CIDA-Net: Optimized YOLOv9s-based Young Fruit Detection During Thinning Period

Rongfeng Xu, Xiangchang Huang, Jingyu Yan, Xiude Chen, Weikuan Jia

Abstract—**Fruit thinning operations are a crucial part of the early management process in smart orchards, and the key to automation is the accurate detection of young fruits during the thinning period. However, in unstructured orchard environments, influenced by complex backgrounds and variable lighting, young fruits are often occluded, overlapped, or exhibit similar colors, which poses significant challenges for accurate detection. Therefore, this research employs YOLOv9s as the foundational network to enhance the deep learning model CIDA-Net, tailored for detecting young fruits during the thinning period. This method effectively addresses the challenges of identifying and locating green young fruits that are occluded or overlapped. Firstly, a serpentine dynamic convolution module is incorporated into the backbone network to improve the extraction of edge features. Secondly, the CARAFE structure is incorporated in the feature fusion part, optimizing the upsampling process with a content-aware mechanism to improve the perception of subtle features and effectively process complex image information. Finally, the InnerIoU loss calculation method is introduced into the detection head's localization branch, using auxiliary bounding boxes and dynamic scaling factors to improve loss computation and further enhance bounding box regression accuracy. To evaluate the effectiveness of the algorithm, a dataset of green young fruits during the thinning period is constructed, and experiments are conducted. The results indicate that the accuracy and recall values achieve 91.4% and 79.7%, respectively, highlighting the model's superior efficacy when compared to current mainstream algorithms. This model not only satisfies the detection performance and robustness demands in challenging orchard conditions, but also offers theoretical backing for automated fruit thinning and young fruit growth monitoring in other orchards.**

Index Terms—**Green young fruits, Fruit thinning, CIDA-Net, Object detection**

Manuscript received May 20, 2025; revised August 21, 2025.

This work is supported by Natural Science Foundation of Shandong Province in China (No.: ZR2020MF076); Young Innovation Team Program of Shandong Provincial University (No.: 2022KJ250); New Twentieth Items of Universities in Jinan (2021GXRC049).

R. Xu is a postgraduate student of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (e-mail: 3526974059@qq.com);

X. Huang is an undergraduate of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (e-mail: 2138468601@qq.com);

J. Yan is a postgraduate student of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (e-mail: 1543670705@qq.com)

X. Chen is an associate professor of National Research Center for Apple Engineering and Technology, Taian 271018, China (e-mail: chenxiude@163.com).

W. Jia is an associate professor of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (Corresponding author, phone: +86-531-86181755; fax:+86-531-86181750; e-mail: jwk_ 1982@163.com)

I. Introduction

FRUIT thinning is an essential aspect of early-stage orchard management, aimed at reducing the burden on fruit trees and promoting fruit growth and development [1–4]. As a key orchard management practice, fruit thinning not only improves fruit quality and yield, but also supports the healthy growth and long-term productivity of fruit trees. In modern orchards, fruit thinning is primarily performed manually, which not only consumes significant labor, but also suffers from issues such as insufficient precision and low efficiency, ultimately affecting both the thinning results and the overall effectiveness of orchard management. With the advancement of intelligent agricultural technology, computer vision systems can quickly identify and locate fruits in orchards [5]. When integrated with automated systems, these technologies enable efficient and accurate fruit detection and removal, laying the foundation for research on fruit thinning machinery [6] and further enhancing the intelligence of orchard management.

Fruit recognition methods primarily consist of traditional techniques that rely on manually defined features and deep learning approaches that autonomously extract features. Traditional fruit recognition methods rely on manually extracting features based on visual attributes such as color, geometry, and surface texture, followed by the use of machine learning techniques like Support Vector Machines (SVM) and decision trees to perform target detection. Several studies have utilized these conventional approaches for fruit recognition. For example, Sashuang Sun et al. [7] employed the GrabCut model, utilizing color features to isolate fruits from the background for precise detection of the growth status of green fruits. They employed the Ncut algorithm to precisely segment overlapping fruits and applied geometric feature extraction techniques to reconstruct the segmented targets. Sengupta et al. [8] integrated conventional image processing techniques with machine learning algorithms to identify unripe green citrus fruits within tree canopies. They identified occluded citrus using geometric features, applied SVM for texture classification to locate fruits, employed Canny edge detection and Hough transform to enhance precision, and utilized the Scale-Invariant Feature Transform (SIFT) for keypoint detection, achieving a detection accuracy of 80.4%. Xiaoyang Liu et al. [9] proposed a detection method based on chromatic and geometric attributes. They applied the Simple Linear Iterative Clustering (SLIC) algorithm to segment fruit images into superpixels, extracted chromatic cues to identify candidate areas, and utilized the

Histogram of Oriented Gradients (HOG) to describe shapes and edges, thus achieving fruit recognition and positioning. To efficiently classify different fruit types, Jana et al. [10] combined texture and color features into feature descriptors and trained support vector machines for fruit classification. Although traditional object detection algorithms are relatively mature, they encounter limitations in challenging orchard conditions. Factors such as branch and leaf occlusion, varying lighting conditions, and background clutter complicate manual feature extraction. Moreover, feature descriptors derived from simple visual cues struggle to capture deep semantic information. In addition, traditional methods often suffer from low detection accuracy, slow processing speed, and high computational cost. As a result, traditional fruit recognition techniques struggle to meet the requirements of efficient and accurate fruit identification and localization in real-world applications.

The continuous development of deep learning technology has promoted its wide application in the field of fruit recognition [11-13]. Unlike traditional techniques relying on manual feature extraction, deep learning can autonomously learn multi-level feature from large datasets [14-15]. This feature learning approach not only effectively captures detailed information in images but also adapts to various changes in complex environments, thus offering a clear advantage in fruit detection tasks. One such study by Kong et al. [16] presented an improved Faster R-CNN framework enhanced with a window-based Transformer, designed to mitigate the limitations of conventional Convolutional Neural Networks (CNNs) in complex orchard detection scenarios. Xu et al. [17] introduced a fruit detection approach optimized with YOLOX_m, utilizing CSPDarkNet as the backbone for extracting meaningful features across different scales. A feature fusion pyramid network was incorporated to gather multi-scale information, and the ASPP module was applied to expand receptive fields, resulting in significant performance gains when tested with apple and persimmon data samples. Wang et al. [18] proposed an efficient YOLOv5s-based model with channel pruning to enable fast and accurate identification of immature apples, yielding 95.8% accuracy, a recall rate of 87.6%, and an F1 measure reaching 91.5%. The pruned model is only 1.4MB in size and processes images in an average of 8 milliseconds, thereby laying the groundwork for the creation of portable fruit thinning devices. Ma et al. [19] proposed a lightweight model aimed at detecting and counting small apples, where skip connections were added to the shallow layers of YOLOv7-tiny. P2BiFPN was utilized to fuse and recycle multi-scale features, and a compact ULSAM attention module was introduced to boost both feature retention and the recognition capability for small-sized targets. These research efforts underscore the significant promise of deep learning techniques in the domain of fruit recognition, with advantages in addressing complex environments, handling diverse targets, and improving detection accuracy. Through accurate fruit detection, the automation level of orchard management can be significantly improved, promoting the development of intelligent agricultural equipment and providing strong support for precision agriculture, automated fruit thinning, and large-scale orchard management.

