IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 May 2024, accepted 5 June 2024, date of publication 10 June 2024, date of current version 22 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411644

RESEARCH ARTICLE

PeachYOLO: A Lightweight Algorithm for Peach Detection in Complex Orchard Environments

TINGXUAN LI[1], QIYANG CHEN[1], XINYI ZHANG[1], SHAOYUN DING[1], XIANYAO WANG[1], AND JIONG MU[1,2,3]

[1]College of Information Engineering, Sichuan Agricultural University, Ya'an 625000, China
[2]Sichuan Key Laboratory of Agricultural Information Engineering, Ya'an 625000, China
[3]Ya'an Digital Agricultural Engineering Technology Research Center, Ya'an 625599, China

Corresponding author: Jiong Mu (jmu@sicau.edu.cn)

ABSTRACT Precise fruit recognition is crucial for the automated picking of peaches. However, practical implementation encounters challenges, including high costs and low efficiency, which hinder the commercialization of picking robots. To tackle these challenges, this study establishes a synthetic peach dataset and introduces PeachYOLO, an efficient and lightweight model for peach object detection in complex orchard environments. Specifically, based on the lightweight object detection model You Only Look Once version 8 (YOLOv8), this study first replaces traditional convolutions in the detection head structure with Partial Convolution (PConv). This improvement reduces computational and memory requirements while effectively extracting spatial features. Secondly, within the feature output of the neck network, Deformable Convolutional Networks version 2 (DCNv2) is employed in place of traditional convolutions to improve the recognition of irregular targets. Finally, Coordinate Attention (CA) is integrated into the head network to focus precisely on essential image information. Experimental results demonstrate that PeachYOLO achieves a mAP of 93.8%, surpassing the original model by 1.0%. Furthermore, PeachYOLO's computation is only 5.1 FLOPs (G), the number of parameters is 2.6M, and has an inference time of 1.9ms, which is a reduction of 37.0%, 13.6%, and 5.6%, respectively, compared with the original YOLOv8n algorithm. These results underscore the substantial improvements in detection speed, accuracy, and model size offered by PeachYOLO. Moreover, its suitability for peach detection in intricate orchard settings lays the groundwork for the realization of unmanned intelligent peach picking.

INDEX TERMS Deep learning, digital agriculture, lightweight, orchard environment, peach detection, YOLOv8.

I. INTRODUCTION

As one of the oldest cultivated fruits in China, peaches are an important economic pillar for improving farmers' living standards [1]. In recent years, the peach industry in China has witnessed a significant expansion in planting area and production, contributing to its overall development [2]. However, the issue of labor costs in the peach production process is becoming increasingly prominent. Particularly during the harvest season, the extensive work of peach picking is time-consuming, labor-intensive and inefficient [3]. Furthermore, with consumers' increasing demand for the appearance and taste of peaches, the industry's requirements for picking techniques have also increased, which further increases the difficulty and complexity of picking work [4]. With the rapid development of smart agriculture, the emergence of picking robots has effectively addressed the issues of labor shortages, high manual picking costs, and low efficiency.

Accurately detecting peach fruit is crucial for automated picking. However, the biggest challenges in commercializing picking robots are cost and efficiency. Three significant conditions need to be met for the commercialization of agricultural robots: high detection accuracy, fast model inference, and lightweight deployment of models [5]. Consequently,

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

conducting extensive research on peach fruit detection technology is imperative. This will significantly enhance the development of the peach industry, ensure the timely harvesting of ripe fruits, and boost peach market competitiveness.

Orchards present a complex environment due to changes in lighting conditions and fruit overlapping on branches and leaves. Consequently, encountering problems such as missed and false detections in fruit detection tasks is highly probable [6]. Furthermore, the model's size not only tests the configuration and computing power of hardware devices but also influences the model's deployment cost and the efficiency of picking robots [7]. Therefore, the challenge and focus of research lie in achieving lightweight models and improving detection speed while maintaining accurate peach detection.

To tackle these challenges, this study presents a novel approach to peach fruit recognition utilizing the YOLOv8 [8] model, termed the PeachYOLO model. In this study, the main innovations and contributions are as follows:

1) Establish a database of peaches under different lighting and viewing conditions. Diversify the image dataset through data enhancement, which enhances the anti-interference ability under complex conditions.
2) DCNv2 replaces the traditional convolution and improves the ability to recognize targets with different scales and irregularities. The inclusion of CA in the original YOLOv8n network enhances the model's precision in accurately detecting and identifying densely populated peaches.
3) Introducing PConv into the head structure enhances spatial feature extraction efficiency. This reduces redundant computations and memory accesses, achieving lightweight model enhancements. This saves equipment resources and fulfills the peach object detection needs during mechanized harvesting.

The study is systematically structured into six main sections. Section II reviews the related work. Section III outlines the proposed approach. Section IV details the experimental setup, including dataset and implementation specifics. Section V outlines results and discussion. Finally, Section VI summarizes the research undertaken in this study.

II. RELATED WORK

In this section, the development of target detection networks is initially reviewed, followed by a discussion on the development of fruit and vegetable detection techniques.

A. DEVELOPMENT OF OBJECT DETECTION NETWORKS

The categorization of deep learning-based target detection involves two primary groups: two-stage object detection algorithms and one-stage object detection algorithms. In the two-stage approach, the algorithm first generates candidate object locations and then classifies these locations. Several widely used two-stage detection algorithms include Faster R-CNN [9], Mask R-CNN [10], and R-FCN [11] and so on. Although two-stage object detection algorithms demonstrate high accuracy and robust generalization, they may fail to meet real-time requirements due to their slow execution speed and large model size. Compared to two-stage object detection methods, one-stage algorithms like SSD [12], YOLO [13], and RetinaNet [14] offer faster execution speeds and streamlined architectures.

