Optimization hyperparameters of YOLO using the coati algorithm for detecting and counting unripe citrus fruits

Shilei Lyu[1,2,3], Yicong Chen[1,2], Peng Gao[1*], Xueya Liu[1,2], Zhen Li[3*], Jieyu Chen[1], Zijie Li[1], Jiahong Chen[1]

(1. *College of Artificial Intelligence and Low-Altitude Technology, South China Agricultural University, Guangzhou 510642, China*;
2. *Pazhou Lab, Guangzhou 510335, China*;
3. *Division of Citrus Machinery, China Agriculture Research System, Guangzhou 510642, China*)

Abstract: Real-time inspection of citrus orchards can provide reliable support for production management processes, such as fruit thinning and setting, sunburn prevention and control, and yield prediction. To address problems such as the difficulty of identifying unripe citrus fruits in a natural environment and the high labor cost of counting, this study proposes an unripe citrus fruit detection, tracking, and counting algorithm based on YOLOv7-tiny-Convolutional Block Attention Module-Wise IoU (YOLO-CW) combined with StrongSORT. First, to improve the feature extraction ability of the model for unripe citrus fruits in small targets and complex backgrounds, this study employed YOLOv7-tiny as the baseline model. The Convolutional Block Attention Module (CBAM) was integrated into the ELAN-T module ahead of the P3 detection head in YOLOv7-tiny and redesigned as the ELAN-TC module. Second, the loss function CIoU in YOLOv7-tiny was replaced with Wise IoUv3 to improve the model's ability to detect overlapping unripe citrus fruits in complex backgrounds, thereby reducing the impact of harmful gradients produced by low-quality unripe citrus fruit samples. Third, to address the challenge of hyperparameter optimization that relies on experience and manpower, this study employed the Coati Optimization Algorithm (COA) to optimize the hyperparameters of the model, further enhancing detection accuracy. Finally, the YOLO-CW model was ported to an edge computing platform, enabling real-time data collection of unripe citrus fruits using multiple wireless IP cameras. The collected data were tracked and counted using StrongSORT. The experimental results demonstrate that the Precision and mAP@0.5 of the YOLO-CW model optimized by COA were 93.80% and 96.62%, respectively. Compared to the baseline model YOLOv7-tiny, the YOLO-CW model exhibited improvements of 1.68% in Precision and 0.71% in mAP@0.5. The YOLO-CW model's Computation and Parameters were 13.2 GFlops and 6.01×10^6, respectively. Compared to the baseline model YOLOv7-tiny is reduced by 5.03% and 3.53%. An edge computing platform and a dual-channel wireless IP camera were used for image acquisition and processing to evaluate the performance of the proposed YOLO-CW model in tracking and counting unripe citrus fruits. The results indicate that the average frame rate was 23 FPS and the overall system power consumption was 17.11 W. Compared to the baseline model YOLOv7-tiny has 10% increase in average frame rate and 7.62% reduction in power consumption. These findings indicate that the proposed model can track and count unripe citrus fruits in real time in natural environments, providing valuable theoretical support for citrus yield assessment.

Keywords: unripe citrus fruit, target detection, YOLO, hyperparameter optimization, multi-objective tracking
DOI: 10.25165/j.ijabe.20261904.9599

Citation: Lyu S, Chen Y C, Gao P, Liu X Y, Li Z, Chen J Y, et al. Optimization hyperparameters of YOLO using the coati algorithm for detecting and counting unripe citrus fruits. Int J Agric & Biol Eng, 2026; 19(4): 268–281.

1 Introduction

Mechanized management of orchards requires information collection and monitoring throughout the fruit growth cycle[1,2]. The application of computer vision techniques for fruit detection has become a prominent research area[3-5]. These techniques are used to identify, track, and count unripe fruits, provide data during the growing season, and predict yields at harvest. Qiu et al.[6] proposed an improved SM-YOLOv4 lightweight model that employed binocular stereo vision to pre-locate grape clusters, thereby enabling target recognition of unripe and ripe grapes. Li et al.[7] introduced the MHSA-YOLOv8 grading model, which was used for the identification, grading, and counting of unripe and ripe tomatoes. Yang et al.[8] combined the YOLOv8s model with the LS-Swin Transformer, achieving a target detection accuracy of 94.4% for both unripe and ripe strawberries. Chen et al.[9] introduced a cross-cutting attention mechanism with distance and intersection ratio-based non-maximal transplantation to improve the YOLOv7 model, achieving the target recognition of immature and mature oil tea fruits. Although these approaches are effective for fruits that differ

Received date: 2024-12-10 **Accepted date:** 2026-04-17
Biographies: Shilei Lyu, Professor, research interest: agricultural IoT and artificial intelligence, Email: lvshilei@scau.edu.cn; **Yicong Chen**, ME, research interest: new-generation electronic information technology, Email: yicongchen@stu.scau.edu.cn; **Xueya Liu**, ME, research interest: new-generation electronic information technology, Email: liuxueya@stu.scau.edu.cn; **Jieyu Chen**, ME, research interest: new-generation electronic information technology, Email: chenjieyu@stu.scau.edu.cn; **Zijie Li**, ME, research interest: artificial intelligence, Email: 746315841@stu.scau.edu.cn; **Jiahong Chen**, ME, research interest: new-generation electronic information technology, Email: 20233169004@stu.scau.edu.cn.

***Corresponding author: Peng Gao**, Lecturer, research interest: digital agriculture. College of Artificial Intelligence and Low-Altitude Technology, South China Agricultural University, Guangzhou 510642, China. Tel: +86-18819427306, Email: gaopeng.peng@scau.edu.cn; **Zhen Li**, Professor, research interest: smart agriculture. Division of Citrus Machinery, China Agriculture Research System, Guangzhou 510642, China. Tel: +86-13610189829, Email: lizhen@scau.edu.cn.

significantly from the background, greenish immature fruits pose a challenge because they often resemble the background environment. In addition, foliage shade and fruit overlap further complicate target identification. Despite these challenges, various solutions have been proposed for detecting immature green fruits. Wang et al.[10] proposed an innovative fruit segmentation method, SE-COTR, which achieved an average accuracy of 61.6% for the recognition and segmentation of green apples in complex orchards. Ren et al.[11] improved YOLOv8s to develop the YOLO-GEW model, which achieved an average recognition accuracy of 88.83% for green yucca pears. Zhang et al.[12] employed the RepVGG structure with Focal-EIoU to improve the YOLOv5s model to the YOLOMS model, achieving a recognition accuracy of 92.19% for mangoes. Yue et al.[13] designed the ME convolution module with Inner-CIoU to refine the YOLOv8n model into the unripe citrus fruit recognition model YOLOv8-MEIN, achieving a Recall of 91.7%. These studies focused on identifying static images of immature fruits. However, for orchard yield measurement in a natural environment, dynamically collecting images of immature fruits in real time is essential, incorporating real-time tracking and counting.

In recent years, studies have increasingly combined target detection models with target tracking to follow, count, and measure fruit yield. Zhang et al.[14] integrated an improved OrangeYolo model with the fruit tracking system OrangeSort, achieving an average detection accuracy of 95.7%. Tu et al.[15] combined the improved YOLOv5s-little model with DeepSORT for counting, obtaining an average identification accuracy of 95.1% for passion fruit. Shi et al.[16] optimized the YOLOv3-tiny model and ported it to the ARM platform, achieving an F1 score of 94.4% for the identification of green mangoes. Huang et al.[17] reported an improved YOLOv5 unripe citrus fruit model to an edge computing platform, achieving an accuracy of 93.32% and a processing speed of 180 ms. Lyu et al.[18] deployed an improved YOLOv5-CS unripe citrus fruit detection model on the NVIDIA Jetson Xavier NX, achieving an inference speed of 37 ms and counting frame rate of 28 fps. Liu et al.[19] implemented the improved YOLOv5x model on a picking robot, achieving a real-time detection frame rate of 47 FPS.

Modifying the model network structure can improve model alignment with the task environment, which enhances detection performance of the model[20,21]. However, previous studies have employed default hyperparameters during suboptimal task training. Hyperparameter optimization effectively improves detection performance and model adaptability to the task environment. Manual hyperparameter adjustment requires significant expertise and resources[22,23]. Optimization algorithms provide an alternative that allows intelligent self-tuning during training. Yu et al.[24] applied a Genetic Algorithm (GA) to optimize the YOLOv7 hyperparameters. The foreign object identification on the transmission lines achieved an average accuracy of 92.2%. Lyu et al.[25] employed the Black Widow Optimization Algorithm (BWOA) to optimize hyperparameters of the citrus louse detection model YOLO-SCL, increasing mAP@0.5 to 97.18%. Stefano et al.[26] employed a Genetic Algorithm (GA) to optimize the hyperparameters of Hypertuned-YOLO, achieving an optimized F1 score of 0.867 and mAP@0.5 of 0.922. The hyperparameters significantly affected model training, and the optimal hyperparameters further enhanced detection accuracy.

Machine-vision technology has promise for fruit recognition and detection. However, current research has primarily focused on static fruits, with limited studies on real-time recognition, tracking, and counting of moving fruits. Citrus orchards are located in hilly areas and complex environments; thus, employing portable tracking and counting equipment can address these environmental challenges and enable real-time tracking and counting of unripe citrus fruits. Hyperparameter optimization of models requires significant expertise and resources and limited research on unripe green fruit models. To address the challenge of real-time detection, tracking, and counting of unripe citrus fruits in natural environments for subsequent yield measurements, this study proposes a lightweight target detection model for YOLO-CW. In addition, the Long-nosed Raccoon intelligent optimization algorithm was employed to optimize the hyperparameters of the YOLO-CW model.

(1) Feature extraction capability enhancement: In this study, the ELAN-T module ahead of the P3 detection head was incorporated into the CBAM to create the ELAN-TC module. This integration improves the feature extraction capability of unripe citrus fruits with complex backgrounds and reduces the Computation and Parameters.

(2) Loss function optimization: In this study, the loss function WIoUv3 was used to reduce the gradient gain of low-quality samples. This approach mitigates the production of harmful gradients and further improves the model's ability to detect unripe citrus fruits with complex backgrounds.

(3) Hyperparametric optimization: This study optimizes the YOLO-CW model parameters using the Coati Optimization Algorithm (COA) to enhance the detection performance. The proposed method addresses the challenges associated with manual hyperparameter tuning, which is reliant on expertise and significant resources.