With the deepening application of deep learning technology in the field of fruit recognition, many studies have made remarkable progress through innovative methods and models to address detection challenges such as similar target and background colors, fruit overlap, and occlusion by branches and leaves. For example, Lu et al. [18] proposed a detection head tailored for recognizing early-stage small fruits, utilizing rich semantic cues present in the highest-level feature representation to pinpoint vague targets and gradually transmit this information to deeper layers, thereby refining feature localization and enhancement step by step. Liu et al. [19] designed a single-stage detection model that accurately detects and segments occluded green fruits by replacing the FPN in FCOS with an RFPN, thereby improving the detection of green fruits across different sizes. Zhang et al. [20] integrated an attention mechanism into YOLOv5's feature extraction network, enabling the model to better highlight green apple features, thus improving its performance in detecting green apples in backgrounds with similar colors. Sun et al. [21] proposed an enhanced RetinaNet-PVTv2 model (GHFormer-Net) for identifying small green apples and crabapples under low-light conditions, leveraging the global receptive field of the Transformer to capture feature information. Sun et al. [22] proposed a balanced feature pyramid network (BFP Net) aimed at addressing the difficulty of identifying small and immature green fruits within intricate orchard environments, especially focusing on problems such as background interference and the diminutive size of the fruits. Zhao et al. [23] introduced an optimized model based on the FCOS (Fully Convolutional One-Stage Detector), integrating the LSC multidimensional attention mechanism, and adopted an enhanced ResNet50 architecture to construct the core of the feature extractor, along with an improved sample selection strategy to accurately identify and locate green fruits under challenges such as overlap, illumination variation, and camera angles. These studies have made significant progress in addressing the complex challenges of fruit recognition and established a theoretical basis for identifying green fruits during the thinning process.

To enhance the detection precision of green fruits in challenging orchard environments and meet the efficiency demands of orchard thinning robots, this paper presents the CIDA-Net model based on YOLOv9s for detecting fruits during the thinning stage. This research highlights the following essential contributions:

(1) To strengthen the model's capability in extracting edge and shape features of green fruits, DySnakeConv is incorporated into the backbone network, allowing for better differentiation between green fruits and the background, thus enhancing recognition performance in complex environments.

(2) In the neck network, CAFAFE is used for upsampling, and contextual information is refined by dynamically reconfiguring the feature maps, which enhances the expressive power of the feature maps while preserving spatial resolution.

(3) To optimize the bounding box regression loss, the Inner-IoU loss method is adopted, utilizing auxiliary bounding boxes of varying scales to calculate the loss and adaptively adjust samples at different IoU thresholds, thereby improving the localization accuracy of the target fruit.

II. MATERIALS AND METHODS

A. Green Fruit Dataset

The study focuses on images of green fruits during the thinning period, as they are often similar in color to the background and prone to occlusion by branches and leaves, significantly increasing the complexity of recognition. Such factors can readily result in false positives and undetected green fruits, presenting substantial challenges for accurate identification.

Dataset Collection

The location of green fruit data image collection is Zhangjiazhuang Village, Yuexhuang Town, Yiyuan

County, Zibo City, Shandong Province (118°29'N, 36°23'E). The collection period is from late April 2024 to early May 2024, and the capturing time is from 6:00 to 22:00. The capturing device is the HUAWEI Nova7 smartphone, and the images are saved in "JPG" format.

All data and images are collected in the natural environment of the apple orchard, with backgrounds such as the sky and soil. Green fruit images are captured at different times, under varying illumination, from various angles, and in diverse environments to enhance data variety and strengthen the model's resilience during training. As the images are collected randomly in an unstructured orchard environment, photos with substantial fruit overlap and blurriness are discarded, leaving a total of 1000 images.

The images collected are shown in Fig. 1, and Fig. 1 1a to 1d display green fruit images under different lighting conditions, including natural light (front light and backlight) during the day and LED light at night. Figure 1e and 1g display the green fruit images taken from various shooting angles, which accurately mimic the perspective of the fruit thinning robot in a natural orchard setting. To enhance the model's stability and accuracy in challenging environments, images of occluded and overlapping green fruits are gathered and displayed in Figures 1h to 1i.

Dataset Creation

Given that the existing green fruit images are mainly used for target detection tasks, they lack the object location boxes and category labels required for detection. To solve this problem, LabelImg software labels the green fruits in the dataset with the label "prethion_apple". The annotation data of the image is stored as an XML file, which includes the positional details of the green fruit and its associated label information. Finally, through a normalization operation, the XML file is converted to a TXT file, and the dataset is converted to YOLO format, providing standardized input data for the model.

a Daytime prethin apple Image
b Frontlighting prethin apple image
c Backlighting prethin apple image
d Nighttime prethin apple image
e Overhead shot of prethin apple image
f Close-up image of prethin apple image
g Distant view prethin apple image
h Block the prethin apple image
i Overlapping prethin apple image

Fig. 1. Green apple images in the fruit thinning stage under different environments

To effectively train and evaluate the model, the dataset is split into 70% for training and 30% for validation, with 700 images allocated to the training set and 300 images to the validation set. By employing this data partitioning strategy, the model can thoroughly capture the data characteristics during training and receive an unbiased performance evaluation in the validation phase, thus enhancing its stability and reliability for practical use.

B. CIDA-Net Young Fruit Detection

In deep neural networks, the original information of input data is gradually lost after multi-layer feature extraction and spatial transformation, which impairs the model's capacity to effectively utilize advanced semantic information for accurate target detection and classification, thereby creating an information bottleneck. [26] Aiming at the above problems, YOLOv9 [27] proposes a systematic solution through innovative optimization in the training mechanism and network architecture design. The transfer of gradient information is optimized by introducing the Programmable Gradient Information (PGI) mechanism. By assisting with reversible branches and multi-level auxiliary information, PGI makes the gradient information more complete and reliable, alleviating the issues of gradient vanishing or explosion in deep networks. A Generalized Efficient Layer Aggregation Network (GELAN) is introduced to optimize the feature fusion process through efficient hierarchical aggregation. GELAN enhances the network's capability to capture intricate features, minimizes information loss, and boosts detection precision as well as the overall performance of the model. Through these two innovative designs, YOLOv9 enhances target detection performance while also offering a more efficient and stable solution for the training and inference processes of deep neural networks, effectively handling detection tasks in complex scenarios. YOLOv9s is better suited for real-world deployment environments, maintaining high detection accuracy, minimizing computational overhead, and enhancing operational efficiency. This study uses the YOLOv9s object detection model as the foundational framework for detecting green fruits during the fruit thinning process. Additionally, the CIDA-Net model for detecting young fruits is introduced, and its structural overview is shown in Fig. 2.

The backbone network employs AConv and DySnakeConv modules for feature extraction. The neck network is enhanced through multi-layer convolution and upsampling modules, while the introduction of the CARAFE module improves the accuracy and efficiency of feature recombination. In the detection head, the integration of InnerIoU loss strengthens bounding box refinement, thereby improving the model's overall performance in complex environments.

Target Shape Adaptation Based on DySnakeConv

This study conducts a thorough investigation of YOLOv9s, particularly the RepNCSPELAN4 module. Although this module excels in extracting features and combining information across multiple scales, it demands a substantial number of parameters and results in increased computational overhead. Therefore, this study integrates Dynamic Snake Convolution (DySnakeConv) [28] into the backbone to enhance processing efficiency and reduce model complexity, as shown in Fig. 3. The module adaptively adjusts the convolutional kernel's field of view by applying deformable offsets to better align with the shapes of different targets. Unlike traditional convolution operations, DySnakeConv not only performs standard local perception within the convolution kernel but also enables more refined feature extraction according to the morphological properties