The YOLO series models, in particular, are popular for their balance of speed and accuracy. The YOLOv1-v4 [13], [15], [16], [17] models established the foundation of the YOLO series. They established a single-stage detection framework that includes a backbone network, neck, and head components, and features that utilize multi-scale branching to predict targets of different sizes. This framework enables the YOLO series to pursue efficient and accurate target detection by directly predicting the bounding box and target class through an end-to-end manner. The YOLOv5 [18] algorithm improves performance and accuracy through the application of optimization techniques and data enhancement strategies. YOLOX [19] incorporates multiple positive samples, eliminates anchors, and employs a decoupled head structure into the model structure, thereby establishing a new paradigm in model design for the YOLO series. YOLOv6 [20] represents a further advancement in the YOLO series, building upon the strengths of its predecessors and enhancing performance through the use of larger models, more intricate network architectures, and refined training methodologies. YOLOv7 [21] is constructed based on the foundation of YOLOv6 and achieves enhanced detection accuracy and accelerated detection speeds through further model architecture optimization and training strategy adjustment. Launched officially by Ultralytics in January 2023, YOLOv8 builds upon the core competencies of its predecessors in the YOLO lineage, introducing additional refinements to boost both performance metrics and operational versatility. The backbone and neck of the model use a C2f structure, and different channel numbers have been adjusted for models of different scales. The head section has been changed from a coupled head to a decoupled head [19]. YOLOv9 [22] introduces Programmable Gradient Information (PGI) to address the issue of data loss during transmission. After comprehensively considering the research object and the network's detection accuracy and lightweight requirements, the YOLOv8 network was used as the benchmark model.

B. DEVELOPMENT OF FRUIT AND VEGETABLE DETECTION

In the conventional detection process, the identification of fruits and vegetables is predominantly conducted through manual sensory evaluation, resulting in significant discrepancies in evaluation outcomes, being prone to subjective influences, and necessitating the involvement of a considerable workforce. In contemporary agricultural research, the methods employed for fruit detection have evolved from initially relying on image processing techniques to the use of deep learning-based approaches. The environment of fruit

and vegetable gardens is complex due to variations in light conditions and the overlapping of fruits with branches and leaves.

Deep learning algorithms are powerful in data characterization and feature extraction, making them an effective tool for detecting fruits and vegetables in complex environments such as plantations. The YOLO series, in particular, has gained popularity for its ability to detect fruits and vegetables owing to its flexible structure and rapid and convenient operation.

For instance, Lai et al. [23] developed a system for the real-time detection of ripe fresh fruit bunches on oil palm trees using Yolov4. In the test of oil palm orchards, the system operated at a rate of approximately 21 frames per second (FPS) in real-time and achieved a mAP of 87.9%. An et al. [24] proposed a system based on the YOLOX model, replacing the original CSP block in the backbone network with a self-designed feature extraction module, the C3HB block, to effectively address the issue of low accuracy in monitoring the growth status of strawberry fruits in complex environments. Sapkota et al. [25] proposed a green apple detection scheme based on an enhanced YOLOv8 neural network combined with a geometric shape fitting technique on 3D point cloud data to detect green apples in commercial orchards with an average accuracy of 0.94. Du et al. [26] proposed the DSW-YOLO network model to improve feature extraction from ground-grown strawberries with irregular shapes by incorporating DCNv3. Tang et al. [27] introduced YOLOv7-plum, a novel recognition method. They integrated the Convolutional Block Attention Module (CBAM) into YOLOv7, enhancing the model's ability to focus on crucial information related to plum fruits in complex backgrounds. Zhang et al. [28] introduced a lightweight network based on an improvement of YOLOv5, which reduces the complexity of the model by incorporating a ghost module. This improvement realizes the detection of dragon fruits in complex orchard environments. The above methods presented demonstrate their considerations regarding model accuracy, inference speed, and lightweight, thereby offering valuable references for this study.

III. PROPOSED APPROACH

A. PEACH YOLO

The architectural design of the PeachYOLO network proposed in this study is seen in Figure 1. Firstly, to realize the model's lightweight, this study defines a new detection head, Partial_C_Detect, and introduces PConv [29] into the head structure to enhance spatial feature extraction efficiency by reducing redundant computations and memory accesses. Secondly, in order to capture the scale change of peaches caused by the distance between peaches and the camera in the orchard, and to extract irregular shape features caused by the overlapping of branches and leaves, deformable convolution is introduced to improve the feature extraction capability of the backbone network. Specifically, all convolutions in

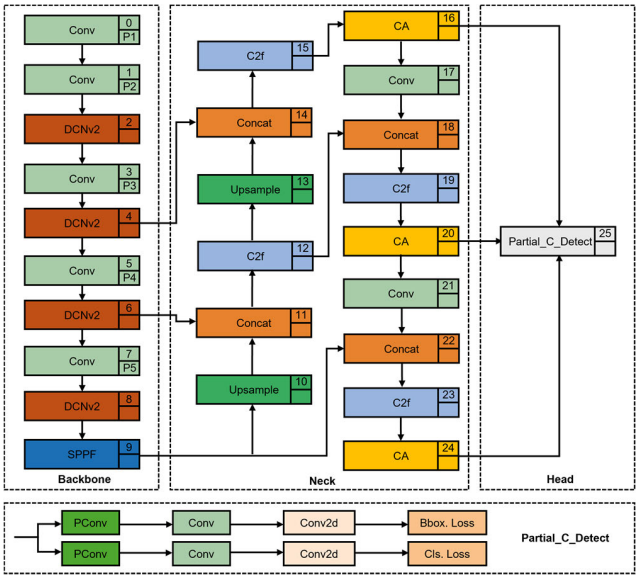

FIGURE 1. PeachYOLO's network structure.

the C2f module of the YOLOv8 backbone network were replaced by DCNv2 [30] to improve the model's performance in recognizing peaches in complex scenes. Finally, in order to locate and identify peaches in dense scenes more accurately, this study added CA [31] in the head network, effectively improving the model detection capability and speed, and reducing missed and wrong detection. These three modules are explained in detail next.

1) PARTIAL_C_DETECT

The basic idea of Partial Convolution (PConv) [29] is to execute a regular convolution operation solely on a portion of the channels in the input feature map for spatial feature extraction while maintaining the information in the other channels unchanged [32]. Figure 2(a)(b) shows the structural diagrams of Partial Convolution and Conventional Convolution, respectively. Compared to traditional convolution operations, PConv can better capture partial features in an image, thus enhancing the model's ability to identify accurately.