(4) Model deployment assessment study: The optimized model is then ported to an edge computing platform. Multiple wireless IP cameras collect unripe citrus fruit data in real time. The StrongSORT target tracking algorithm processes these data to detect, track, and count unripe citrus fruit.

2 Unripe citrus fruit dataset construction

Images of unripe citrus fruits were collected from September 2022 to June 2023 between 9:00 am and 4:00 pm in citrus orchards in South China Agricultural University, Guangzhou; Conghua District, Guangzhou; Deqing County, Zhaoqing, Guangdong Province; Sihui, Guangdong Province; and Yizhang County, Hunan Province. Data were captured using a Panasonic DMC-G7 camera, Honor 20, iPhone 14 ProMax, and Redmi K60 smartphones. The dataset contains over 3000 images of unripe citrus fruits under various conditions: sunny, cloudy, front-lit, back-lit, shaded, clear, and blurry. The dataset samples are shown in Figure 1.

The unripe citrus fruit dataset was augmented using pretzel noise, image inversion, vertical and horizontal mirroring, and additional enhancements. The expanded dataset, which contains 1376 images, was created by combining the original and augmented images. The unripe citrus fruit dataset was labeled using the Labelimg tool. The labeled dataset was divided into 1100 images (training set) and 276 images (validation set) with a training to validation ratio of 8:2.

3 Design of the YOLO-CW detection model

Compared to the Detect head of YOLOv7, the IDetect head of YOLOv7-tiny introduces an implicit knowledge learning mechanism, which improves the overall detection accuracy of the model without increasing the computational burden, and achieves higher detection accuracy when facing small and dense target detection tasks. In addition to the IDetect detection head, thanks to

the ELAN-T module, the YOLOv7-tiny model achieves fast detection speed and high detection accuracy while maintaining low Computational Cost and few Parameters. Compared to the YOLOv7 model, the YOLOv7-tiny model achieves high accuracy while being lightweight and offering high-speed inference. Therefore, this experiment adopts YOLOv7-tiny as the baseline model and proposes YOLOv7-tiny-Convolutional Block Attention Module-Wise IoU (YOLO-CW), a target detection model for unripe citrus. Figure 2 illustrates the architecture of the proposed detection model. First, a Convolutional Block Attention Module (CBAM) was integrated into the ELAN-T module ahead of the P3 detection head to form the ELAN-TC module (Figure 3). This enhancement effectively improves the feature extraction of unripe citrus fruits on small targets with complex backgrounds and reduces the Computation and Parameters. Second, the CIoU loss function was modified to WIoUv3 to mitigate the harmful gradient of low-quality samples generated by the overlap of unripe citrus fruits with foliage occlusion during training. Finally, the Coati Optimization Algorithm (COA) was employed to optimize model hyperparameters to improve detection performance through the strategies of coati hunting, attacking iguanas, and fleeing from predators.

a. Back lighting b. Front lighting c. Less shade d. More shade

Figure 1 Unripe citrus fruit dataset

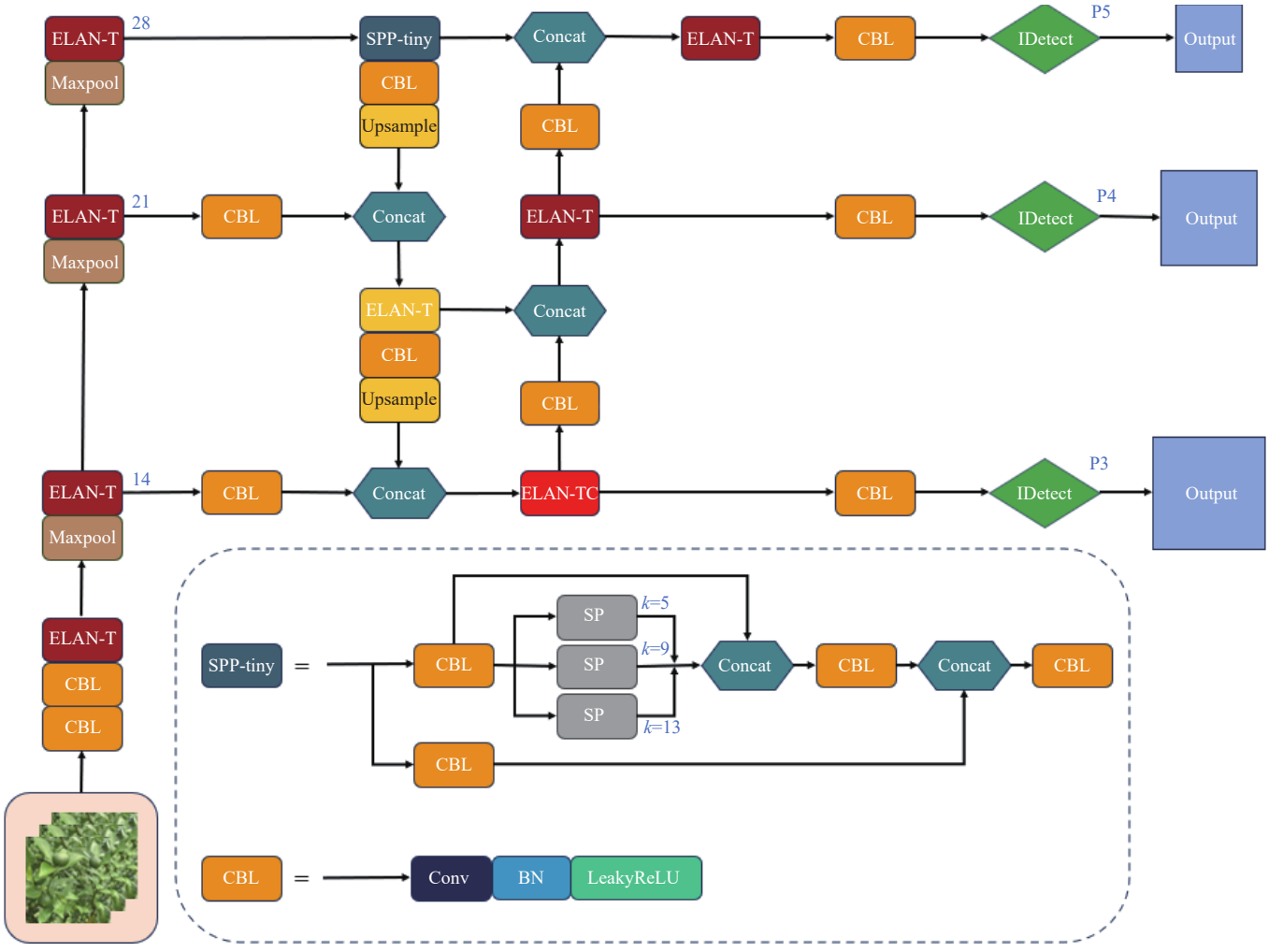

Figure 2 Flowchart of YOLO-CW structure

3.1 ELAN-TC module design

In the ELAN-T module in YOLOv7-tiny[27], the feature learning and robustness of the network are enhanced by controlling the shortest and longest gradient paths (Figure 4). The ELAN-T module

consists of a branch with one 1×1 and two 3×3 convolutional layers in ① and one 1×1 convolutional layer in ②, all of which employ LeakyReLU activation functions. Shallow and deep feature information are fused using two residual joins in ①. Then the results of ① and ② are connected through a 1×1 convolutional layer to form a complete ELAN-T module. The ELAN-T module enables the YOLOv7-tiny model to achieve fast and accurate detection with reduced Computation and Parameters. Consequently, the YOLOv7-tiny model was selected for optimization in this study to enhance its migration and deployment on an edge computing platform to track and count unripe citrus fruits.

Unripe citrus fruit target detection presents significant challenges because of the prevalence of small targets and their similarity to leaves, which hinder effective feature extraction. To address these issues, an ELAN-T module was incorporated ahead of the P3 detection head of the YOLOv7-tiny model. By integrating the Convolutional Block Attention Module (CBAM)[28] into the ELAN-T module, channel and spatial attention mechanisms can be enhanced, thereby improving the feature extraction and overall model performance for recognizing citrus fruits in complex backgrounds and at small scales.

The proposed CBAM comprises two independent modules: the Channel Attention Module (CAM) and Spatial Attention Module (SAM) (Figure 5). The operation of the Channel Attention Module can be mathematically expressed as follows:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (1)$$

where, F is the feature map to the input of the channel attention module, MLP is the two-layer neural network, and M_c is the final output of the channel attention module. The feature map F is initially processed by the channel attention module, where global maximum pooling and global average pooling are applied across the width and height dimensions, respectively. The resulting feature vectors are then input to the MLP. The outputs of the MLP undergo element-wise addition with sigmoid activation to produce the channel attention weight M_c (Figure 6).

Figure 3 Flowchart of the ELAN-T and ELAN-TC modules

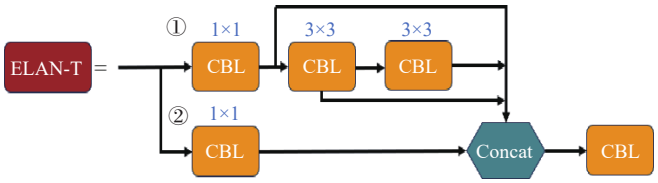

Figure 4 ELAN-T module

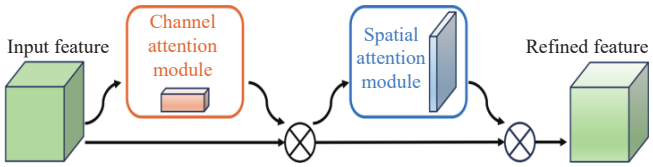

Figure 5 Convolutional Block Attention Module

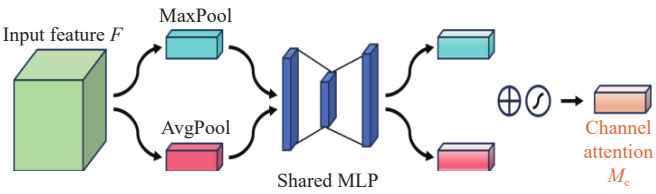

Figure 6 Channel convolution operation

M_c are then multiplied element-wise with the original feature map F, resulting in a refined feature map F' that is subsequently used as input to the spatial attention mechanism. The operation of the spatial attention mechanism is mathematically defined as:

$$M_s(F') = \sigma(f^{7\times7}([MaxPool(F'), AvgPool(F')])) \quad (2)$$

where, F' is the feature map of the input to the spatial attention module, $f^{7\times7}$ is the 7×7 convolution kernel, and M_s is the final output of the spatial attention module. F' performs global maximum and global average pooling of channels, followed by channel splicing operations; M_s is produced through a 7×7 convolution operation for dimensionality reduction with sigmoid activation. Finally, M_s undergoes element-wise multiplication with F to complete the process. During the feature extraction process conducted by the CBAM through channel convolution and spatial convolution, superfluous feature information is eliminated. This elimination subsequently decreases the Computation and Parameters associated with superfluous feature information (as shown in Figure 7).