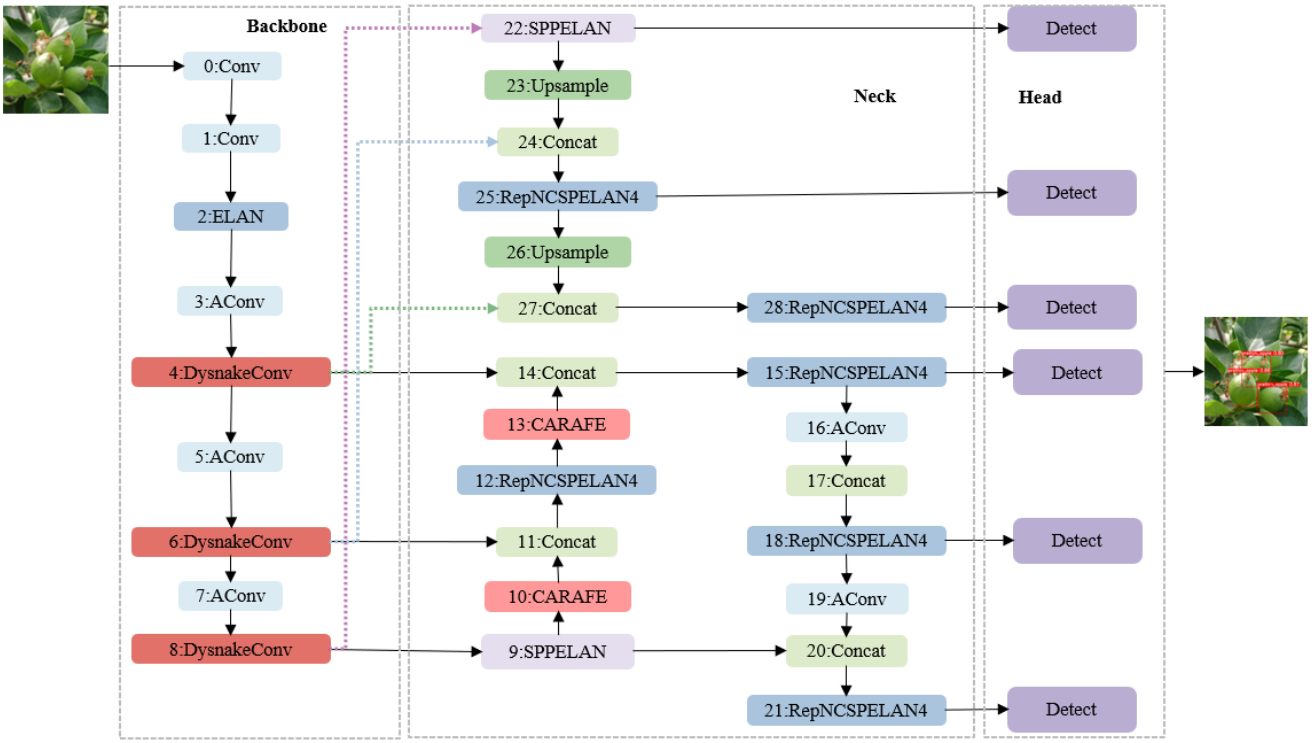


Fig. 2. Overall architecture of the CIDA-Net model

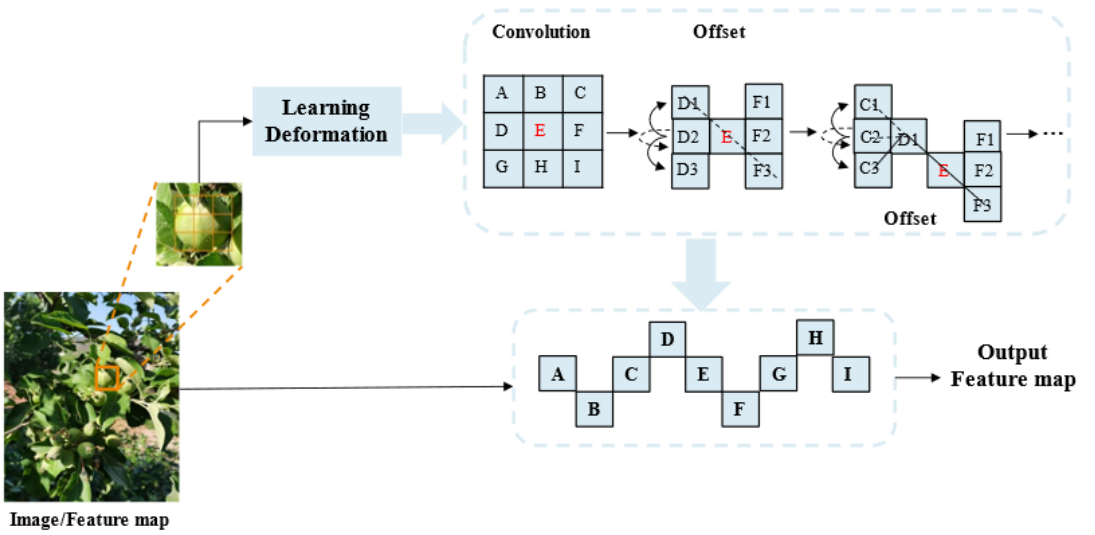


Fig.3. Diagram of Dynamic Snake Convolution

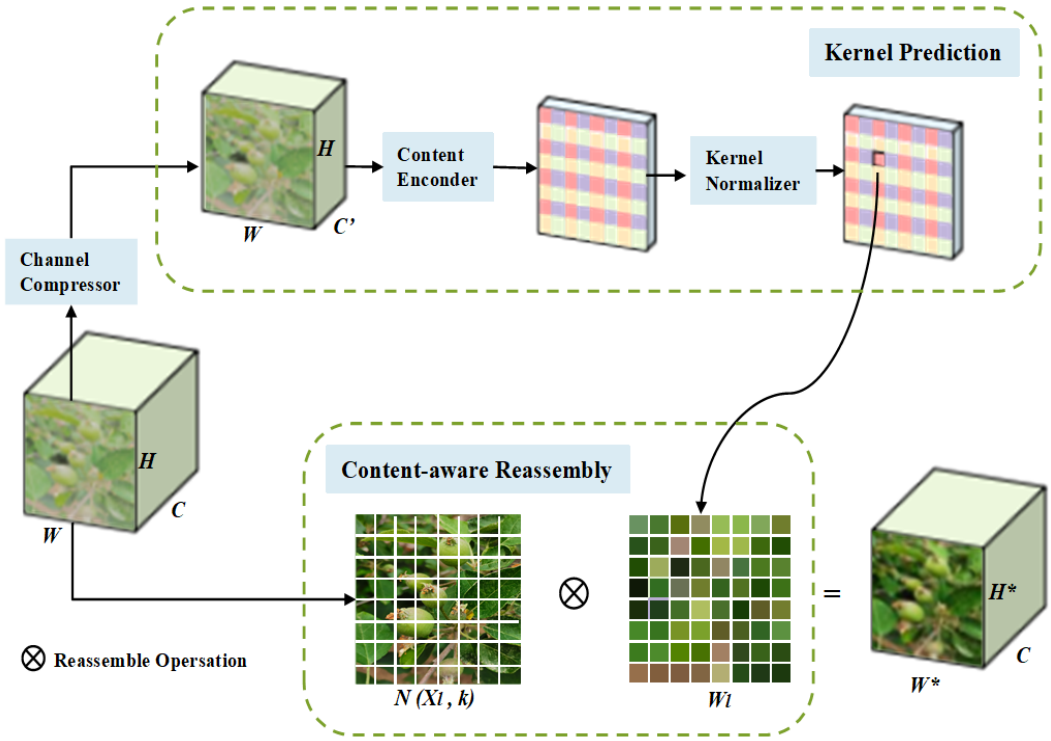


Fig. 4. Overall framework of CARAFE.