Simultaneously, the model's parameters can be significantly reduced, making it more lightweight and suitable for resource-constrained devices. The input and output feature maps possess an equal number of channels, maintaining generality. The number of FLOPs for convolution using a kernel size of $k \times k$ for a tensor with an input shape of $h \times w \times c_p$ can be computed as follows:

$$h \times w \times k^2 \times c_p^2 \tag{1}$$

For a typical value of r = 1/4, the number of FLOPs in PConv is only 1/16 of that in regular Convolution. In addition, PConv incurs lower memory access costs, it is only 1/4 of the Conventional Convolution, therefore:

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p \tag{2}$$

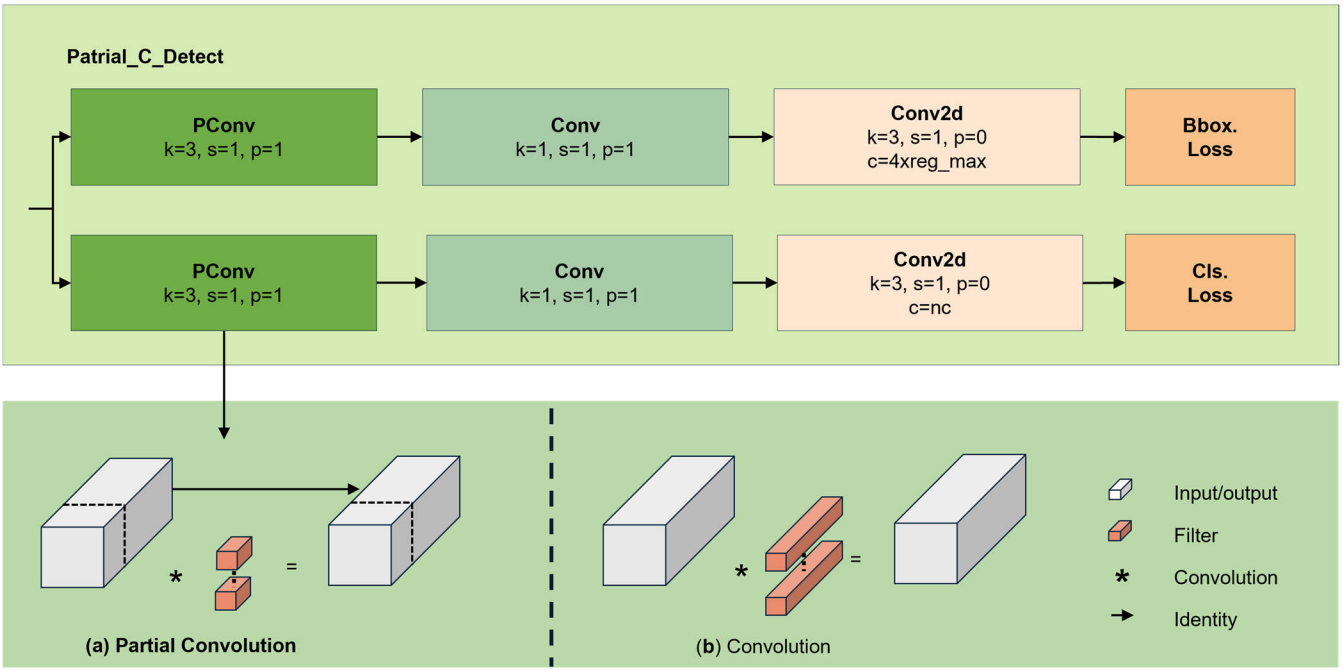

FIGURE 2. Partial_C_Detect structure diagram.

In practical situations pertaining to the detection of peach, it is frequently necessary for models to function on embedded robotic devices that have limited resources. Therefore, the development of lightweight models is of utmost importance in order to attain efficient and real-time detection. In order to maintain performance while decreasing the number of parameters in the network model, this study defines a new detection head, Partial_C_Detect, as depicted in Figure 2. The decoupled head structure of YOLOv8 consists of two parallel branches to extract categorical features and location features, respectively, and then one layer of 1×1 convolution each to accomplish the classification and localization tasks. The first layer of 3×3 convolution kernels is replaced with PConv, and then the first layer of 3×3 convolution kernels is adjusted to 1×1 convolution kernels.

2) DCNV2

In traditional convolution, the kernel scans the input data with a fixed step size and network architecture. Conventional Convolution is suitable for detecting objects with clear boundaries, but is less robust to unknown geometric transformations and has poor generalization ability [33]. In contrast, Deformable Convolution [34] employs a learnable offset (Δp_n) to dynamically adjust the position of the convolution kernel, better matching the local features of the input data, as illustrated in Figure 3. The equation representing the relationship is displayed below. Here, p_0 denotes each position of the input data, p_n denotes a position in the sampling network, and w represents the weight coefficients of the current position, which collectively form the weight matrix of the convolution kernel.

$$y\left(p_0\right)=\sum_{Pn\in R} w\left(p_n\right) x\left(p_0+p_n+\Delta p_n\right) \tag{3}$$

DCNv2 [30] optimizes the learning offset and modulation scalar based on DCN. Additionally, it introduces the offset modulation mechanism Δm_k to optimize the impact of the offset on the model to some extent. For the points falling outside the range, the algorithm gradually guides them back towards the target object. This enhances the convolution kernel's capability to concentrate on discrete feature information and reconstruct the features of targets affected by occlusion and overlap. This process is calculated as follows:

$$y\left(p_0\right)=\sum_{Pn\in R} w\left(p_n\right) x\left(p_0+p_n+\Delta p_n\right) \Delta m_k \tag{4}$$

In the natural orchard scenes, the spatial position, shape, and size of peaches frequently vary, influenced by various factors like lighting conditions, occlusion, and overlapping with both identical and different objects [35]. To tackle these challenges, DCNv2 is used instead of the traditional convolution in the feature output of the neck network to improve the feature extraction capability of the network for non-regular targets. This approach can more effectively address accuracy issues arising from shape variations due to fruit occlusion. This allows the network model to be better suited for detecting peach fruit in complex natural environments, thereby enhancing its effectiveness in real-world picking operations.

3) COORDINATE ATTENTION

In the domain of deep learning, the deployment of attention mechanisms empowers networks to prioritize and allocate

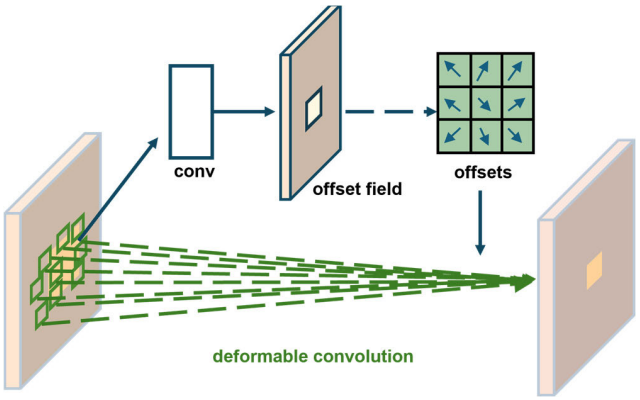

FIGURE 3. DCN structure diagram.

resources to essential regions selectively. This tactic has gained widespread acceptance for its efficacy in architecture. However, the computational overhead of most attention mechanisms is unaffordable due to the limitations of the model size of lightweight networks and the parameters, making the development of attention mechanisms lag in lightweight networks.