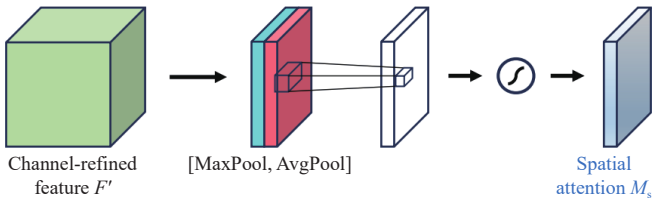

Figure 7 Spatial convolution operation

In this study, an ELAN-T module was designed ahead of the P3 detection head. As shown in Figure 8, a CBAM was initially added after channel and spatial convolution in ①. Subsequently, CBAM was added after performing channel convolution in ②. The addition in ① enhances feature extraction in deeper layers. In ②, the CBAM addresses the limited shallow feature information retained by single channel convolution, which improves small target detection, such as unripe citrus fruit, and handling complex backgrounds. The improved ELAN-T module is referred to as ELAN-TC. The model was trained to achieve optimal detection performance for unripe citrus fruit target detection. Upon integrating the CBAM, both the Computation and Parameters in the ELAN-TC module are diminished in comparison to the ELAN-T module. This enhancement facilitates the model's migration to the edge platform, thereby elevating inference speed and mitigating power consumption on the edge platform.

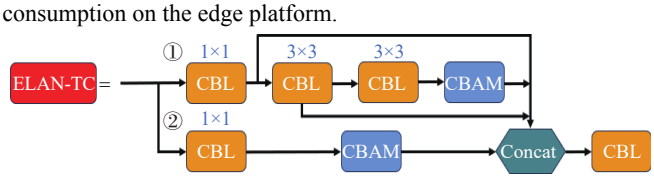

Figure 8 Improved ELAN-TC module

3.2 Loss function improvement

In the labeling process for unripe citrus fruit images, problems such as overlapping unripe citrus fruit and foliage occlusion can result in incomplete labeling. Although the CIoU[29] used in YOLOv7-tiny considers the overlap area, centroid distance, and aspect ratio of bounding box regression, it does not accurately reflect the difference between the width and height and their respective confidence levels. This causes the model to intensify these types of low-quality samples, reducing its generalizability, particularly when predicting overlapping or occluded citrus fruits along with the background as unripe citrus fruits during training. To address this issue, the loss function CIoU was replaced with WIoU (Wise IoU), which mitigates the negative effects of low-quality samples on the training process.

There are three versions of WIoU[30]: WIoUv1, WIoUv2, and WIoUv3. The WIoUv1 constructs an attention-based edge loss. The WIoUv2 and WIoUv3 enhance the WIoUv1 by using focusing coefficients to introduce a focusing mechanism.

WIoUv1 constructs distance attention to form a two-layer attention mechanism as follows:

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU}\mathcal{L}_{IoU} \tag{3}$$

$$\mathcal{R}_{WIoU} = \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{\left(W_g^2+H_g^2\right)}\right) \tag{4}$$

where, x and y represent the positions of the center points of the prediction box; x_{gt} and y_{gt} represent the positions of the center points of the calibration frame; W_g and H_g represent the width and height of the calibration frame, respectively; $\mathcal{L}_{IoU}$ is the degree of overlap between the predicted and real frames in the target detection task. When $\mathcal{L}_{IoU}\in[0,1]$, the $\mathcal{R}_{WIoU}$ of high-quality anchor frames is significantly reduced. This reduction produces high-quality anchor frames with improved overlap with calibration frames, thereby reducing the sensitivity to centroid distances.

WIoUv2 employs a monotonic focusing mechanism as follows:

$$\mathcal{L}_{WIoUv2} = \left(\frac{\mathcal{L}_{IoU}^*}{\overline{\mathcal{L}_{IoU}}}\right)^{\gamma}\mathcal{L}_{WIoUv1},\ \gamma>0 \tag{5}$$

where, $\mathcal{L}_{IoU}^*$ is the monotonic focusing factor. Given that $\mathcal{L}_{IoU}^*$ decreases with decreasing $\mathcal{L}_{IoU}$, $\overline{\mathcal{L}_{IoU}}$ is introduced as a normalization factor. $\overline{\mathcal{L}_{IoU}}$ represents the sliding mean with momentum m, which dynamically updates the gradient gain of $\left(\frac{\mathcal{L}_{IoU}^*}{\overline{\mathcal{L}_{IoU}}}\right)^{\gamma}$ to address the problem of slow convergence during the later stages of training.

WIoUv3 defines outliers to represent the quality of the anchor frame.

$$\beta = \frac{\mathcal{L}_{IoU}^*}{\overline{\mathcal{L}_{IoU}}} \tag{6}$$

$$\mathcal{L}_{WIoUv3} = r\mathcal{L}_{WIoUv1},\ r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{7}$$

$\beta\in[0,+\infty)$ in Equations (6) and (7). A small outlier, also known as a high-quality anchor frame, was assigned a small gradient gain to focus on normal-quality anchor frames. Conversely, a large outlier, also called a low-quality anchor frame, was assigned a smaller gradient gain to effectively prevent harmful gradients generated by low-quality samples. When β is equal to a constant value, the anchor frame receives the maximum gradient gain. Given that the $\overline{\mathcal{L}_{IoU}}$ is dynamically changing, $\mathcal{L}_{WIoUv3}$ continuously performs gradient gain allocation to match the current training optimum.

Problems such as overlapping unripe citrus fruits and foliage shading can result in calibration frames containing a low effective portion of unripe citrus fruits during data labeling. During training, prediction frames may calibrate such samples to the background, which allows low-quality samples to influence the training process continuously. This adversely affects the generalizability and detection ability of the overall model. To address this issue, WIoUv3 was employed. For overlapping unripe citrus fruits and foliage shading, the gradient gain of low-quality samples was continuously adjusted during training. This effectively prevented harmful gradients generated by low-quality samples, thereby allowing the model to focus on both normal-quality and high-quality samples (Figure 9).

3.3 Hyperparameter optimization of the YOLO-CW detection model

The hyperparameter values directly affect model training, and their optimization varies for different tasks. This study uses the Coati Optimization Algorithm (COA)[31] to optimize the hyperparameters of the YOLO-CW model to further improve detection performance. This approach addresses the problem of adjusting hyperparameters based on experience and significant labor costs.

Inspired by coati's hunting behavior, the COA primarily consists of strategies that reflect coati's hunting and attack tactics on iguanas, as well as escape strategies from predators. In this algorithm, the first half of the coati individuals climb trees to hunt iguanas, while the other half gather under the tree, waiting for the iguanas to land. When an iguana is found, the coatis collectively hunt it (Figure 10).

a. True boxes at the time of labeling b. Low-quality samples c. High-quality samples

Figure 9 Low-quality and high-quality samples produced during training

For the coatis on the trees, their position is updated as follows:

$$x_i(j+1) = x_i(j) + random\cdot(x_{best}(j) - I\cdot x_i(j)),\ i = 1,2,\ldots,\frac{N}{2} \tag{8}$$

where, I is a random integer in the range [1, 2]; N is the coati population size; $x_i(j)$ represents the current i-th individual coati; and $x_{best}(j)$ denotes the current best position of the coati population.

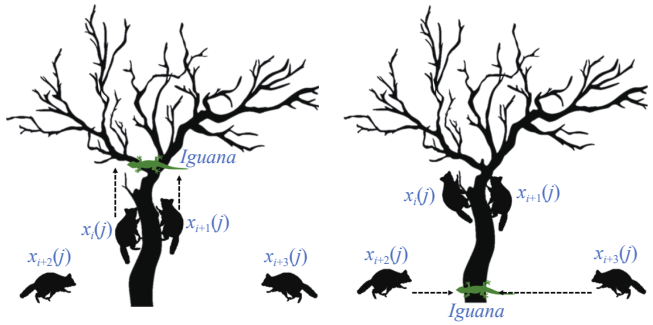

Figure 10 Coatis hunting iguanas

For iguanas and coatis on the ground, their positions are updated as follows:

$$Iguana_{ground}(j) = lb_j + random \cdot (ub_j - lb_j) \tag{9}$$

$$x_i(j+1) = \begin{cases} x_i(j) + r \cdot (Iguana_g(j) - I \cdot x_i(j)), \ if \ F(Iguana_g) < F(x_i) \\ x_i(j) + r \cdot (x_i(j) - Iguana_g(j)), \ else \\ i = \dfrac{N}{2} + 1, \ \dfrac{N}{2} + 2, \ldots, N \end{cases} \tag{10}$$

where, ub_j and lb_j are the upper and lower limits of the current hyperparameter, respectively; $Iguana_g(j)$ is the current position of the iguana, which represents the global optimal solution. $F(Iguana_g)$ and $F(x_i)$ represent the fitness values of the iguana and the i-th individual coati, respectively. The current position of the individual coati after trapping movement was calculated using Equation (10).

Coatis will flee their original location and seek a safe location to escape from predators (Figure 11). This behavior is modeled as described by the following equations:

$$lb_j^l = \frac{lb_j}{t}, \ ub_j^l = \frac{ub_j}{t}, \ t = 1,2,...,T \tag{11}$$

$$X_i(j+1) = X_i(j) + (1 - 2random)(lb_j^l + random(ub_j^l - lb_j^l)), i = 1,2,\ldots,N \tag{12}$$

$$X_i(j+1) = \begin{cases} X_i(j+1), & F(X_i(j+1)) < F(X_i(j)) \\ X_i(j), & F(X_i(j+1)) < F(X_i(j)) \end{cases} \tag{13}$$

where, t represents the current number of iterations. The upper and lower limits ub_j and lb_j of the current hyperparameters are used to obtain lb_j^l and ub_j^l by division with t. Equation (12) calculates the new position of the individual coati moving after escaping from the predator. Each coati in the population employs a greedy strategy to update its position after movement.