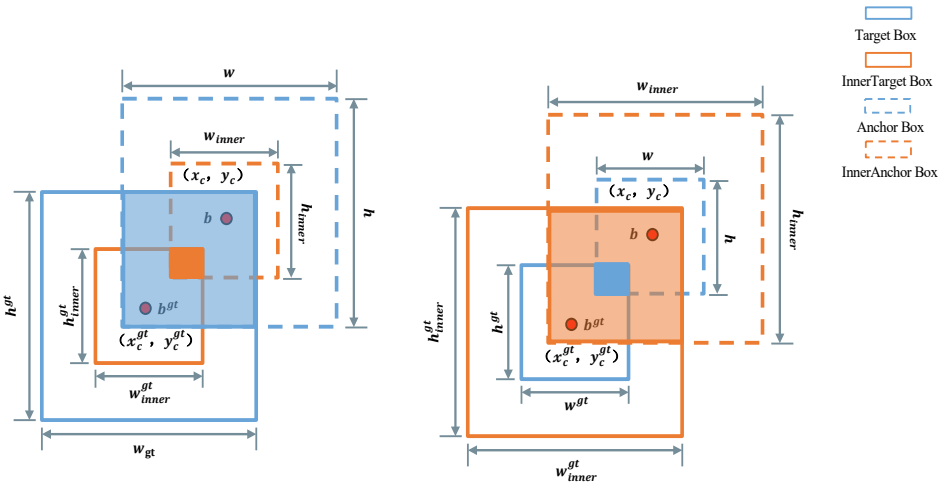


Fig.5. Schematic of the InnerIoU calculation process

of the target. DySnakeConv gradually adjusts the position of the convolution kernel in different directions through the accumulated deformation offsets. For instance, considering x as an example, (x_i, y_i) represents the center position of the convolution kernel, $K_i \pm c = (x_i \pm c, y_i \pm c)$ represents the position of each element in the convolution kernel K, c is the deformation offset, $\Sigma_i^{i+c}\Delta y$ denotes the accumulated offset along the y from the center point y_j to the current position being processed by the convolution kernel, and similarly, $\Sigma_{i-c}^{i}\Delta y$ represents the backward-calculated offset. The equation representing this process is given below:

$$K_{i\pm c} = \begin{cases} (x_{i+c}, y_{i+c}) = \left(x_i + c, y_i + \Sigma_i^{i+c}\Delta y\right) \\ (x_{i-c}, y_{i-c}) = \left(x_i - c, y_i + \Sigma_{i-c}^{i}\Delta y\right) \end{cases} \quad (1)$$

Similarly, the offset of the convolution kernel along the y direction is computed as:

$$K_{j\pm c} = \begin{cases} \left(x_{j+c}, y_{j+c}\right) = \left(x_j + \Sigma_j^{j+c}\Delta x, y_j + c\right) \\ (x_{j-c}, y_{j-c}) = (x_j + \Sigma_{j-c}^{j}\Delta x, y_j - c) \end{cases} \quad (2)$$

Deformation offsets are typically small values. To accurately calculate the feature values at each position in the convolution kernel, bilinear interpolation is used to handle non-integer positions. Bilinear interpolation computes the interpolation at the target position by performing a weighted average from the four neighboring pixel positions, and the calculation process is represented by the following formula:

$$K = \Sigma_{K'} B(K', K) \times K' \quad (3)$$

Where $B(K, K')$ represents the bilinear interpolation kernel, which is weighted based on the relative shift between the input and target positions to ensure the precision of the interpolation outcomes. Moreover, to streamline the calculation and enhance efficiency, the bilinear interpolation kernel is calculated separately along the horizontal and vertical axes:

$$B(K, K') = b(K_x, K_x') \times b(K_y, K_y') \quad (4)$$

By introducing the Dynamic Snake Convolution module, this study effectively enhances the target detection capability of YOLOv9s in complex environments. DySnakeConv adaptively modifies the convolution kernel's receptive field to accommodate targets with varying geometric shapes, particularly irregularly shaped green fruits impacted by occlusion and overlap, allowing for more accurate detection of the target's edges and details. By introducing deformation offsets and combining bilinear interpolation methods, the module performs accurate feature extraction across different scales, showing significant advantages in handling fruit overlap and occlusion issues.

Content-Aware Feature Upsampling Based on CARAFE

To enhance both precision and efficiency in feature recombination, the CARAFE module [29] is introduced in the neck network to better preserve detail information during the upsampling process and enhance the feature map's expressive power. Traditional interpolation methods (such as bilinear interpolation and nearest neighbor interpolation) [30] mainly rely on positional information for feature reconstruction, which often leads to the loss of feature details. In contrast to traditional methods, the CARAFE module employs a content-aware mechanism that dynamically adjusts the recombination strategy based on the semantic content embedded in the feature map. Specifically, the CARAFE module uses adaptive convolution kernels to recombine features and finely adjust them for different regions, preserving more local detail information. Therefore, the CARAFE module enhances the effectiveness of feature fusion and further boosts the model's accuracy in object detection and recognition.

CARAFE mainly comprises an upsampling convolution kernel prediction module and a feature rearrangement unit, and its structural design is shown in Fig. 4. CARAFE processes computation in two distinct stages. The process begins by constructing reorganization kernels tailored to each target point, leveraging adjacent local feature information. Specifically, the reorganization convolution kernel is dynamically adjusted based on neighborhood data, enabling the feature map to more accurately represent the semantic information within the target region, thereby optimizing the upsampling process, as shown below:

$$W_l = \psi(N(X_l, k_{encoder})) \quad (5)$$

In this case, W_l denotes the convolution kernel predicted for the target position l, $N(X_l, k_{encoder})$ represents the $k_{encoder} \times k_{encoder}$ neighborhood centered at position l in the input feature map X, and ψ denotes the function used to predict the convolution kernel.

The second step of the feature reorganization module employs the predicted convolution kernel to restructure the feature map. The features at each target location are fused with those in its neighborhood through weighted summation to form a new upsampled feature map. The process is as follows:

$$X_l = \phi(N(X_l, k), W_l) \quad (6)$$

In this case, ϕ is the content-aware reorganization function, which combines the local region features $N(X_l, k)$ with the predicted convolution kernel W_l to generate the final features for the target position.

CARAFE not only boosts the transformation efficiency from coarse to fine feature maps, but also significantly refines their quality, resulting in more accurate and detailed high-resolution representations that are better
adapted for object detection and recognition in complex environments. Through this adaptive upsampling method, CARAFE effectively boosts the model's effectiveness in multi-scale target detection, particularly in green fruit recognition tasks, where it can more accurately capture the target's details, thereby enhancing detection precision and robustness.

Optimizing Bounding Box Regression Based on InnerIoU Loss

The localization process in object detection models heavily depends on bounding box regression loss, which measures the difference between the predicted and actual bounding boxes. A common approach for bounding box regression loss involves using the Intersection over Union (IoU), which measures how well the anchor box aligns with the ground truth by calculating the proportion of their overlapping region to the combined area. The calculation for this is expressed as follows:

$$\mathrm{Io}U = \frac{|B \cap B_{\mathrm{gt}}|}{|B \cup B_{\mathrm{gt}}|} \quad (7)$$

$$L_{IoU} = 1 - IoU \quad (8)$$

Directly using the overlap between the predicted box and the ground truth to measure localization accuracy is simple and

intuitive, making it suitable for various object detection tasks. However, common methods cannot flexibly adjust to changes in the target's scale. They converge quickly for high IoU samples but more slowly for low IoU samples, which in turn affects detection accuracy. Therefore, to address these shortcomings, this study introduces the InnerIoU loss [31] calculation method in the localization branch, as shown in Fig. 5. This method regulates the size of the supplementary bounding box by introducing a supporting box and a dynamic adjustment factor. When the IoU of the predicted box is high, a smaller auxiliary box is used to accelerate convergence; When the IoU is low, a bigger supplementary box is used to enhance the regression results. The locations of the supplementary bounding box are calculated by adjusting the center coordinates and size of the bounding box. The central positions

of the ground truth box and its inner bounding box are represented as（x_c^{gt}， y_c^{gt}）, whereas those of the anchor box and its corresponding inner box are indicated as（x_c， y_c）. The ground truth box's width and height are represented as w_{gt} and h_{gt}, respectively, while the width and height of the anchor box are denoted as w and h. ratio represents the scaling factor, with a value range between [0.5, 1.5].