Coordinate Attention (CA) [31] is a novel attention mechan- ism designed for lightweight networks proposed by Hou et al., where the model focuses on direction-aware and position-aware information by embedding position information into channel attention. Due to its simplicity, flexibility, and efficiency, it is often inserted into classical lightweight networks to improve the accuracy of the network with little additional computational overhead [36]. Its specific structure is shown in Figure 4.

Departing from the norm where channel attention typically reduces multidimensional inputs to individual channel feature vectors via 2D global pooling, CA adopts a distinctive approach. It fractionates the channel attention mechanism into two 1D feature encoding processes that aggregate features along different directions. This unique design empowers CA to effectively discern long-range correlations along a chosen spatial orientation while concurrently holding onto the precise spatial positioning along the perpendicular dimension.

Subsequently, the feature maps that have been generated are encoded individually in order to create a set of feature maps that are aware of direction and sensitive to position. These feature maps can then be applied in conjunction with the input feature map to improve the representation of the object of interest. Here, z_C is the output of the *c-th* channel, and H and W denote the height and width of the pooling kernel, respectively. The output of the *c-th* channel with height h after horizontal decomposition is as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i) \tag{5}$$

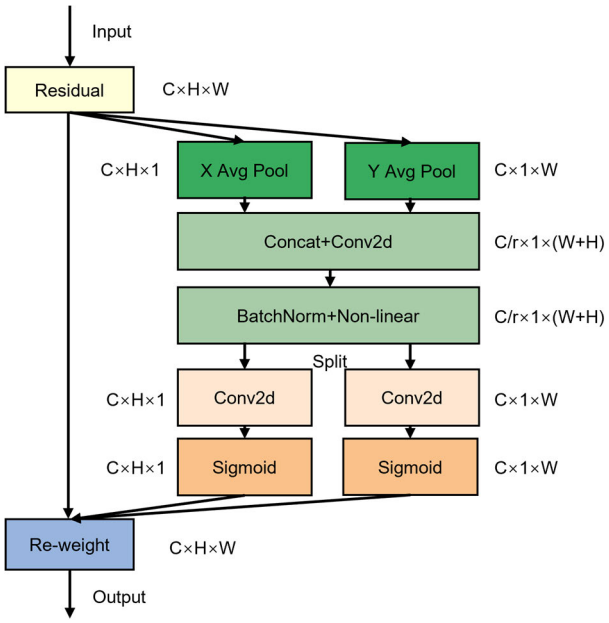

FIGURE 4. CA structure diagram.

The output of the *c-th* channel with width W after vertical decomposition is as follows:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j, w) \tag{6}$$

CA was incorporated into the head of the YOLOv8 network structure to better focus on the important information of peach fruits within complex backgrounds, thereby improving the model's detection accuracy.

IV. EXPERIMENTAL SETUP

A. DATA ACQUISITION

A portion of the peach images utilized in this study were gathered from June to August 2023 in several peach orchards in Longquanyi District, Chengdu City, Sichuan Province, China. The dataset comprises 1,300 images captured under diverse weather conditions, including sunny and cloudy, across three periods: morning, noon, and afternoon. Additionally, images were acquired under various lighting conditions, including downlight, sidelight, and backlight. During the shooting process, simulations of robot operations for picking were conducted. The shooting angle and distance were consistently varied to capture images featuring diverse colors, sizes, lighting conditions, backgrounds, and fruit overlaps and blockages. A set of images of peaches in a typical complex environment is shown in Figure 5. Given that the collected peach images mainly consisted of pippin peach and honey peach, considering the versatility of the vision system, additional images depicting various peach varieties (nectarine, flat peach, Eagle's Beak Honey Peach, etc.) in complex environments of other orchards were collected through the Internet. A total of 2116 peach images were obtained through screening.

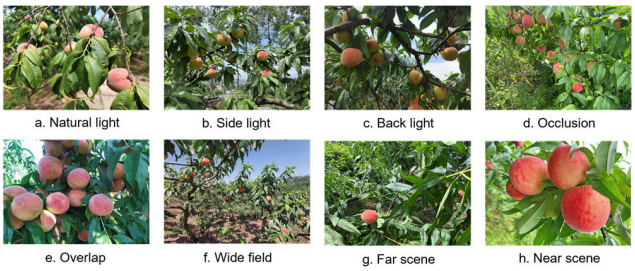

FIGURE 5. Peach images in complex scenes.

TABLE 1. The division of the dataset.

Name	Proportion	Number of Pictures	Number of Labeled Peach
training set	70%	1481	12498
validation set	10%	212	1708
test set	20%	423	3617
total	100%	2116	17823

B. DATASET PRODUCTION

In this study, the LabelImg [37] software was utilized to manually label a rectangular box around the region containing the target fruit in each image, with the label set to 'peach'. The rectangular box was closely adhered to the fruit's contour during labeling. In the case of other peach fruits that overlapped and were obscured by branches and leaves, only the visible part was annotated.

To achieve a balanced dataset distribution, 2116 images of peaches, totaling 17823 fruits, are categorized into four groups: A, B, C, and D. These categories are defined based on the quantity of peaches in the images, as described below:

A: The count of peaches in one image is five or fewer, totaling 989 instances;

B: The count of peaches in one image ranges from greater than five to ten, totaling 561 instances;

C: The count of peaches in one image exceeds ten but is fifteen or fewer, totaling 269 instances;

D: The count of peaches in one image surpasses fifteen, totaling 297 instances.

Subsequently, the peach images were renamed, and the dataset was partitioned according to a ratio of 7:1:2. Ultimately, the training set consisted of 1481 images, the validation set consisted of 212 images, and the test set consisted of 423 images. The division of the datasets is presented in Table 1.

C. DATA ENHANCEMENT

In practical agricultural contexts, the environment associated with automated peach picking displays significant variability and complexity. For our model to simulate and account for this diversity, data augmentation is considered an effective method. Augmented, diverse training instances

FIGURE 6. Illustration of data augmentation. (a)The initial image; (b) Images after data augmentation.

are generated by applying transformative operations to the current dataset without necessitating its expansion. Implementing this approach can significantly mitigate the issue of overfitting in the model, hence enhancing its robustness and ability to generalize [38]. In this experiment, a random data augmentation approach was employed to process 900 training images. This method includes random cropping, flipping, rotating, erasing, adding noise, scaling, and adjusting contrast, aiming to simulate various scenarios that may occur during peach picking. It was ensured that each enhanced image underwent at least one type of processing. Figure 6 shows the effect of data augmentation, where Figure 6(a) shows the initial image, while Figure 6(b) shows the image after data augmentation process.