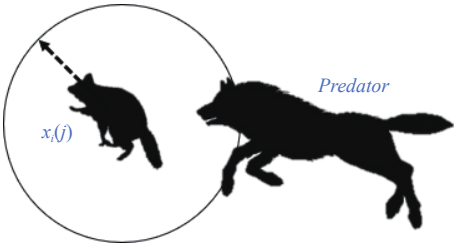

Figure 11 Coati escaping from a predator

Hyperparameter optimization steps in the YOLO-CW model.

(1) Data division: the collected dataset is divided into training and validation sets.

(2) Initialization of parameters and updating of initial historical optimum: Relevant parameters, such as the number of populations N, number of iterations T, and upper and lower limits u_b and l_b, are set. The initial position of each individual is initialized, and the initial global optimal solution X_{gbest} is saved.

(3) Hyperparameter optimization and positional update: First, each individual is subjected to global search using Equations (8) and (10). The local search is performed using Equations (12) and (13). Finally, the position update is performed using the greedy strategy comparison. The value with the higher fitness value is selected for the final position update.

(4) Update the historical optimum: After updating the position, the optimal solution X_{best} in the current iteration t is selected and compared with the global optimal solution X_{gbest}. If $X_{best} > X_{gbest}$, the global optimal solution is updated; otherwise, no change is made.

(5) Algorithm termination: When the termination condition is reached, the global optimal solution X_{gbest} is used to train the model to obtain the final model (Figure 12).

4 Experimental results and analysis

4.1 Experimental environment and model metrics

The primary environment for this study was based on the Pytorch 1.9.0 deep learning framework with related software, including CUDA 11.1, cuDNN 11.1, and Python 3.6.9. All training was conducted on Windows 10, with the server equipped with an Intel (R) Core (TM) i9-10900K CPU @ 3.70 GHz, an RTX3080 GPU, and 32 GB of RAM. Each model was trained using a cosine annealing strategy to reduce the learning rate along with a stochastic gradient descent optimizer. The epoch and batch sizes were set to 300 and 32, respectively.

Given that unripe citrus fruits are prone to misdetection and omission in complex backgrounds, and considering that the model will eventually be ported to an edge computing platform, the main evaluation metrics used in this study are Precision (P), mean Average Precision with an IoU threshold of 0.5 (mAP@0.5), Computation (GFlops), Parameters, and Model Size (Size). Other evaluation metrics included Recall (R) and mAP calculated at multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05 (mAP@0.5:0.95). Precision, Recall, and mAP were calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{16}$$

$$mAP = \frac{1}{n}\left(\sum_{i=1}^{n} AP_i\right) \tag{17}$$

True Positive (TP) refers to the number of correctly predicted unripe citrus fruits, and False Positives (FP) indicate the number of samples where the model predicted the wrong target as unripe citrus fruit. False Negatives (FN) represent the number of unripe citrus fruits present in the image but not detected by the model. To evaluate the model, Precision and Recall were plotted on the horizontal and vertical axes, respectively, to create a Precision-Recall (P-R) curve. The area under the P-R curve is referred to as the AP. The mAP is the average AP across different categories. Given that this study focused on a single target category—unripe citrus fruits— the value of n was 1.

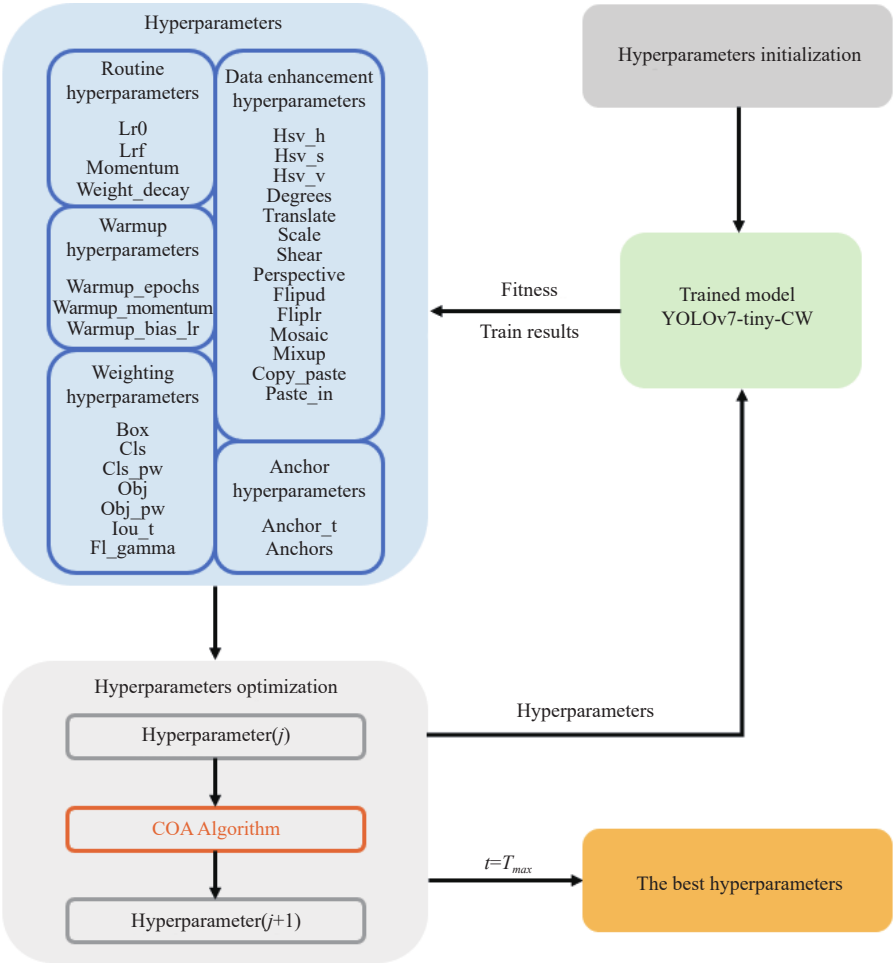

Figure 12 Hyperparameter optimization using the Coati Optimization Algorithm

4.2 Analysis of loss function WIoU versions

To evaluate the efficacy of different WIoU variants for citrus fruit target detection, YOLOv7-tiny was used as the baseline model. The CIoU loss function was replaced with three WIoU variants, and the models were trained. The WIoU version with superior performance in terms of Precision, Recall, mAP@0.5, and mAP@0.5:0.95 was selected for subsequent optimization.

Table 1 and Figure 13 list that WIoUv1 achieved the highest Recall (90.79%), followed by WIoUv3 (90.49%). The other three models had lower values than WIoUv1 and WIoUv3. Although the WIoUv2 system exhibited the highest Precision (93.56%), its Recall and mAP@0.5:0.95 metrics were notably inferior to those of the WIoUv1 and WIoUv3. WIoUv3 outperformed the other versions in terms of mAP@0.5 and mAP@0.5:0.95, with improvements of 0.35% and 3.83% higher, respectively, over WIoUv1.

Table 1 Experimental results of three WIoU versions on an unripe citrus fruit dataset

Methods	*P*/%	*R*/%	mAP@0.5/%	mAP@0.5:0.95/%
YOLOv7-tiny+WIoUv1	92.00	90.79	95.47	74.52
YOLOv7-tiny+WIoUv2	93.56	88.89	95.85	73.78
YOLOv7-tiny+WIoUv3	91.68	90.49	96.20	78.35

WIoUv1 employs anchor frame centroid distance attention based on the distance metric. However, given the prevalence of overlapping unripe citrus fruits and foliage shade in this study, the potential for low-quality samples is high. Anchor frame centroid distance attention was ineffective in handling the challenges posed by overlapping unripe citrus fruits and foliage shade, leading to suboptimal mAP@0.5 and mAP@0.5:0.95 scores. Although the monotonic focusing mechanism in WIoUv2 has demonstrated efficacy in classification tasks, its application in this study, involving a single target class (unripe citrus fruits), resulted in low Recall and low mAP@0.5 and mAP@0.5:0.95 scores. WIoUv3 uses outliers to measure the quality of anchor frames. High-quality anchor frames are assigned a small gradient to allow the model to focus on normal quality anchor frames. Conversely, low-quality anchor frames are assigned a smaller gradient gain to prevent low-quality samples from generating harmful gradients during training. This gradient gain changes dynamically throughout training, effectively addressing issues such as unripe citrus fruit overlap and foliage shade. Consequently, the WIoUv3 achieved higher mAP@0.5 and mAP@0.5:0.95 scores than the WIoUv1 and WIoUv2. Therefore, WIoUv3 was selected for subsequent modeling improvements.

4.3 Analysis of ablation experiment results

To analyze the individual contributions of the ELAN-TC module and WIoU loss function, ablation experiments were conducted on an unripe citrus fruit dataset. Using YOLOv7-tiny as the base model, the ELAN-TC module and the improved loss function WIoU were incrementally added to the base model. The experimental results are presented in Table 2.

Table 2 demonstrates that incorporating the improved ELAN-TC module and loss function significantly enhanced model accuracy and mAP@0.5. The Computation and Parameters are reduced compared to the baseline model. This can be attributed to the strategic integration of the CBAM into the ELAN-T module ahead of the P3 detection head to improve the model's feature

extraction capability for small targets in both channel and spatial dimensions. The model's ability to detect unripe citrus fruits in small targets and complex backgrounds was enhanced through the integration of the WIoUv3 loss function, which uses a dynamic, non-monotonic focusing mechanism that leverages 'outliers' as an alternative to the IoU for quality assessment of the anchor frame and provides a judicious gradient gain assignment strategy. WIoUv3 effectively mitigates the detrimental gradients in low-quality samples, such as overlapping or obscured unripe citrus fruits.