Determining the dimensions of the inner bounding box for the ground truth box:

$$b_l^{gt} = x_c^{gt} - \frac{w_{\mathrm{gt}}}{2} \times \mathrm{ratio}，\ b_r^{gt} = x_c^{gt} + \frac{w_{\mathrm{gt}}}{2} \times \mathrm{ratio} \tag{9}$$

$$b_t^{gt} = y_c^{gt} - \frac{h_{\mathrm{gt}}}{2} \times \mathrm{ratio}，\ b_b^{gt} = y_c^{gt} + \frac{h_{\mathrm{gt}}}{2} \times \mathrm{ratio} \tag{10}$$

Determining the dimensions of the inner bounding box for the anchor box:

$$\mathrm{b_l} = \mathrm{x_c} - \frac{\mathrm{w}}{2} \times \mathrm{ratio}，\ \mathrm{b_r} = \mathrm{x_c} + \frac{\mathrm{w}}{2} \times \mathrm{ratio} \tag{11}$$

$$\mathrm{b_t} = \mathrm{y_c} - \frac{\mathrm{h}}{2} \times \mathrm{ratio}，\ \mathrm{b_b} = \mathrm{x_c} + \frac{\mathrm{h}}{2} \times \mathrm{ratio} \tag{12}$$

The intersection (inter) and union (union) areas are computed using the edge coordinates of the internal bounding boxes corresponding to both the ground truth and anchor boxes, after which the IoU (Intersection over Union) is derived to assess the degree of overlap between these inner boxes. The formula for this calculation is as follows:

$$\mathrm{inter}=(min(b_r^{\mathrm{gt}},b_r)\text{-}max(b_l^{\mathrm{gt}},b_l))\times(min(b_b^{\mathrm{gt}},b_b)\text{-}max(b_t^{\mathrm{gt}},b_t)) \tag{13}$$

$$\mathrm{union} = (\mathrm{w_{gt}} \times \mathrm{h_{gt}} \times (\mathrm{ratio})^2) + (\mathrm{w} \times \mathrm{h} \times (\mathrm{ratio})^2) - \mathrm{inter} \tag{14}$$

$$\mathrm{IoU_{inner}} = \frac{\mathrm{inter}}{\mathrm{union}} \tag{15}$$

Calculation of InnerIoU loss:

$$\mathrm{L_{Inner\ IoU}} = 1 - \mathrm{IoU_{inner}} \tag{16}$$

Based on retaining the original loss function structure, the InnerIoU loss calculation method is adopted, introducing auxiliary bounding boxes and dynamic scaling factors to improve the loss computation. This method adaptively modifies the dimensions of the auxiliary box, thereby accelerating the convergence process of the model, effectively mitigating the challenges of fruit overlap and occlusion, enhancing the precision of green fruit localization, and optimizing overall detection performance.

III. Results and Analysis

In order to evaluate the performance of the CIDA-Net network model in detecting green fruits during the thinning period, we first provide a detailed description of the experimental setup, evaluation criteria, and the procedures followed during testing. The model is then trained using the green fruit dataset from the thinning period, and its performance is assessed on the test set using the best-performing configuration. Finally, under identical experimental settings, the CIDA-Net model is benchmarked against leading object detection models, and the results are visualized. Through comparative analysis, the model's superiority in detecting green fruits is demonstrated.

A. Experimental Setup and Model Optimization Strategies

The experimental environment in this paper is based on the Ubuntu 18.04 64-bit system, with the deep learning framework being Pytorch. The GPU used for the experiments is a 24GB NVIDIA A30, and the CUDA version is 11.4. All models are implemented using Python 3.10.14 and Pytorch 1.13.During the model training process, the initial learning rate, momentum, and weight decay are set to 0.01, 0.937, and 0.0005, respectively, with Stochastic Gradient Descent (SGD) used as the optimization algorithm. The training model is set to 80 epochs, with a batch size of 4. The training dynamics and detection performance of the model are further illustrated in Fig. 6, including the learning rate schedule and the precision-recall curve.

B. Evaluation Metrics

Since the model is required to accurately predict the outcomes of green fruit detection, precision and recall are adopted as effective evaluation metrics. Precision is calculated using formula (17), while recall is computed using formula (18):

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{17}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{18}$$

In this equation, True Positives (TP) refer to the green fruits correctly detected by the model, where the Intersection over Union (IoU) between the detection box and the ground truth box exceeds the specified threshold (IoU_threshold); False Negatives (FN) refer to the green fruits missed by the model. To comprehensively assess the model, the accuracy of individual-category predictions under different thresholds is calculated via formula (19), and the average recall is computed using formula (20):

$$\mathrm{AP_{IoU=i}} = \frac{1}{101}\sum_{\mathrm{r\in Recall}} \mathrm{Precision\ (r)} \tag{19}$$

$$\mathrm{AR} = \frac{1}{\mathrm{T}}\sum_{\mathrm{i=1}}^{\mathrm{T}} \mathrm{Recall_{IoU_i}} \tag{20}$$

The AP value is calculated based on a specific IoU threshold, which is set within the range of 0 to 1. In this study, multiple thresholds within the range of [0.5, 0.95] (with a step size of 0.05) are used as the metric to evaluate the model's detection accuracy. $\sum_{\mathrm{r\in Recall}} \mathrm{Precision\ (r)}$ refers to the summation of precision at different recall values, and $\frac{1}{101}$ refers to the average precision across different recall rates, that is, the precision calculated at 101 different recall values within the range of [0, 1].

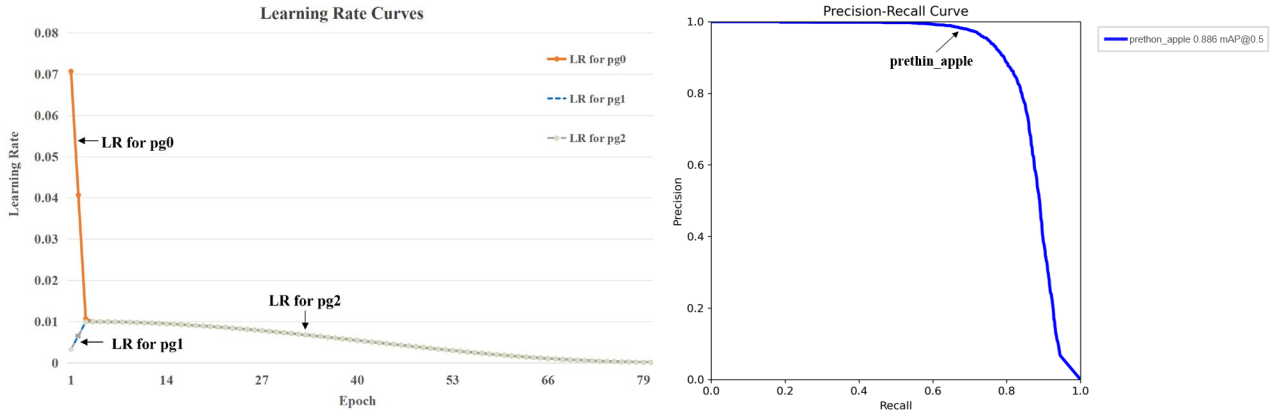


Fig. 6. Left: Learning rate change during training. Right: Precision-recall curve.

TABLE I
COMPARISON OF ABLATION EXPERIMENT IMPROVEMENTS

Baseline	DySnakeConv	CARAFE	Precision	Recall	AP50	AP75	AP50-95	Params/M	GFlops
Yolov9s	-	-	90.7	76.1	85.8	70.6	61.3	9.7	39.6
	√	-	90.6	78.6	87.1	71.6	62.2	8.8	35.8
	-	√	92.8	77.5	87.2	71.9	62.5	9.8	39.9
	√	√	93.3	77.5	88.1	73.6	64.0	8.9	36.0
Yolov9s + InnerIoU	-	-	91.7	77.0	86.5	72.8	63.2	9.7	39.6
	√	-	90.5	78.7	88.0	74.0	64.3	8.8	35.8
	-	√	92.4	77.6	87.4	72.8	63.4	9.8	39.9
	√	√	91.4	79.7	88.6	74.7	65.0	8.9	36.0

C. *Ablation experiment*

In order to evaluate how DySnakeConv, CARAFE, and InnerIoU influence model performance, this study conducted a set of ablation experiments aimed at comprehensively validating the improvements introduced by these components. In the experiment, DySnakeConv and CARAFE were successively integrated into the YOLOv9s base model, and the performance of green fruit detection before and after adding different modules was compared with different loss calculation methods to evaluate their improvements. To guarantee the fairness and validity of the experiment, the models were maintained with consistent experimental conditions and hyperparameter configurations. The outcomes obtained from testing on the green fruit dataset during the thinning stage are summarized in Table I.