D. IMPLEMENTATION DETAILS

The research was carried out at the Sichuan Key Laboratory of Agricultural Information Engineering on a Lenovo Thinkstation P920. The specific experimental configuration is shown in Table 2. This ensures efficient and accurate training of the neural network models used in the study. The chosen configuration facilitated the training of the neural network models utilized in the study, resulting in both efficiency and accuracy. It should be noted that the test environment is identical to the training environment to ensure consistency and reproducibility of the experimental results.

In this study's experimental model, the relevant hyperparameters are set as follows: the model receives a uniform input image with a resolution of 640 $\times$ 640 pixels, the initial learning rate is 0.01, the learning rate momentum is 0.937, the optimizer uses Adam, and the weight decay value is 0.0005. The training batch size is set to 16, and the model undergoes 200 rounds of training epochs. To ensure objectivity, this study assesses the performance of the proposed method by conducting a series of experimental trials with the same hyperparameter settings.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EVALUATION INDEX

When assessing the computational complexity and efficiency of peach fruit recognition algorithms, several vital metrics are taken into account, including Frames Per Second (FPS), model size (measured in terms of parameters), and GFLOPs (Giga Floating Point Operations Per Second). FPS quantifies the real-time performance potential

TABLE 2. Experimental configurations were used in this study.

Environment	Version
CPU	lntel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz
GPU	NVIDIA Quadro RTX 5000 16G
RAM	128G
CUDA/Cudnn	V10.2/V8.1.1
Python	V3.8
Pytorch	V1.8.1

of the model by calculating how many frames it can process each second. The number of model parameters is an indicator of the model's compactness, ewer parameters typically imply better resource efficiency. Meanwhile, GFLOPs represent the massive-scale computing capacity, defining the number of billion floating-point operations that a hardware platform can carry out in a single second, thus serving as a standard benchmark for gauging computational strength.

The evaluation metrics used to assess the accuracy of the peach fruit recognition algorithm in this experiment were precision (P), recall (R), the reconciled mean of P and R (F1), average precision (AP), and mean average precision (mAP). The research focuses on a single objective, resulting in equal values for AP and mAP when only one class is considered. The evaluation metrics that follow utilize mAP values. TP (True Positive) is the number of correctly detected peach fruits; TN (True Negative) is the number of correctly detected non-peach objects; FP (False Positive) is the number of non-peach objects detected as peach fruits; FN (False Negative) is the number of missed peach fruits.

B. ABLATION EXPERIMENTS

To assess the effectiveness of the modules introduced in this study, an ablation experiment was conducted using the dataset constructed in this study, with P representing the Partial_C_Detect module, D representing the DCNV2 module, and C representing the CA module. Table 3 displays the results of the ablation experiments, with "√" denoting the utilization of this improved method. Eight experiments were conducted, each with different modules added, and compared with the original YOLOv8n model using F1, mAP, FPS, Parameters, and GFLOPs as metrics.

As depicted in Table 3, incorporating the Partical_C-_Detect module, DCNV2 module, and CA module into YOLOv8 effectively enhances the detection accuracy of the network, resulting in an improvement of 0.4%, 0.4%, and 0.4%, respectively, compared to the YOLOv8n model. In addition, it is observed that the parameters and the amount of computation of YOLOv8 can be significantly reduced by incorporating the Partial_C_Detect module into the network, reaching 2.4M and 5.5 GFLOPs, which are 19.5% and 32.1% lower than the original version. Replacing the original c2f layer with the DCNV2 module in YOLOv8 allows better adaptation to the geometrical transformations of the target, thereby improving accuracy. Furthermore, by combining

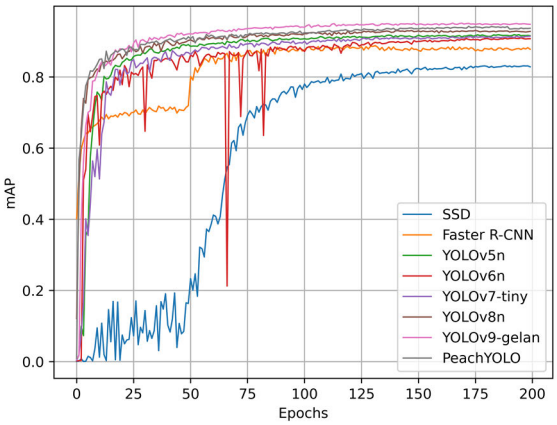

FIGURE 7. Accuracy variation of eight object detectors.

modules in pairs, the combination of Partial_C_Detect + DCNV2 module achieves parameters and GFLOPs metrics of 2.6M and 5.1 GFLOPs, respectively, representing a reduction of 14.0% and 37.0% compared to the original version. The combination of Partial_C_Detect + CA module can significantly improve the inference speed of the model to 532.5 FPS, and the mAP of the model is 93.3%. Although the combination of DCNV2 + CA module increases the number of parameters and GFLOPs, the mAP is as high as 93.6%. Taking into account accuracy, number of parameters, computational resources, and inference time, the combination of Partial_C_Detect, DCNV2, and CA modules (i.e., PeachYOLO) is ultimately selected. PeachYOLO requires only 2.6M parameters, with an inference time of 1.9ms for a single image and 3G fewer GFLOPs than the original YOLOv8, which most hardware devices can accept. This is due to the Partial_C_Detect and DCNV2 modules, which bring lightweight and high-precision performance to the model while adding the CA module, resulting in a qualitative enhancement to its detection capabilities. In summary, while our approach sacrifices some inference speed, it notably enhances accuracy, which constitutes a valuable improvement.

C. COMPARISON EXPERIMENTS

To demonstrate the merits of the proposed algorithm, this experiment compares the PeachYOLO model with SSD, Faster R-CNN, YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8n, and YOLOv9-gelan networks. The improved model is compared with seven advanced object detectors, and the results are illustrated with line plots indexed by mAP. First, the eight curves with different colors in Figure 7 demonstrate the advantage of the YOLO series in peach detection. The Faster R-CNN and SSD models converge more slowly, and the early training of the SSD model exhibits more significant fluctuations.

Table 4 presents comparisons of F1, Recall, Precision, mAP, FPS, parameters, and GFLOPs among different models. Although the accuracy of YOLOv9-gelan is as high as

TABLE 3. Ablation experiment result.