Model 1 (YOLOv7-tiny + ELAN-TC model) exhibited improvements of 2.37% and 0.33% in Precision and mAP@0.5, respectively, compared to the baseline model. These enhancements highlight that the improved channel and spatial attention of the ELAN-TC module effectively improved the detection of targets. And compared to the baseline model YOLOv7-tiny, the Computation and Parameters are reduced by 5.03% and 3.53%. Model 2 (YOLOv7-tiny + WIoUv3 model) yielded improved Precision and mAP@0.5 of 0.44% and 0.29% compared with the baseline model. These results underscore the efficacy of the dynamic non-monotonic focusing mechanism of WIoUv3 in enhancing detection accuracy. Model 3 (YOLOv7-tiny+ELAN-TC+WIoUv3 model) achieved the most substantial improvements, with Precision and mAP@0.5 increasing by 2.59% and 0.59%, respectively, compared to the baseline model.

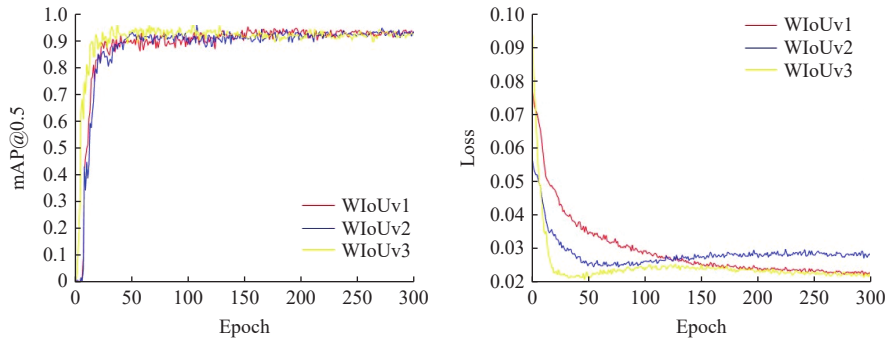

Figure 13 mAP@0.5 curves and loss curves for the three versions of WIoU on the unripe citrus fruit dataset

Table 2 Results of ablation experiments on an unripe citrus fruit dataset

Methods	Baseline	ELAN-TC	WIoUv3	*P*/%	mAP@0.5/%	GFlops	Parameters/10^6
YOLOv7-tiny	√			92.12	95.91	13.9	6.23
Model 1	√	√		94.49	96.24	13.2	6.01
Model 2	√		√	92.56	96.20	13.9	6.23
Model 3	√	√	√	94.71	96.50	13.2	6.01

4.4 Grad-CAM analysis of the YOLO-CW model

To further illustrate the efficacy of the ELAN-TC module in extracting unripe citrus fruit features, particularly in challenging scenarios involving small targets and complex backgrounds, a comparative analysis was conducted between YOLOv7-tiny and the P3 detection head of YOLO-CW using Grad-CAM. The P3 detection head is mainly responsible for recognizing small-sized targets and can capture unripe citrus in the input images, while the P4 and P5 detection heads are mainly responsible for recognizing medium-sized and large-sized targets, respectively.

A comparison of Figures 14a and 14b reveals that YOLOv7-tiny exhibits a degree of feature information loss, which adversely affects detection performance. In contrast, YOLO-CW not only preserves the target feature information, but also improves attention to the target region. Figures 14c and 14e show that although YOLOv7-tiny focused on the target, its attention was dispersed and affected by the complex background, which led to suboptimal detection. In contrast, Figures 14d and 14e show that the proposed YOLO-CW focused on the target, effectively preserving target features and improving detection accuracy.

4.5 Comparative analysis with other detection models

The proposed YOLO-CW model was compared to other single-stage target detection models, including YOLOv7, YOLOR-p6[32], YOLOv8-n, YOLOv9-c[33], YOLOv10[34], YOLO11-n[35], YOLO12-n[36], and YOLO13-n[37]. The comparison results are listed in Table 3 and Figure 15.

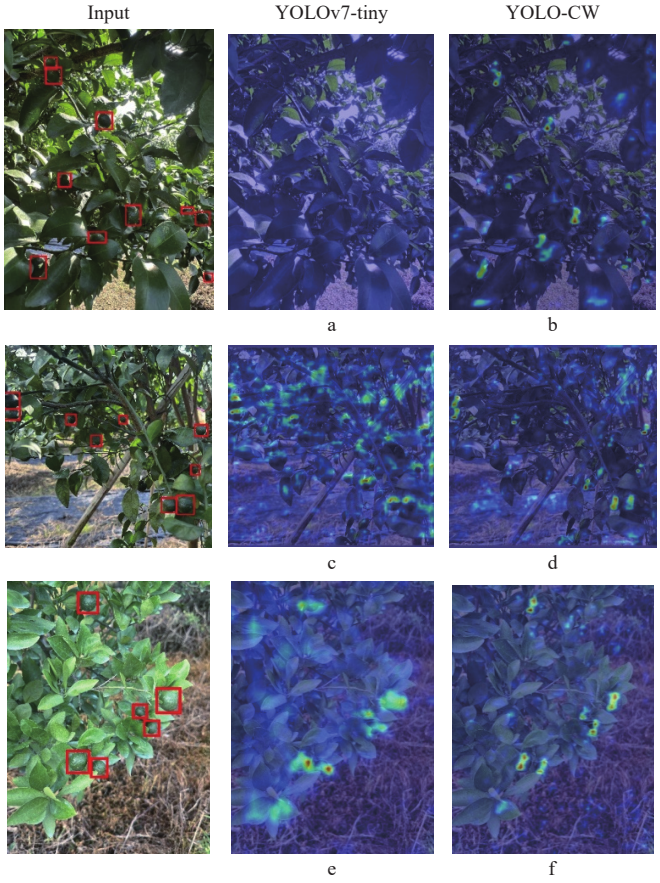

Figure 14 Grad-CAM of the P3 detection heads of YOLOv7-tiny and YOLO-CW trained on an unripe citrus fruit dataset

The YOLO-CW model outperformed the other detection models in terms of both Precision and mAP@0.5. Although the YOLOv7 model introduced an efficient ELAN module, the proposed YOLO-CW model exhibited superior performance in these metrics by 3.4% and 0.09%, respectively. The YOLOv7

incurred a Computation 8.07 times greater, with 6.25 times more Parameters and Size 6.09 times larger than the YOLO-CW model. The YOLOr-p6 model performed 1.32% and 1.05% lower than the YOLO-CW model in terms of Precision and mAP@0.5, respectively. The Computation, Parameters, and Size of the YOLOr-p6 model were 6.15, 6.19, and 6.04 times those of the YOLO-CW model, respectively. The YOLOv8-n model, despite its smaller Computation, Parameters, and Size than the YOLO-CW model, exhibited 2.8% and 8.3% lower Precision and mAP@0.5, respectively. The YOLOv9 model, which incorporates a generalized ELAN (GELAN) module, performed lower than the YOLO-CW model in terms of Precision and mAP@0.5 by 3.91% and 0.19%, respectively. Notably, the Computation, Parameters, and Size of YOLOv9 were significantly higher, at 21.85, 10.07, and 8.77 times those of the YOLO-CW model. YOLOv10 reduces inference latency by eliminating the reliance on non-maximum suppression. The YOLOv10-n model has lower Computational Cost, fewer Parameters, and smaller Model Size than the YOLO-CW model, while its Precision and mAP@0.5 are 6.34% and 7.55% lower, respectively. YOLO11 improves CSPNet and incorporates a more efficient C2PSA attention mechanism. The YOLO11-n model has lower Computational Cost, fewer Parameters, and smaller Model Size than the YOLO-CW model, but its Precision and mAP@0.5 are 2.41% and 1.1% lower than those of the YOLO-CW model. YOLO12 adopts an attention-centric framework. The YOLO12-n model has lower Computational Cost, fewer Parameters, and smaller Model Size than the YOLO-CW model, while its Precision and mAP@0.5 are 0.11% and 2.1% lower, respectively. YOLO13 introduces the HyperACE mechanism to achieve global cross-position and cross-scale feature fusion and enhancement. The YOLO13-n model has lower Computational Cost, fewer Parameters, and smaller Model Size than the YOLO-CW model, but its Precision and mAP@0.5 are 0.91% and 2% lower than those of the YOLO-CW model. As shown in Figure 15, the proposed YOLO-CW occupied the largest area enclosed by the Precision and mAP@0.5 space while maintaining smaller areas in terms of Computation, Parameters, and Model Size.

Table 3 Performance comparison of other detection models on the unripe citrus fruit dataset

Methods	*P*/%	mAP@0.5/%	GFlops	Parameters	Size/MB
YOLOv7	91.31	96.41	106.5	3.76×10^7	71.3
YOLOR-p6	93.39	95.45	81.8	3.72×10^7	70.7
YOLOv8-n	91.91	88.20	8.2	3.01×10^6	6.2
YOLOv9-c	90.80	96.31	288.4	6.05×10^7	102.8
YOLOv10-n	88.37	88.95	8.2	2.69×10^6	5.5
YOLO11-n	92.30	95.40	6.3	2.58×10^6	5.2
YOLO12-n	94.60	94.40	5.8	2.5×10^6	5.2
YOLO13-n	93.80	94.50	6.2	2.49×10^6	5.2
YOLO-CW	94.71	96.50	13.2	6.01×10^6	11.7

In conclusion, the proposed YOLO-CW model achieved the highest Precision and mAP@0.5 of 94.71% and 96.50%, respectively. The Computation, Parameters, and Size of the YOLO-CW model were 13.2 GFlops, 6.01×10^6, and 11.7 MB.

Tracking and counting unripe citrus fruits in natural citrus orchards requires a small, computationally efficient, and highly accurate model that can be ported to a portable edge computing platform. In the comparison experiments, the YOLO-CW model outperformed the other models in terms of Computation, Parameters, and Size. This makes it suitable for deployment on edge computing platforms, effectively addressing the challenge of tracking and counting unripe citrus fruits.