As shown in the table, with the integration of the DySnakeConv module into the base model, compared to the original version, the model's AP50, AP75, and AP50-95 improve by 1.3, 1.0, and 0.9 percentage points, respectively, while the number of parameters decreases by 0.9M and the computational complexity is reduced by 3.8GFlops. When the CARAFE module is integrated into the base model, in comparison to the original model, the number of parameters and computational demands increase, but AP50, AP75, and AP50-95 improve by 1.4, 1.3, and 1.2 percentage points, respectively. The recall rate decreases, but precision improves. When both DySnakeConv and CARAFE modules are incorporated into the base model, in comparison to the original model, precision and recall show improvements of 2.6 and 1.4 percentage points, respectively, while AP50, AP75, and AP50-95 increase by 2.3, 3.0, and 2.7 percentage points, respectively. The parameter count reduces by 0.8M, and the computational load decreases by 3.6GFlops. The introduction of the InnerIoU loss method into the base model results in improvements of 1.0, 0.9, 0.7, 2.2, and 1.9 percentage points for P, R, AP50, AP75, and AP50-95, respectively. When the InnerIoU loss method is introduced on top of DySnakeConv, compared to using only DySnakeConv, AP50, AP75, and AP50-95 increase by 0.9, 2.4, and 2.1 percentage points, respectively. When the InnerIoU loss method is introduced on top of CARAFE, compared to using only CARAFE, AP50, AP75, and AP50-95 increase by 0.2, 0.9, and 0.9 percentage points, respectively. Finally, when the InnerIoU loss method is introduced with both DySnakeConv and CARAFE, compared to using both modules alone, AP50, AP75, and AP50-95 increase by 0.5, 1.1, and 1.0 percentage points, respectively.

In comparison to the baseline model, P, R, AP50, AP75, and AP50-95 showed improvements of 0.7, 3.6, 2.8, 4.1, and 3.7 percentage points, respectively. The model saw a reduction of 0.8M in parameters and a decrease of 3.6GFlops in computational complexity. The experimental results show that the enhanced approach can notably improve the model's detection performance in challenging environments, while simultaneously reducing its complexity, thus confirming the efficacy of the multi-module integration optimization strategy.

D. Comparative Experiments

To further evaluate the effectiveness of the algorithm, this study conducts comparative experiments by evaluating the optimized model against several advanced and representative object detection models on the green fruit dataset during the fruit thinning period. A total of 11 models are selected for comparison, including classical representative models such as FasterRCNN [32], Dino [33], DDQ [34], RT-DETR [35], as well as mainstream YOLO series models: YOLOv6 [36], YOLOv8, YOLOv9, YOLOv10 [37], YOLOv11 [38], YOLOv12 [39], and the proposed CIDA-Net. Accuracy, recall rate, F1 score, AP50, AP75, AP50-95, model complexity, and computational performance are selected as evaluation metrics for the comparative experiments. The results of the comparison experiments are summarized in Tables Ⅱ and Ⅲ.

The table presents the performance of various object detection models on the green fruit dataset during the fruit thinning period. The proposed CIDA-Net model demonstrates significant improvements across multiple core evaluation metrics. Based on YOLOv9s, the optimized CIDA-Net achieves AP50, AP75, and AP50-95 scores of 88.6, 74.7, and 65.0, respectively, indicating stable and superior detection performance under different confidence thresholds. Compared to Faster R-CNN, CIDA-Net improves by 3.5, 13.8, and 10.7 percentage points; compared to DINO, by 11.5, 13.1, and 12.2 percentage points; compared to DDQ, by 2.4, 5.8, and 6.2 percentage points; and compared to RT-DETR, by 4.4, 8.9, and 6.5 percentage points. These results demonstrate that, in comparison with high-performance models such as DDQ and RT-DETR, CIDA-Net achieves a more balanced performance in terms of both accuracy and detection stability. Within the YOLO series, CIDA-Net also shows a clear performance advantage. Compared to lightweight models such as YOLOv6, YOLOv8n, and YOLOv12, it achieves an average increase of over 8 percentage points in the AP50-95 metric. Even when compared to more powerful models such as YOLOv9s, YOLOv10, and YOLOv11, CIDA-Net still achieves noticeable gains in precision, particularly excelling in comprehensive performance metrics such as F1 score, recall rate, and average recall. While YOLOv10 to YOLOv12 offer advantages in lightweight design, CIDA-Net maintains a

TABLE II
THE DETECTION RESULTS OF EACH DETECTION MODEL

Model	P	R	F1	$AR_{maxDet=100}$ /%
FasterRCNN	67.8	87.4	76.3	60.9
Dino	87.8	77.2	82.2	68.9
DDQ	97.3	69.5	81.1	68.7
RT-DETR	90.4	75.6	82.3	75.6
Yolov6	89.8	72.8	80.4	72.7
Yolov8n	88.4	72.8	79.9	72.8
Yolov9s	90.7	76.1	82.8	76.1
YOLOv10	91.0	78.8	84.6	78.8
YOLOv11	90.4	75.0	81.8	75.1
YOLOv12	89.4	70.0	78.5	70.4
Ours	91.4	79.7	85.2	80.0

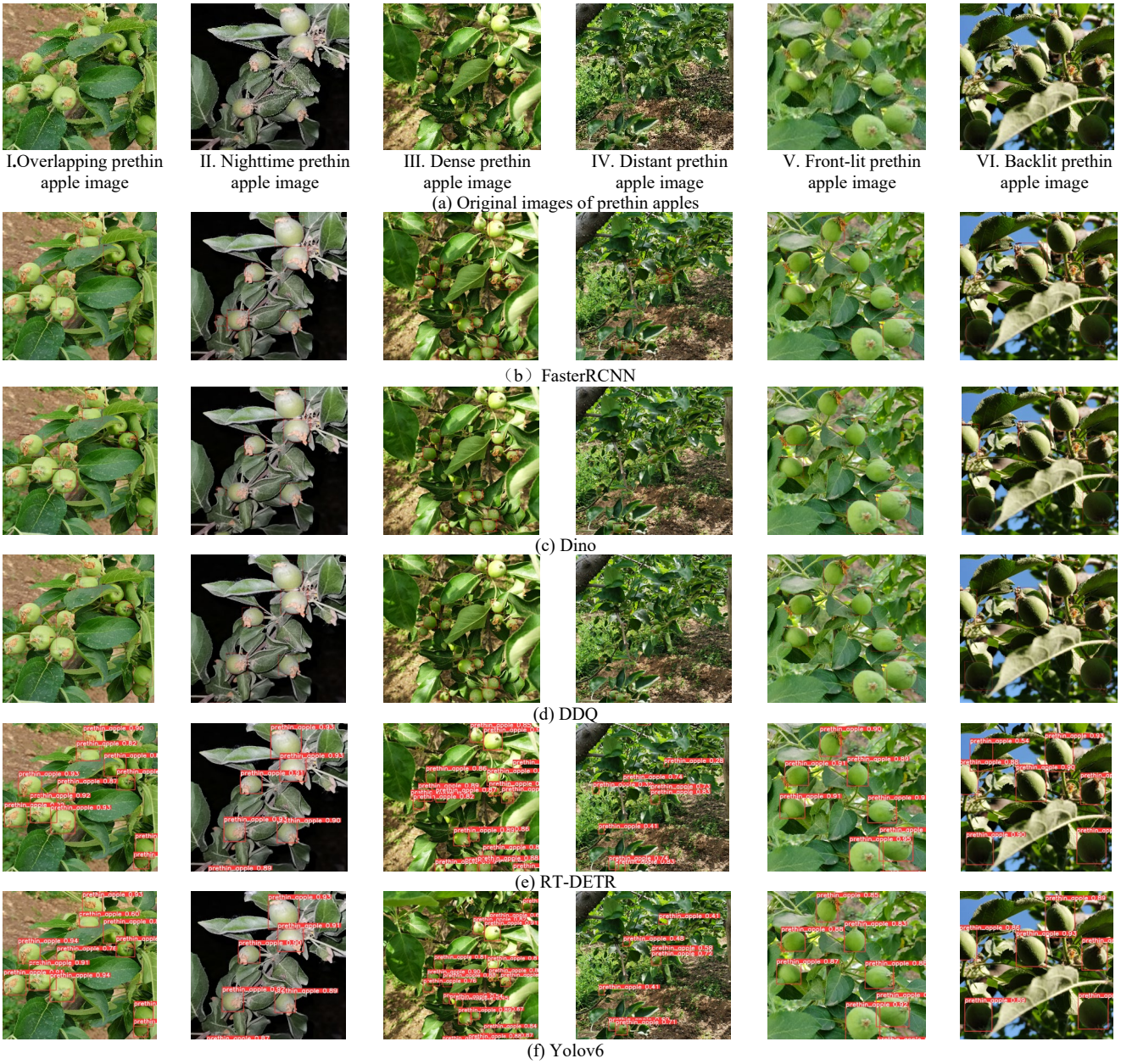
TABLE III
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS FOR GREEN FRUIT DETECTION