Methods	P	D	C	F1	Recall/%	Precision/%	mAP/%	FPS	parameters	GFLOPs
YOLOv8n				0.88	86.2	89.1	92.8	497.1	3005843	8.1
YOLOv8-P	√			0.88	87.2	87.8	93.2	579.9	2419283	5.5
YOLOv8-D		√		0.88	93.6	91.9	93.2	297.8	3170807	7.7
YOLOv8-C			√	0.88	86.7	88.4	93.2	472.2	3017563	8.1
YOLOv8-P+D	√	√		0.88	85.7	90.1	93.2	322.9	2584247	5.1
YOLOv8-P+C	√		√	0.88	86.0	89.6	93.3	532.5	2431003	5.5
YOLOv8-D+C		√	√	0.88	86.5	89.3	93.6	291.5	3182527	7.8
YOLOv8-P+D+C	√	√	√	0.88	86.1	90.2	93.8	526.3	2595967	5.1

TABLE 4. Comparison test table with other algorithms.

Model	F1	Recall/%	Precision/%	mAP/%	FPS	parameters/10^6	GFLOPs
SSD	0.77	69.0	86.2	81.4	30.2	4.3	9.8
Faster R-CNN	0.71	75.5	66.8	72.9	15.2	-	-
YOLOv5n	0.86	84.6	88.3	91.7	660.6	1.7	4.2
YOLOv6n	0.54	79.5	73.7	90.7	400.3	4.6	11.3
YOLOv7-tiny	0.88	85.9	89.6	91.5	576.1	6.0	13.2
YOLOv8n	0.88	86.2	89.1	92.8	497.1	3.0	8.1
YOLOv9-gelan	0.90	90.9	88.4	95.1	64.3	31.2	116.8
PeachYOLO(ours)	0.88	86.1	90.2	93.8	526.3	2.6	5.1

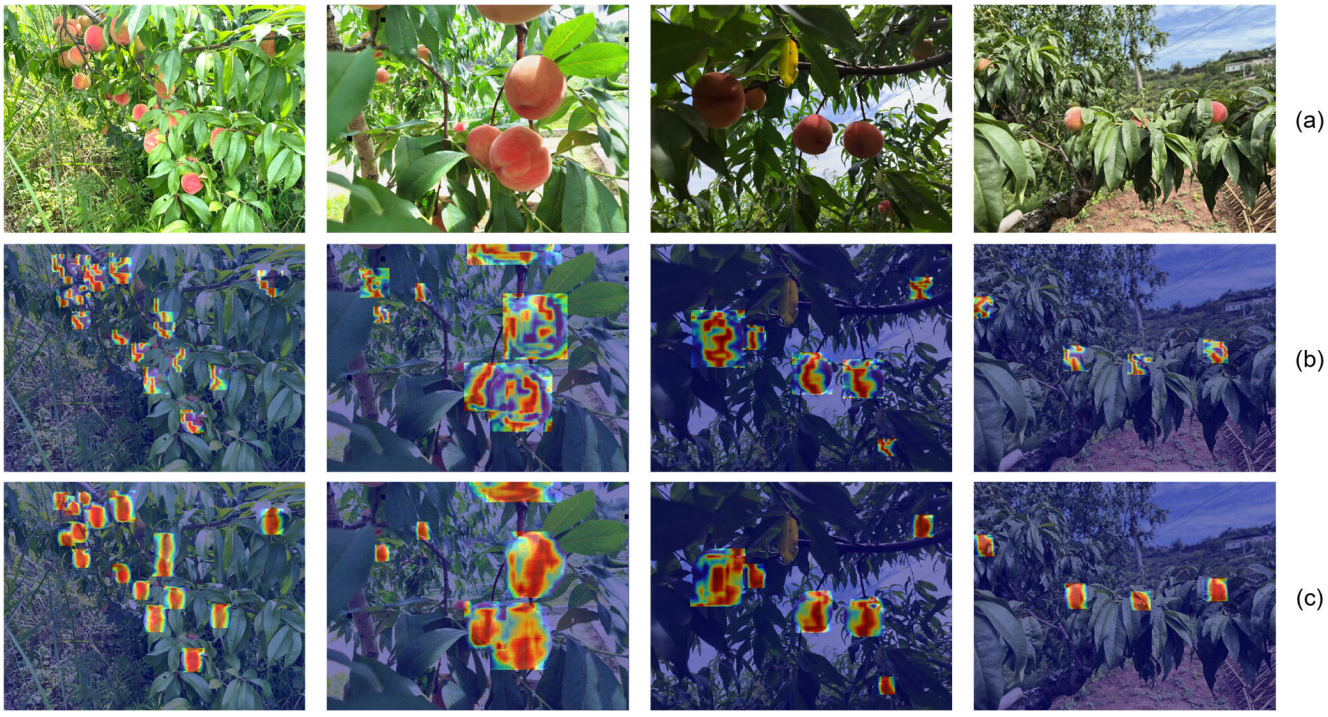

FIGURE 8. The visualization examples generated by GradCAM.(a)Original images;(b)YOLOv8n results;(c)PeachYOLO results.

95.1%, the number of parameters is up to 31.2M, which is not suitable for practical application and will not be discussed in the following. The mAP of the PeachYOLO model is 12.4%, 20.9%, 2.1%, 3.1%, 2.3%, and 1.0% higher than that achieved by the SSD, Faster R-CNN, YOLOv5n, YOLOv6n, YOLOv7-tiny, and YOLOv8n models, respectively. This indicates that the PeachYOLO algorithm achieves high accuracy in recognizing peach fruits. The PeachYOLO

model achieves the highest F1 score of 0.88. Given that Faster R-CNN is a two-stage object detection model with a lengthy average detection time per image, it will not be further discussed. In terms of parameters and GFLOPs, the YOLO family outperforms; however, despite YOLOV5n having only 1.7M parameters, comprehensive consideration must be given to accuracy. Overall, the PeachYOLO model achieves a balance between accuracy and parameters, making it more suitable for practical applications.

D. VISUALIZATION ANALYSIS

1) GRAD-CAM ANALYSIS

To further validate the performance of the improved algorithm for detecting peach fruits in complex backgrounds, the Gradient-weighted Class Activation Mapping(Grad-CAM) [39] heat map visualization module was employed. The red areas are the areas that the network model focuses on, with darker colors indicating greater attention. The comparison of the heat map prior to and following algorithm improvement is depicted in Figure 8. Observations reveal that the improvement focuses more attention on the region where the target is situated, concentrating computational resources in its vicinity. This effectively suppresses non-target regions from consuming computational resources, thereby verifying the effectiveness of the improved model.