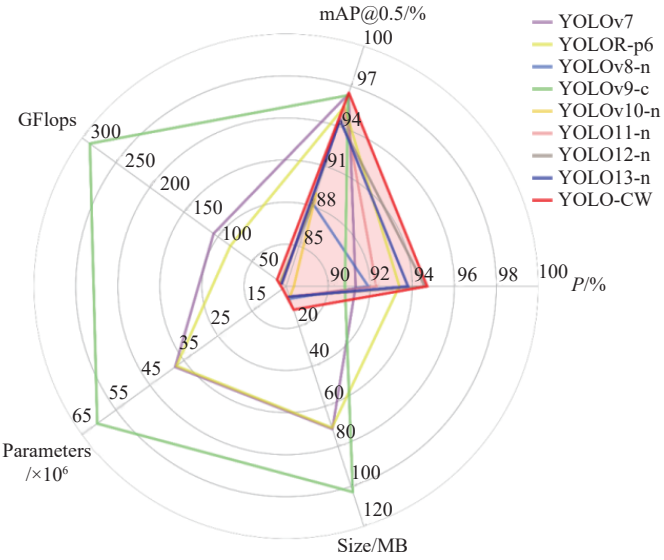

Figure 15 Radar plot of each model parameter

4.6 Hyperparametric optimization based on the Coati Optimization Algorithm

In this study, 20 hyperparameters of the YOLO-CW model were optimized to obtain the optimal set of hyperparameters for the unripe citrus fruit target detection model, which further improved the detection accuracy. The hyperparameters optimized include: routine hyperparameters, warmup hyperparameters, data enhancement hyperparameters, and anchor hyperparameters. Optimizing the routine hyperparameters effectively avoided model overfitting and underfitting. Optimizing the warmup hyperparameters allowed the network to gradually adapt to the characteristics of the given dataset. Optimizing the data enhancement hyperparameters improved the generalizability of the model. Optimizing the anchor hyperparameters improved the accuracy of anchor frames when matching targets. The optimization process was conducted over 50 epochs with 100 iterations.

Scatterplots were plotted with mAP@0.5 and mAP@0.5:0.95 on the horizontal and vertical coordinates, respectively, with the number of iterations t=0 to t=50, as shown in Figure 16. After population initialization, the hyperparameters were optimized using COA. When $t \in [0,20]$, mAP@0.5 and mAP@0.5:0.95 continuously converged to the optimal solution. When $t \in [21,50]$, mAP@0.5 and mAP@0.5:0.95 converged to the region around the optimal solution and oscillated back and forth in this region. This indicates that the COA quickly converged to the desired optimal solution for mAP@0.5 and mAP@0.5:0.95 when optimizing the hyperparameters of the YOLO-CW model.

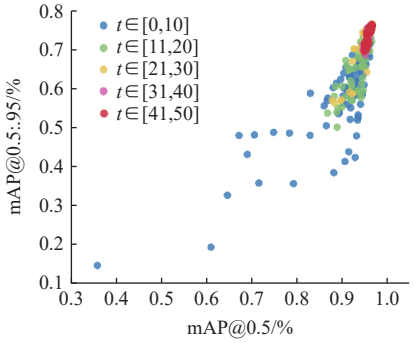

Figure 16 50 iterations of scatterplot for all individuals

Figure 17 shows the optimal solutions of mAP@0.5 and mAP@0.5:0.95 after each iteration. Table 4 and Table 5 compare the hyperparameters and detection results before and after hyperparameter optimization, respectively. YOLO assigns weights of 0.1 and 0.9 to mAP@0.5 and mAP@0.5:0.95, respectively, in the fitness function Fitness(x). During the iteration process, mAP@0.5 fluctuates and eventually stabilizes, whereas mAP@0.5:0.95 generally exhibits an upward trend and then stabilizes. For the hyperparameters in Table 4, the data augmentation hyperparameters include Mosaic, Mixup, Copy_paste, and Paste_in. The values of these hyperparameters represent the probability of enabling the corresponding augmentation. For example, after hyperparameter optimization, the value of the Mosaic hyperparameter changed from 1.0 to 0.600 15, indicating that the probability of Mosaic augmentation occurring was optimized from 1 to 0.60015.

As shown in Figure 17, when the YOLO-CW hyperparameters were optimized by COA, mAP@0.5 fluctuated but generally increased until the 25th iteration, whereas mAP@0.5:0.95 consistently increased. After the 25th iteration, mAP@0.5 and mAP@0.5:0.95 stabilized at 96.62% and 76.71%, respectively. As shown in the table, the YOLO-CW model optimized by COA exhibited improvements in Recall, mAP@0.5, and mAP@0.5:0.95 compared to the model without optimization, with improvements of 0.65%, 0.12%, and 0.55%, respectively. These results indicate that the COA converged rapidly during hyperparameter optimization of the YOLO-CW model, which effectively improved detection accuracy. In addition, this enhancement improved the performance of the YOLO-CW model. Porting the model to an edge computing platform made it more suitable for tracking and counting unripe citrus fruits in natural environments.

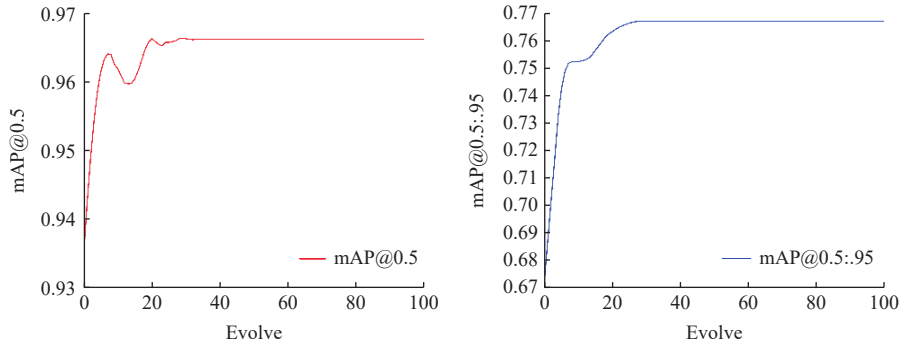

Figure 17 Line plots of the optimal solutions of mAP@0.5 and mAP@0.5:0.95 for 100 iterations

Table 4 Results before and after hyperparameter optimization

Hyperparameters	Before optimization	After optimization	(*lb*, *ub*)
Lr0	0.001	0.024 50	(0.000 05, 0.1)
Lrf	0.01	0.013 94	(0.01, 1.0)
Momentum	0.937	0.604 70	(0.6, 0. 98)
Weight_decay	0.0005	0.000 05	(0.0, 0.001)
Warmup_epochs	3.0	2.147 80	(0.0, 5.0)
Warmup_momentum	0.8	0.321 60	(0.0, 0.95)
Warmup_bias_lr	0.1	0.057 22	(0.0, 0.2)
Anchor_t	4.0	2.509 40	(2.0, 8.0)
Hsv_h	0.015	0.003 88	(0.0, 0.1)
Hsv_s	0.7	0.474 71	(0.0, 0.9)
Hsv_v	0.4	0.189 83	(0.0, 0.9)
Degrees	0.0	1.357 30	(0.0, 45.0)
Translate	0.2	0.160 13	(0.0, 0.9)
Scale	0.9	0.038 93	(0.0, 0.9)
Shear	0.0	0.088 86	(0.0, 10.0)
Flipud	0.0	0.000 00	(0.0, 1.0)
Mosaic	1.0	0.600 15	(0.0, 1.0)
Mixup	0.0	0.290 85	(0.0, 1.0)
Copy_paste	0.0	0.277 47	(0.0, 1.0)
Paste_in	0.15	0.042 40	(0.0, 1.0)

Table 5 Comparison of detection results before and after hyperparameter optimization

	P/%	*R*/%	mAP@0.5/%	mAP@0.5:0.95/%
Before optimization	94.71	89.53	96.50	76.16
After optimization	93.80	90.18	96.62	76.71

5 Unripe citrus fruit tracking and counting based on StrongSORT combined with edge computing platform

Citrus orchards are predominantly located in hilly areas, which are characterized by complex natural environments. The large number of unripe citrus fruits in these orchards requires manual counting for yield measurement, which incurs significant labor costs. Therefore, this study employed StrongSORT combined with an edge computing platform to track and count unripe citrus fruits. The devices used include Jetson AGX Xavier (NVIDIA Corporation, Santa Clara, USA), a touchscreen display, a mobile power supply, and multiple tablets or mobile phones. This equipment offers high portability and fast detection speed. The use of multiple wireless IP cameras improves tracking and counting efficiency. This portable detection system is well-suited for the complex terrain of citrus orchards, addressing the problem of high labor costs for yield measurement and paving the way for precise management and protection of citrus orchards.

5.1 StrongSORT-based unripe citrus fruit target tracking algorithm

The SORT target tracking algorithm integrates a target detector and tracker. The target detector is a pre-trained target detection model, and the tracker is a combination of the Kalman filter and Hungarian algorithm[38]. To achieve target tracking, Kalman filtering predicts the position and state of the target in the next frame, and the Hungarian algorithm matches the predicted location and state with the actual detection data. The StrongSORT algorithm improves upon the SORT algorithm[39] by introducing re-recognition technology, which uses a model to extract the target's appearance features while tracking its movement. To enhance tracking accuracy, the features extracted from consecutive frames are used to determine object identity or persistence. StrongSORT uses a BoT

with ResNeSt50 as the backbone network, which is more effective for extracting the individual characteristics of unripe citrus fruits and reducing the influence of complex backgrounds. In addition, StrongSORT employs an EMA feature library with NSA Kalman filtering to reduce noise during real-time tracking and counting of unripe citrus fruit. Using linear matching improves tracking effectiveness and reduces the likelihood of losing track of unripe citrus fruits. This allows the StrongSORT algorithm to track more consistently, match more accurately, and adapt to more complex movement patterns (Figure 18).

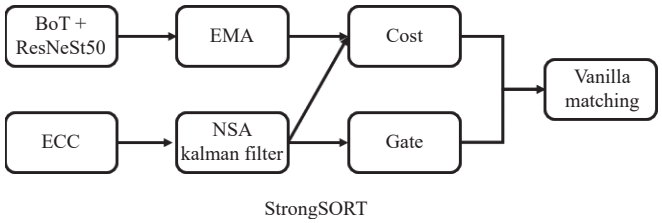

Figure 18 StrongSORT architecture

This study employed the StrongSORT algorithm for the target tracking detection and counting of unripe citrus fruits, which was performed using the YOLO-CW model. The StrongSORT algorithm assigns unique IDs to each detected unripe citrus fruit and tracks them individually as the fruit moves. When a new ID is assigned, the count increases by one; duplicate IDs do not result in cumulative counts. At the end of the inspection, the total number of citrus fruit was counted to track citrus yield.