Model	AP_{50}	AP_{75}	AP_{50-95}	Params/M	GFlops
FasterRCNN	85.1	60.9	54.3	41.3	174
Dino	77.1	61.6	52.8	47.7	235
DDQ	86.2	68.9	58.8	48.3	119
RT-DETR	84.2	65.8	58.5	32.8	108
Yolov6	82.3	65.7	57.6	4.2	11.9
Yolov8n	81.9	66.0	57.3	3.1	8.1
Yolov9s	85.8	70.6	61.3	9.7	39.6
YOLOv10	87.0	73.1	63.1	2.7	8.4
YOLOv11	83.6	66.5	58.7	2.6	6.4
YOLOv12	80.8	64.2	56.1	2.5	6.0
Ours	88.6	74.7	65.0	8.9	36.0

reasonable model size while achieving higher detection accuracy and better boundary fitting. It also exhibits enhanced detection stability in complex scenarios involving blurred boundaries, dense fruit clusters, and irregular fruit shapes. In summary, CIDA-Net achieves a well-balanced trade-off between accuracy, stability, and model complexity. It effectively meets the high-precision requirements of green fruit detection during the fruit thinning stage and demonstrates strong potential for deployment and practical applications.

To offer a clearer comparison of the detection performance across different algorithms on the green fruit dataset, this paper performs comparative experiments under various lighting conditions (including daytime and nighttime) and visualizes the results. The visualization results are shown in Fig. 7. In complex environments, such as scenes with branch and leaf occlusion or multiple overlapping targets, other algorithms struggle with failing to detect targets and generating incorrect detections. In contrast, the improved CIDA-Net model handles the challenges posed by occlusion and target overlap more effectively. The experimental results demonstrate that this model efficiently addresses the complexities and dynamic changes in orchard environments, resulting in a notable improvement in green fruit detection accuracy, while also exhibiting robust performance and adaptability.

The green fruit detection model proposed in this study effectively identifies and locates targets, addressing issues such as branch and leaf occlusion and fruit overlap. However, when fruit size is small and heavily obstructed by branches and leaves, the model struggles to accurately capture complete target information, as shown in Fig. 8, where missed fruits are highlighted with triangles. Because of the resemblance between the fruit and the surrounding background, coupled with low contrast, the model exhibits lower sensitivity in detecting small fruits, leading to missed detections. Therefore, the model still has limitations in detecting small targets under occlusion scenarios. To overcome these limitations, future research could concentrate on refining feature extraction techniques in occluded environments and improving the model's robustness and precision in detecting small targets.



I.Overlapping prethin apple image | II. Nighttime prethin apple image | III. Dense prethin apple image | IV. Distant prethin apple image | V. Front-lit prethin apple image | VI. Backlit prethin apple image

(a) Original images of prethin apples

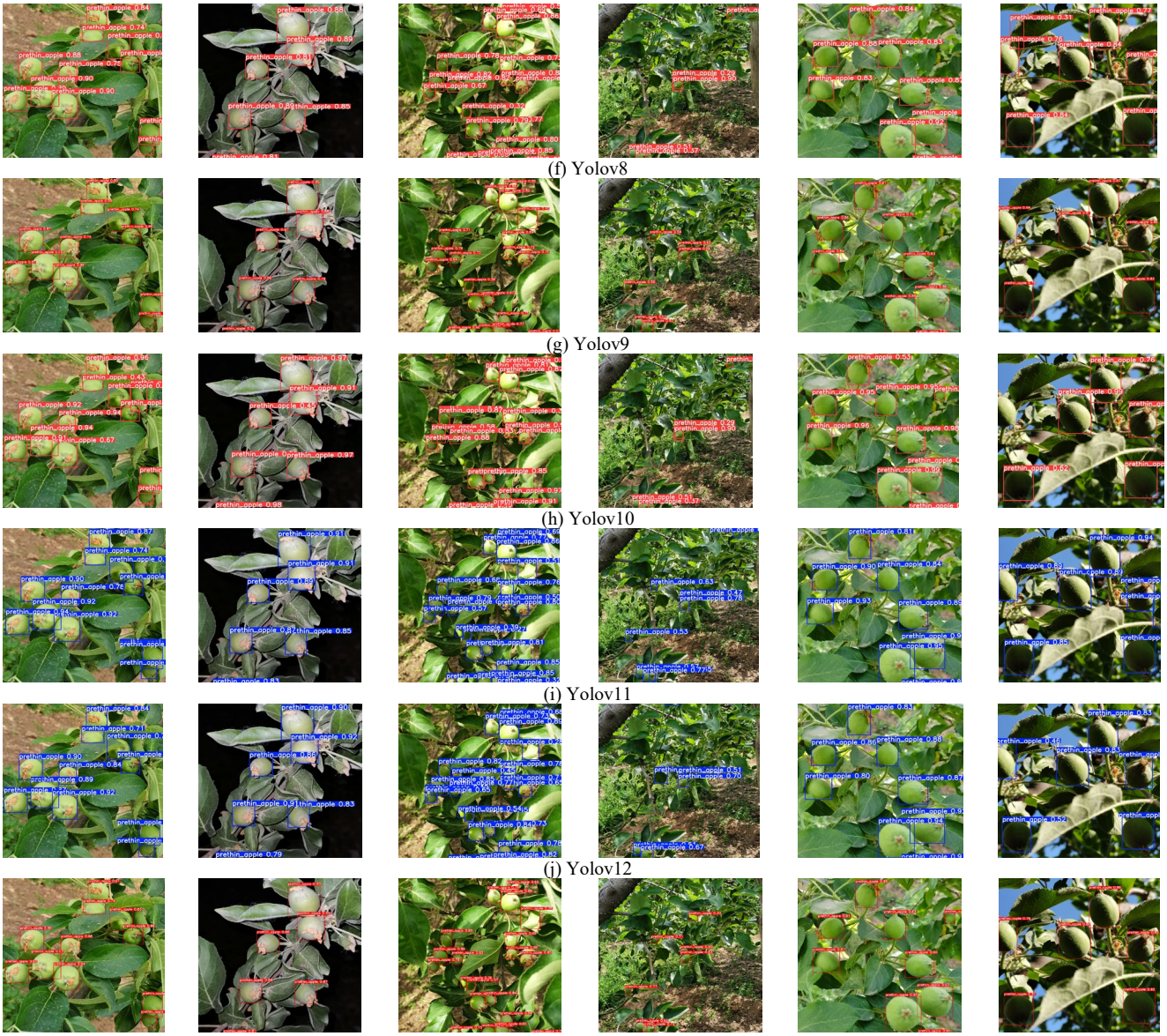
（b）FasterRCNN

(c) Dino

(d) DDQ

(e) RT-DETR

(f) Yolov6



(f) Yolov8

(g) Yolov9

(h) Yolov10

(i) Yolov11

(j) Yolov12

(k) Ours

Fig.7. Visual Comparison of Experimental Results

Fig. 8. Visualization of CIDA-Net Detection Results.