2) FRUIT OVERLAP AND BRANCH SHADDING

In natural environments, shading of fruits from each other and shading of fruits from branches, leaves, and trunks occur frequently. The absence of contour information for fruit parts heightens the challenge of fruit detection. In cases of heavy occlusion, where contour information is diminished, detecting fruits becomes challenging. Therefore, this study tested scenarios where the fruits overlapped and examined different degrees of branch shading. The original YOLOv8 network exhibited issues of leakage and misdetection, as evidenced by the undetected fruits marked with blue circles in Figure 9, as well as the fault detected fruits indicated by yellow circles. When branches and leaves heavily occlude the fruits, the prediction box size of the improved PeachYOLO network is closer to the size of the actual contour.

3) FRUITS WITH DIFFERENT DENSITIES

In cases where the fruit size is large, the quantity is low, and the target contour is distinct, training can yield more valid data, resulting in effective detection. Conversely, when the fruit size is small, and the quantity is large, the available valid data decreases, leading to increased difficulty in recognition and a higher likelihood of missed detections. Therefore, this study conducted a comparative experiment on the detection effect of fruits with different degrees of densification. The detection results are shown in Figure 10.

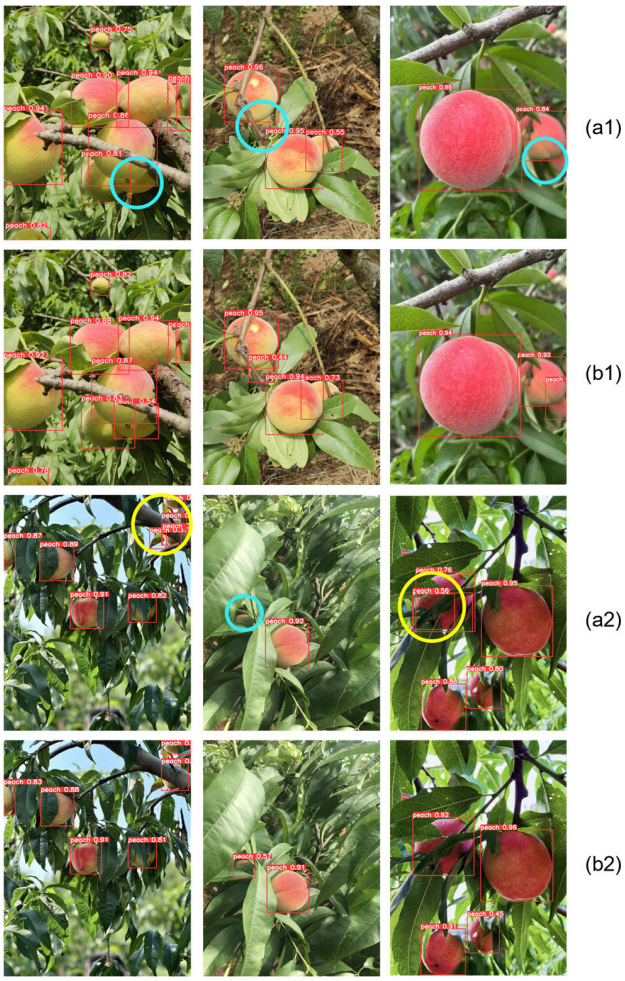

FIGURE 9. Comparative experimental analysis of fruit overlap and branch shading. (a1) YOLOv8 results with overlapping fruit; (a2) PeachYOLO results with overlapping fruit; (b1) YOLOv8 results with branch and leaf shade; (b2) PeachYOLO results with branch and leaf shade.

E. DISCUSSION

Fruit object detection in complex orchard environments is highly susceptible to problems such as fruit occlusion and overlap. Currently, there are relatively few studies on peach detection in complex orchard environments, and there is an imbalance between accuracy and speed. Meanwhile, the lack of publicly available peach detection datasets is not conducive to developing peach detection. Therefore, this study constructed a peach detection dataset in complex environments containing multiple conditions. This study proposed PeachYOLO, a lightweight and efficient object detection model for peaches, which achieved excellent detection capability in complex orchard environments. The following sections will explore the feasibility of our proposed method and discuss its potential drawbacks.

1) Limitations of data collection: Although this study discussed peach data in the complex environment of the orchard, it has some limitations. For instance, this study did not collect data under extreme weather conditions

FIGURE 10. Comparative experiments with fruits of different densities. (a)YOLOv8n results;(b)PeachYOLO results.

like heavy fog and rain. Therefore, there is still room for improvement in this study. To more comprehensively consider the effects of various environmental conditions on the algorithm's performance, future research can actively collect more diverse data, especially under severe weather conditions, to improve the robustness and generalization ability of the model.

2) Practical Challenges of Algorithm Deployment: At present, fruit object detection generally stays in the algorithmic stage, and there is still a big gap between it and its deployment in real scenarios. Although our algorithm performs well theoretically, its application and deployment in natural orchards require further research and exploration. Future efforts should concentrate on converting the algorithm into a practical and deployable system, considering hardware platform constraints, real-time demands, and integration with existing orchard management systems to guarantee the algorithm's effectiveness and reliability in real-world scenarios.
3) Detection accuracy and speed: The mAP of our proposed PeachYOLO reaches 93.8%, with a detection time of only 1.9ms per image, meeting the fundamental requirements of industrial applications. Nevertheless, there is still potential for enhancing the precision of detection.

This study presents a novel solution for peach fruit detection in complex orchard environments. In the future, our goal is to broaden the application of our model to other fruits and further refine the algorithm to cater to practical application scenarios.

VI. CONCLUSION

In response to the challenges of widely varying target scales, irregular fruit shapes, and the imbalance between accuracy and model size in peach fruit detection in complex orchard environments, this study proposes a light-weight peach fruit detection algorithm called PeachYOLO. The effectiveness of these modules was verified through ablation experiments. The experimental results show that the mAP of PeachYOLO reaches 93.8%, which is 1.0% higher than that of the original model; the computation amount is only 5.1GFLOPs. the number of model parameters is 2.6M, and the inference time of a single image is 1.9ms, which is suitable for deployment on Automatic picking robots, and lays the foundation for realizing the unmanned intelligent picking of peaches. The research in this study provides a technical reference for detecting and localizing other fruits and vegetables. Future research will focus on the use of computer vision techniques for crop growth monitoring, crop pest and disease monitoring, and fruit and vegetable yield prediction, thereby advancing the development of modern agricultural automation.

ACKNOWLEDGMENT

The authors would like to thank Yujie Lei and Yan Guan for providing article presentation suggestions.