5.2 Deployment of edge computing platforms

The edge computing platform used in this study was Jetson AGX Xavier, which is based on the ARM architecture and manufactured by NVIDIA, USA. The device measures 100 mm × 87 mm, features an 8-core CPU and 512-core GPU, and can deliver up to 32 TOPS of computing power. The real-time frame capture devices were a Mate Pad tablet (Huawei, Shenzhen, China) and an iPhone 7 Plus (Apple, Cupertino, USA).

Given that most citrus orchards are located in hilly areas with complex terrain, real-time detection and counting are essential. Considering the limitations of wired cameras, real-time image capture can be realized using multiple wireless IP cameras. The YOLO-CW model was ported to an edge computing platform to acquire real-time unripe citrus fruit images from multiple tablets and cell phone cameras via wireless IP. The StrongSORT tracking algorithm tracked and counted unripe citrus fruits. The processed data were then displayed in real time on a touch-screen display, tablet, or mobile phone (Figure 19).

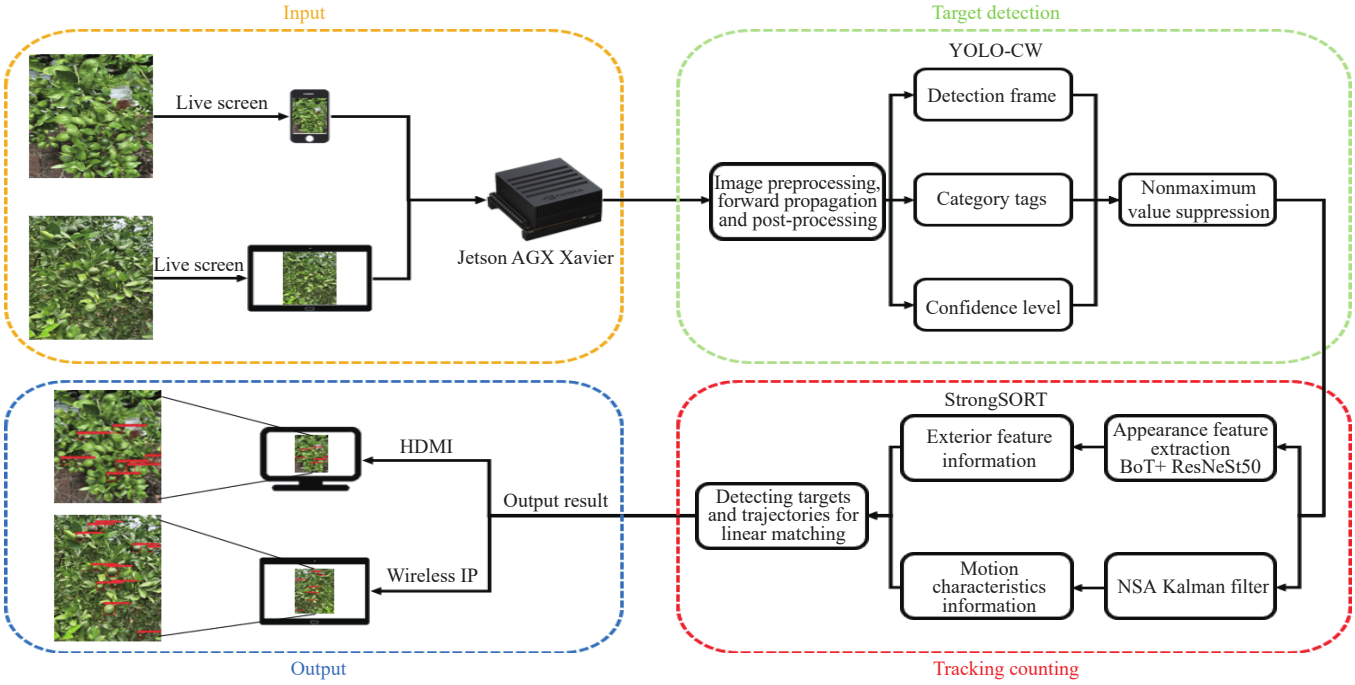

Figure 19 Dual input and output unripe citrus fruit counting flowchart

To evaluate the counting accuracy of the tracking and counting algorithm, manual and detection counting of unripe citrus fruits were compared in orchards with varying fruit density, foliage, and background complexity. The error between manual counting and detection counting was calculated, and the results are shown in Table 6. As indicated in the table, for scenarios with obvious fruit, minimal shade, heavy shade, and complex backgrounds, the error between manual and detected counting was within 3. This result demonstrates that the counting ability of the algorithm closely matches that of manual statistical counting, thus meeting the requirements for counting unripe citrus fruits in citrus orchards.

To evaluate the efficiency and energy consumption of wireless IP camera-based citrus yield detection systems for tracking and counting in natural citrus orchards, a comparative analysis of the baseline model YOLOv7-tiny and the YOLO-CW model was conducted. The comparison encompasses two distinct scenarios: single and multiple wireless IP cameras. The input devices comprised a Mate Pad (Huawei, Shenzhen, China) and an iPhone 7 Plus (Apple, Cupertino, USA), while output was displayed on a touchscreen and a Mate Pad (Huawei, Shenzhen, China) (Figure 20). The results are listed in Table 7.

When processing an input image, if the image size is not 640×640, the YOLO-CW model performs preprocessing using letterbox, such as scaling with aspect ratio preserved and padding with gray borders. So Table 7 indicates that under a screen capture resolution of 640×480, the average frame rate achieved by the YOLOv7-CW model using a single wireless IP camera is 23 FPS. Concurrently, the power consumption on the edge platform is 16.82

W, reflecting a decrease of 3.89% in comparison to the power consumption by the YOLOv7-tiny model. In the context of using multiple wireless IP cameras, the average frame rate for the YOLO-CW model is 23 FPS and 22 FPS. This is not significantly different from the single wireless IP camera's average frame rate of 23 FPS. Moreover, there is a notable improvement of 10% in the average frame rate when compared to the YOLOv7-tiny model. For the YOLO-CW model, the power consumption of the edge platform when using multiple wireless IP cameras is 17.11 W. This is a slight increase of 0.29 W compared to the power consumption when using a single wireless IP camera. Furthermore, it represents a reduction of 7.62% in power consumption compared to the YOLOv7-tiny model when deployed with multiple wireless IP cameras. This indicates that, due to the YOLO-CW model's reduced Computation and Parameters compared to YOLOv7-tiny, fewer resources are expended when the edge platform uses the wireless IP camera for the inference procedure. This results in an enhancement of inference speed and decrease in power consumption. The YOLO-CW model uses multiple wireless IP cameras, which do not exert an adverse effect on the average frame rate and power consumption in comparison to a single camera. Consequently, use of multiple wireless IP cameras in citrus orchards effectively enhances unripe citrus tracking and counting efficiency, thereby reducing labor costs associated with yield assessment.

In summary, this study presents the YOLO-CW model optimized via COA hyperparameter tuning and deployed on an edge computing platform. Integrated with the StrongSORT tracking algorithm and multiple wireless IP cameras, the system accurately tracks and counts unripe citrus fruits in orchards. Compared to manual counting, the proposed method exhibits minimal error. The use of multiple wireless IP cameras addresses the limitations of wired cameras and enhances tracking efficiency through multi-camera outputs. Compared to the OrangeYolo model by Zhang and OrangeSort fruit tracking, the proposed portable unripe citrus fruit tracking and counting device offers advantages in terms of its lightweight design, high detection accuracy, and low power consumption. It effectively performs real-time tracking and counting in complex orchard environments, providing a foundation for future yield estimations of unripe citrus fruits. However, inherent delays in wireless transmission can affect real-time image transmission. An excessive number of wireless IP cameras can strain the edge computing platform, leading to image latency and increased power consumption. Therefore, continuous debugging is required to reduce latency, and replacing the edge computing platform with a more powerful one is recommended.

Table 6 Manual and algorithmic data related to the detection of unripe citrus fruit counts

Detecting target images	Actual number/pc	Model count/pc	Number of errors/each
	9	8	1
	16	15	1
	16	18	2
	35	34	1

Table 7 Analysis of the results of tracking and counting unripe citrus fruit using wireless IP cameras

Model	Number of IP cameras	Input acquisition devices	Resolution	Average frame rate/FPS	Power/ W
YOLOv7-tiny	Single	Device A	640×480	23	17.50W
	Multiple	Device A	640×480	21	18.45W
		Device B	640×480	19	
YOLO-CW	Single	Device A	640×480	23	16.82W
	Multiple	Device A	640×480	23	17.11W
		Device B	640×480	22	

Note: Device A is Huawei Mate Pad and Device B is IPhone7 Plus.

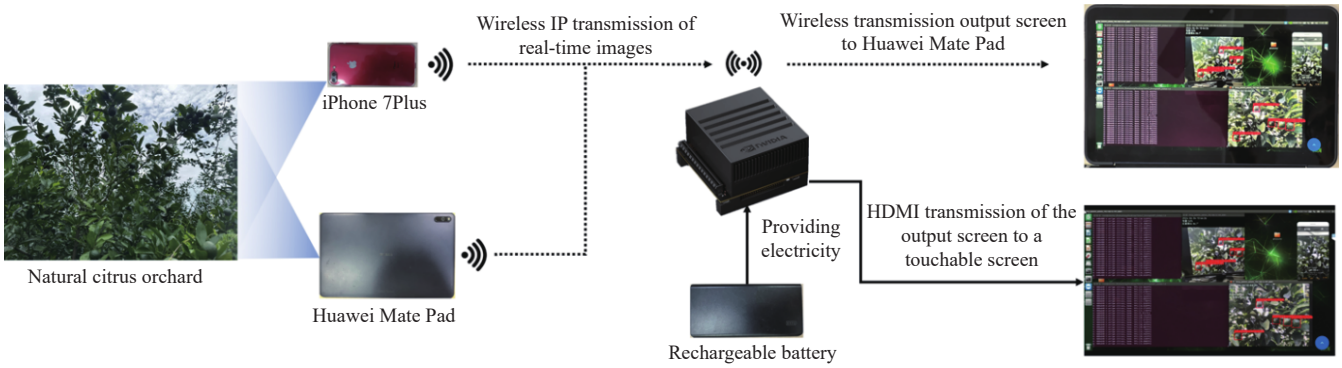

Figure 20 Unripe citrus fruit tracking and counting inspection equipment

6 Conclusions

This study proposes an improved YOLO-CW model using YOLOv7-tiny as the baseline model to address the challenges posed by small, overlapping unripe citrus fruits in complex backgrounds. The proposed model's compact architecture, characterized by reduced Computation and Parameter count, is well-suited for deployment on edge computing platforms.

