 10.1109/CVPRW56347.2022.00297.

[17] Xiong J T, He Z L, Lin R, Liu Z, Bu R B, Yang Z G, et al. Visual positioning technology of picking robots for dynamic litchi clusters with disturbance. Computers and Electronics in Agriculture, 2018; 151: 226–237.

[18] Zhuang J W, Hou C J, Tang Y, He Y, Guo Q W, Zhong Z Y, et al. Computer vision-based localisation of picking points for automatic litchi harvesting applications towards natural scenarios. Biosystems Engineering, 2019; 187: 1–20.