REFERENCES

[1] C. Guo, X. Wang, Y. Li, X. He, W. Zhang, J. Wang, X. Shi, X. Chen, and Y. Zhang, "Carbon footprint analyses and potential carbon emission reduction in China's major peach orchards," *Sustainability*, vol. 10, no. 8, p. 2908, Aug. 2018.
[2] L. Wang, "Current situation and development suggestions of peach industry in China," *China Fruits*, vol. 10, no. 10, pp. 1–5, 2021.
[3] L. Xu and C. Chen, "Economic situation and development countermeasures of Chinese peach industry," *J. Fruit Trees*, vol. 40, no. 1, pp. 133–143, 2023.
[4] M. Crochon, "Quality of peaches as a function of picking time and consumer's preferences," in *Proc. Int. Conf. Peach Growing*, vol. 173, 1984, pp. 433–440.
[5] J. Chen, H. Liu, Y. Zhang, D. Zhang, H. Ouyang, and X. Chen, "A multiscale lightweight and efficient model based on YOLOv7: Applied to citrus orchard," *Plants*, vol. 11, no. 23, p. 3260, Nov. 2022.
[6] Y. Tang, J. Qiu, Y. Zhang, D. Wu, Y. Cao, K. Zhao, and L. Zhu, "Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review," *Precis. Agricult.*, vol. 24, no. 4, pp. 1183–1219, Aug. 2023.
[7] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved YOLOv5," *Remote Sens.*, vol. 13, no. 9, p. 1619, Apr. 2021.
[8] J. Glenn. (2023). *YOLOv8[EB/OL]*. [Online]. Available: https://github.com/ultralytics/ultralytics
[9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[11] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 379–387.
[12] W. Liu, "SSD: Single shot multibox detector," in *Proc. Comput. Vis.-ECCV 14th Eur. Conf.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.
[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[18] N. Martin. (2020). *YOLOv5[EB/OL]*. [Online]. Available: https://github.com/ultralytics/yolov5
[19] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[20] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.
[22] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.
[23] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. W. Hasan, "Real-time detection of ripe oil palm fresh fruit bunch based on YOLOv4," *IEEE Access*, vol. 10, pp. 95763–95770, 2022.
[24] Q. An, K. Wang, Z. Li, C. Song, X. Tang, and J. Song, "Real-time monitoring method of strawberry fruit growth state based on YOLO improved model," *IEEE Access*, vol. 10, pp. 124363–124372, 2022.
[25] R. Sapkota, D. Ahmed, M. Churuvija, and M. Karkee, "Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques," *IEEE Access*, vol. 12, pp. 43436–43452, 2024.
[26] X. Du, H. Cheng, Z. Ma, W. Lu, M. Wang, Z. Meng, C. Jiang, and F. Hong, "DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels," *Comput. Electron. Agricult.*, vol. 214, Nov. 2023, Art. no. 108304.
[27] R. Tang, Y. Lei, B. Luo, J. Zhang, and J. Mu, "YOLOv7-plum: Advancing plum fruit detection in natural environments with deep learning," *Plants*, vol. 12, no. 15, p. 2883, Aug. 2023.
[28] B. Zhang, R. Wang, H. Zhang, C. Yin, Y. Xia, M. Fu, and W. Fu, "Dragon fruit detection in natural orchard environment by integrating lightweight network and attention mechanism," *Frontiers Plant Sci.*, vol. 13, Oct. 2022, Art. no. 1040923.
[29] J. Chen, "Run, don't walk: Chasing higher FLOPS for faster neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 12021–12031.
[30] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9300–9308.
[31] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13713–13722.
[32] H. Xu, L. Wang, and F. Chen, "Advancements in electric vehicle PCB inspection: Application of multi-scale CBAM, partial convolution, and NWD loss in YOLOv5," *World Electric Vehicle J.*, vol. 15, no. 1, p. 15, Jan. 2024.
[33] H. Wei, E. Xu, J. Zhang, Y. Meng, J. Wei, Z. Dong, and Z. Li, "BushNet: Effective semantic segmentation of bush in large-scale point clouds," *Comput. Electron. Agricult.*, vol. 193, Feb. 2022, Art. no. 106653.
[34] J. Dai, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
[35] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Comput. Electron. Agricult.*, vol. 116, pp. 8–19, Aug. 2015.
[36] X. Chen, Y. Lu, B. Cao, D. Lin, and I. Ahmad, "Lightweight head pose estimation without keypoints based on multi-scale lightweight neural network," *Vis. Comput.*, vol. 39, no. 6, pp. 2455–2469, Jun. 2023.
[37] D. Tzutalin. (2015). *LabelImg[EB/OL]*. [Online]. Available: https://github.com/HumanSignal/labelImg
[38] H. S. Ullah and A. Bais, "Evaluation of model generalization for growing plants using conditional learning," *Artif. Intell. Agricult.*, vol. 6, pp. 189–198, 2022.
[39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

TINGXUAN LI is currently pursuing the B.S. degree in computer science and technology with Sichuan Agricultural University, China. Her current research interests include the development and optimization of target detection algorithms, with a specific interest in applying computer vision techniques in the agricultural Internet of Things.

QIYANG CHEN is currently pursuing the bachelor's degree in computer science and technology with Sichuan Agricultural University, China. He is dedicated to enhancing the security and privacy protection of AI models. His research interests include AI security assessment, with a focus on security assessment and vulnerability mining for AI models.

XINYI ZHANG is currently pursuing the Bachelor of Engineering degree in data science and big data technology with Sichuan Agricultural University, China. Her primary research interest includes the application of deep learning techniques for target detection in orchards. With a focus on orchard productivity enhancement, her research goal is to develop efficient and accurate fruit detection systems using deep learning methodologies.

SHAOYUN DING is currently pursuing the degree in computer science and technology with Sichuan Agricultural University, China, with a specific focus on artificial intelligence. He is dedicated to leveraging computer vision techniques to address challenges in agricultural production, aiming to deliver practical applications, and innovative solutions to the agricultural sector.

XIANYAO WANG is currently pursuing the bachelor's degree in Internet of Things Engineering with Sichuan Agricultural University, China. His current research interests include the Internet of Things, edge computing, and distributed earning.

JIONG MU received the B.S. degree from Sichuan Normal University, China, in 1993, and the M.S. degree from Sichuan University, China, in 2008. Currently, she is a Professor with the College of Information Engineering, Sichuan Agricultural University. Her research interests include computer vision, the agricultural Internet of Things, and agricultural engineering.