(1) The integration of the CBAM attention mechanism into the ELAN-T module ahead of the P3 detection head of YOLOv7-tiny significantly enhanced the feature extraction capability of the model for unripe citrus fruits in small targets and complex backgrounds, and reduced the Computation and Parameters.

(2) The adoption of the WIoUv3 loss function mitigated the harmful gradient of low-quality instances caused by overlapping unripe citrus fruits and foliage shading, thereby enhancing the ability of the model to detect unripe citrus fruits in complex scenarios.

(3) Hyperparametric optimization of the YOLO-CW model using the coati algorithm significantly enhanced the detection accuracy. The optimized model achieved *P*(%), *R*(%), mAP@0.5(%), and mAP@0.5:0.95(%) values of 93.80%, 90.18%, 96.62%, and 76.71%, respectively.

(4) The optimized model was deployed on an edge computing platform integrated with the StrongSORT algorithm for tracking and counting. Real-time image acquisition was achieved using multiple wireless IP cameras operating at a frame rate of 23 FPS and power consumption of 17.11 W. The output from the system was displayed in real time on a designated device.

The proposed model has strong potential for deployment on edge computing platforms and exhibits exceptional performance in unripe citrus fruit detection, tracking, and counting. These capabilities provide a solid foundation for future research on citrus orchard yield assessment and fine management. To address limitations such as prolonged hyperparameter optimization, the constrained processing speed of edge computing platforms, and wireless transmission latency, subsequent research will explore less computationally demanding intelligent optimization algorithms, model compression techniques for accelerated inference, and strategies to mitigate latency during wireless data transfer.

Acknowledgements

This work was supported by the Science and Technology Projects in Guangzhou (2024B03J1309), the Natural Science Foundation of China (32271997), the earmarked fund for CARS (CARS-26), and the Guangdong Provincial Special Fund for Modern Agriculture Industry Technology Innovation Teams (2026CXTD10). The authors would also like to thank all those who contributed to this research and the authors cited in the paper.

[References]

[1] Luo X W, Liao J, Zang Y, Ou Y G, Wang P. Developing from mechanized to smart agricultural production in China. Chinese Journal of Chemical Engineering, 2022; 24(1): 46–54. (in Chinese)

[2] Yang T, Sun F C, Huang B, Wu B Q, Ran G Z. Research progress on key technologies of orchard operating platform. Journal of Chinese Agricultural Mechanization, 2024; 45(1): 152–159. (in Chinese)

[3] Gan H, Lee W S, Alchanatis V, Ehsani R, Schueller J K. Immature green citrus fruit detection using color and thermal images. Computers & Electronics in Agriculture, 2018; 152: 117–125.

[4] Wu D H, Lyv S C, Jiang M, Song H B. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers & Electronics in Agriculture, 2020; 178: 105742.

[5] Song H B, Shang Y Y, He D J. Review on deep learning technology for fruit target recognition. Transactions of the CSAM, 2023; 54(1): 1–19. (in Chinese)

[6] Qiu C, Tian G Z, Zhao J W, Liu Q, Xie S J, Zheng K. Grape maturity detection and visual pre-positioning based on improved YOLOv4. Electronics, 2022; 11(17): 2677.

[7] Li P, Zheng J S, Li P Y, Long H W, Li M, Gao L H. Tomato maturity detection and counting model based on MHSA-YOLOv8. Sensors, 2023; 23(15): 6701.

[8] Yang S Z, Wang W, Gao S, Deng Z P. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. Computers & Electronics in Agriculture, 2023; 215: 108360.

[9] Chen F J, Chen C, Zhu X Y, Shen D Y, Zhang X W. Detection of *Camellia oleifera* fruit maturity based on improved YOLOv7. Transactions of the CSAE, 2024; 40(5): 177–186. (in Chinese)

[10] Wang Z F, Zhang Z H, Lu Y Q, Luo R, Niu Y, Yang X B, et al. SE-COTR: A novel fruit segmentation model for green apples application in complex orchard. Plant Phenomics, 2022; 2022: 0005.

[11] Ren R, Sun H X, Zhang S J, Wang N, Lu X Y, Jing J P, et al. Intelligent detection of lightweight "Yuluxiang" pear in non-structural environment based on YOLO-GEW. Agronomy, 2023; 13(9): 2418.

[12] Zhang B, Xia Y Y, Wang R R, Yong W, Yin C H, Fu M, et al. Recognition of mango and location of picking point on stem based on a multi-task CNN model named YOLOMS. Precision Agriculture, 2024; 25: 1454–1476.

[13] Yue K, Zhang P C, Wang L, Guo Z M, Zhang J J. Recognizing citrus in complex environment using improved YOLOv8n. Transactions of the CSAE, 2024; 40(8): 152–158. (in Chinese)

[14] Zhang W L, Wang J Q, Liu Y X, Chen K Z, Li H B, Duan Y L, et al. Deep-learning-based in-field citrus fruit detection and tracking. Horticulture Research, 2022; 9: 003.

[15] Tu S Q, Huang Y F, Liang Y, Liu H X, Cai Y F, Lei H. A passion fruit counting method based on the lightweight YOLOv5s and improved DeepSORT. Precision Agriculture, 2024; 25: 1731–1750.

[16] Shi R, Li T X, Yasushi Y. An attribution-based pruning method for real-time mango detection with YOLO network. Computers & Electronics in Agriculture, 2020; 169: 105214.

[17] Huang H Q, Huang T B, Li Z, Lyu S L, Hong T. Design of citrus fruit detection system based on mobile platform and edge computer device. Sensors, 2022; 22(1): 59.

[18] Lyu S L, Li R Y, Zhao Y W, Li Z, Fan R J, Liu S Y. Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system. Sensors, 2022; 22(2): 576.

[19] Liu Y, Zheng H T, Zhang Y H, Zhang Q J, Chen H L, Xu X Y, et al. "Is this blueberry ripe?": a blueberry ripeness detection algorithm for use on picking robots. Frontiers in Plant Science, 2023; 14: 1198650.

[20] Zhao Z Q, Wang J, Zhao H. Research on apple recognition algorithm in complex orchard environment based on deep learning. Sensors, 2023; 23(12): 5425.

[21] Zhong Z Y, Yun L J, Cheng F Y, Chen Z Q, Zhang C J. Light-YOLO: A lightweight and efficient YOLO-based deep learning model for mango detection. Agriculture, 2024; 14(1): 140.

[22] Kim Y, Chung M. An approach to hyperparameter optimization for the objective function in machine learning. Electronics, 2019; 8(11): 1267.

[23] Li Y, Abdallah S. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 2020; 415: 295–316.

[24] Yu C H, Liu Y K, Zhang W R, Zhang X, Zhang Y H, Jiang X. Foreign objects identification of transmission line based on improved YOLOv7. IEEE Access, 2023; 11: 51997–52008.

[25] Lyu S L, Zhou X, Li Z, Liu X Y, Chen Y C, Zeng W B. YOLO-SCL: a lightweight detection model for citrus psyllid based on spatial channel interaction. Frontiers in Plant Science, 2023; 14: 1276833.

[26] Stefano F S, Laio O S, Anne C R K, Raúl G O, Valderi R Q L. Hypertuned-YOLO for interpretable distribution power grid fault location based on EigenCAM. Ain Shams Engineering Journal, 2024; 15: 102722.

[27] Wang C Y, Bochkovskiy A, Liao H M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022; pp.7464-7475. DOI: 10.48550/arXiv.2207.02696

[28] Woo S, Park J, Lee J, Kweon I. CBAM: Convolutional block attention module. 2018; arXiv preprint arXiv: 1807.06521. DOI: 10.48550/arXiv.1807.06521

[29] Zheng Z H, Wang P, Liu W, Li J Z, Ye R G, Ren D W. Distance-IoU loss: Faster and better learning for bounding box regression. AAAI Conference on Artificial Intelligence, 2020; 34(7): 12993–13000.

[30] Tong Z J, Chen Y H, Xu Z W, Yu R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. 2023; arXiv preprint arXiv: 2301.10051. DOI: 10.48550/arXiv.2301.10051.

[31] Mohammad D, Zeinab M, Eva T, Pavel T. Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems, 2023; 259: 110011.

[32] Wang C, Yeh I, Liao H. You only learn one representation: Unified network for multiple tasks. Journal of Information Science and Engineering, 2021; 39: 691–709.

[33] Wang C, Yeh I, Liao H. YOLOv9: Learning what you want to learn using programmable gradient information. 2024; arXiv preprint arXiv: 2402.13616. DOI: 10.48550/arXiv.2402.13616.

[34] Wang A, Chen H, Liu L H, Chen K, Lin Z J, Han J G, et al. YOLOv10: Real-time end-to-end object detection. 2024; arXiv preprint arXiv: 2405.14458. DOI: 10.48550/arXiv.2405.14458.

[35] Rahima K, Muhammad H. YOLOv11: An overview of the key architectural enhancements. 2024; arXiv preprint arXiv: 2410.17725. DOI: 10.48550/arXiv.2410.17725.

[36] Tian Y J, Ye Q X, David D. YOLOv12: Attention-centric real-time object detectors. 2025; arXiv preprint arXiv: 2502.12524. DOI: 10.48550/arXiv.2502.12524.

[37] Lei M Q, Li S Q, Wu Y H, Hu H, Zhou Y, Zheng X H, et al. YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception. 2025; arXiv preprint arXiv: 2506.17733. DOI: 10.48550/arXiv.2506.17733.

[38] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. IEEE International Conference on Image Processing (ICIP), 2016; pp.3464-3468. DOI: 10.1109/ICIP.2016.7533003

[39] Du Y H, Zhao Z C, Song Y, Zhao Y Y, Su F, Gong T, et al. StrongSORT: Make DeepSORT great again. IEEE Transactions on Multimedia, 2022; 25: 8725–8737.