IV. Conclusion

The precision, stability, and real-time performance of detecting green fruits are vital for advancing fruit thinning equipment. This study proposes and validates a green fruit detection model based on YOLOv9s, addressing the fruit recognition problem in complex orchard environments by incorporating multiple optimization strategies. By introducing the serpentine dynamic convolution module, the model enhances its ability to capture edge features, improving detection accuracy for fruits affected by occlusion and overlap. Additionally, the CARAFE module is employed to refine the upsampling procedure, further enhancing the model's ability to perceive fine features. The InnerIoU loss function is incorporated into the detection head's location branch, enabling the model to dynamically optimize the bounding box regression, better aligning with irregular target boundaries and minimizing the creation of unnecessary boxes. Experimental results show that AP50, AP75, and AP50-95 have increased by 2.8%, 4.1%, and 3.7%, respectively, while precision and recall rates have increased by 0.7% and 3.6%, respectively. The model's parameter size and computational complexity have decreased. Additionally, through a comparative analysis with various models, the enhanced YOLOv9s model satisfies the precision criteria for green fruit detection in challenging orchard conditions.

Early detection and accurate identification of green fruits are of great significance for the fruit thinning task in orchard

management. By precisely detecting the fruits, accurate data support can be provided to orchard managers, optimizing the allocation of tree growth space, reducing the cost of manual thinning, and improving fruit quality and yield. Overall, this study not only provides an efficient solution for green fruit detection but also lays a solid foundation for the implementation of intelligent fruit thinning technology in orchard management, offering important technical support for the future automation of orchard management.

REFERENCES

[1] Sun Y, Zhang Y, Jiang Y, et al. Leaf potential productivity at different canopy levels in densely-planted and intermediately-thinned apple orchards. Horticultural Plant Journal, 2016, 2(4): 181-187.

[2] Costa G, Botton A, Vizzotto G. Fruit thinning: Advances and trends. Horticultural reviews, 2018, 46: 185-226.

[3] Iwanami H, Moriya-Tanaka Y, Honda C, et al. Apple thinning strategy based on a model predicting flower-bud formation. Scientia Horticulturae, 2019, 256: 108529.

[4] Lei X, Yuan Q, Xyu T, et al. Technologies and Equipment of Mechanized Blossom Thinning in Orchards: A Review. Agronomy, 2023, 13(11): 2753.

[5] Hou G, Chen H, Jiang M, et al. An overview of the application of machine vision in recognition and localization of fruit and vegetable harvesting robots. Agriculture, 2023, 13(9): 1814.

[6] Gongal A, Amatya S, Karkee M, et al. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture, 2015, 116: 8-19.

[7] Sun S, Jiang M, He D, et al. Recognition of green apples in an orchard environment by combining the GrabCut model and Ncut algorithm. Biosystems Engineering, 2019, 187: 201-213.

[8] Sengupta S, Lee W S. Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. Biosystems Engineering, 2014, 117: 51-61.

[9] Liu X, Zhao D, Jia W, et al. A detection method for apple fruits based on color and shape features. IEEE Access, 2019, 7: 67923-67933.

[10] Jana S, Basak S, Parekh R. Automatic fruit recognition from natural images using color and texture features. 2017 Devices for Integrated Circuit (DevIC). IEEE, 2017: 620-624.

[11] Gill H S, Murugesan G, Mehbodniya A, et al. Fruit type classification using deep learning and feature fusion. Computers and Electronics in Agriculture, 2023, 211: 107990.

[12] Xiao F, Wang H, Xu Y, et al. Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. Agronomy, 2023, 13(6): 1625.

[13] Ru Jiang, Huichuan Duan, Jingyu Yan, and Weikuan Jia, "Green Tomato Segmentation Model Based on Optimized Swin-Unet Algorithm Under Facility Environments," Engineering Letters, vol. 32, no. 11, pp2114-2126, 2024.

[14] Zhao Z Q, Zheng P, Xu S, et al. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 2019, 30(11): 3212-3232.

[15] Zou Z, Chen K, Shi Z, et al. Object detection in 20 years: A survey. Proceedings of the IEEE, 2023, 111(3): 257-276.

[16] Kong X, Li X, Zhu X, et al. Detection model based on improved faster-RCNN in apple orchard environment. Intelligent Systems with Applications, 2024, 21: 200325.

[17] Jia W, Xu Y, Lu Y, et al. An accurate green fruits detection method based on optimized YOLOX-m. Frontiers in Plant Science, 2023, 14: 1187734.

[18] Wang D, He D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. Biosystems Engineering, 2021, 210: 271-281.

[19] Ma L, Zhao L, Wang Z, et al. Detection and counting of small target apples under complicated environments by using improved YOLOv7-tiny. Agronomy, 2023, 13(5): 1419.

[20] Lu Y, Sun M, Guan Y, et al. SOD head: A network for locating small fruits from top to bottom in layers of feature maps. Computers and Electronics in Agriculture, 2023, 212: 108133.

[21] Liu M, Jia W, Wang Z, et al. An accurate detection and segmentation model of obscured green fruits. Computers and Electronics in Agriculture, 2022, 197: 106984.

[22] Weike Zhang, Yanna Zhao, Yujie Guan, Ting Zhang, Qiaolian Liu, and Weikuan Jia, "Green Apple Detection Method Based on Optimized YOLOv5 Under Orchard Environment," Engineering Letters, vol. 31, no.3, pp1104-1113, 2023

[23] Sun M, Xu L, Luo R, et al. GHFormer-Net: Towards more accurate small green apple/begonia fruit detection in the nighttime. Journal of King Saud University-Computer and Information Sciences, 2022, 34(7): 4421-4432.

[24] Sun M, Xu L, Chen X, et al. Bfp net: balanced feature pyramid network for small apple detection in complex orchard environment. Plant Phenomics, 2022.

[25] Zhao R, Guan Y, Lu Y, et al. FCOS-LSC: A novel model for green fruit detection in a complex orchard environment. Plant Phenomics, 2023, 5: 0069.

[26] Tishby N, Zaslavsky N. Deep learning and the information bottleneck principle. IEEE information theory workshop (itw), 2015: 1-5.

[27] Wang C Y, Yeh I H, Mark Liao H Y. Yolov9: Learning what you want to learn using programmable gradient information. European Conference on Computer Vision. Springer, Cham, 2025: 1-21.

[28] Qi Y, He Y, Qi X, et al. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 6070-6079.

[29] Wang J, Chen K, Xu R, et al. Carafe: Content-aware reassembly of features. Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3007-3016.

[30] Parsania P S, Virparia P V. A comparative analysis of image interpolation algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 2016, 5(1): 29-34.

[31] Zhang H, Xu C, Zhang S. Inner-IoU: more effective intersection over union loss with auxiliary bounding box. arXiv preprint arXiv:2311.02877, 2023.

[32] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.

[33] Caron M, Touvron H, Misra I, et al. Emerging properties in self-supervised vision transformers. Proceedings of the IEEE/CVF international conference on computer vision. 2021: 9650-9660.

[34] Zhang S, Wang X, Wang J, et al. Dense distinct query for end-to-end object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7329-7338.

[35] Zhao Y, Lv W, Xu S, et al. Detrs beat yolos on real-time object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 16965-16974.

[36] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976, 2022.

[37] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458, 2024.

[38] Khanam R, Hussain M. Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725, 2024.

[39] Tian Y, Ye Q, Doermann D. Yolov12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524, 2025.