ReB-DINO: A Lightweight Pedestrian Detection Model with Structural Re-Parameterization in Apple Orchard

Ruiyang Li[1], Ge Song[1], Shansong Wang[1], Qingtian Zeng[1], Guiyuan Yuan[1], Weijian Ni[1], nengfu xie[2], and Fengjin Xiao[3]

[1]Shandong University of Science and Technology
[2]Agricultural Information Institute of CAAS
[3]Beijing Climate Center

July 13, 2024

Abstract

As agricultural machinery evolves towards intelligence and automation, obstacle detection in agricultural environments becomes crucial for safe operations of intelligent agricultural machinery. Pedestrians, as one of the most common obstacles in orchards, usually exhibit autonomy and behavioral unpredictability. Therefore, the development of intelligent agriculture requires reliable pedestrian detection technology. To address this, we propose ReB-DINO, a robust and accurate orchard pedestrian object detection model based on an improved DINO. Initially, we improve the feature extraction module of DINO using structural re-parameterization, enhancing accuracy and speed of the model through training and decoupling inference. In addition, a progressive feature fusion module is employed to fuse the extracted features and improve model accuracy. Finally, the network incorporates a convolutional block attention mechanism and an improved loss function to improve pedestrian detection rates. The experimental results demonstrate a 1.6% improvement in Recall on the NREC dataset compared to the baseline. Moreover, the results show a 4.2% improvement in mAP and the number of parameters decreases by 40.2% compared to the original DINO, enhancing accuracy and real-time object detection in apple orchards while maintaining lightweight attributes, surpassing mainstream object detection models.

ReB-DINO: A Lightweight Pedestrian Detection Model with Structural Re-Parameterization in Apple Orchard

Ruiyang Li[1] | **Ge Song[2]** | **Shansong Wang[1]** | **Qingtian Zeng[1]** | **Guiyuan Yuan[1]** | **Weijian Ni[1]** | **Nengfu Xie[3]** | **Fengjin Xiao[4]**

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

[2]College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China

[3]Agricultural Information Institute of CAAS, Beijing 100081, China

[4]National Climate Center, Beijing 100081, China

Correspondence

Ge Song, College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China.
Email: songge@sdust.edu.cn

Abstract

As agricultural machinery evolves towards intelligence and automation, obstacle detection in agricultural environments becomes crucial for safe operations of intelligent agricultural machinery. Pedestrians, as one of the most common obstacles in orchards, usually exhibit autonomy and behavioral unpredictability. Therefore, the development of intelligent agriculture requires reliable pedestrian detection technology. To address this, we propose ReB-DINO, a robust and accurate orchard pedestrian object detection model based on an improved DINO. Initially, we improve the feature extraction module of DINO using structural re-parameterization, enhancing accuracy and speed of the model through training and decoupling inference. In addition, a progressive feature fusion module is employed to fuse the extracted features and improve model accuracy. Finally, the network incorporates a convolutional block attention mechanism and an improved loss function to improve pedestrian detection rates. The experimental results demonstrate a 1.6% improvement in Recall on the NREC dataset compared to the baseline. Moreover, the results show a 4.2% improvement in mAP and the number of parameters decreases by 40.2% compared to the original DINO, enhancing accuracy and real-time object detection in apple orchards while maintaining lightweight attributes, surpassing mainstream object detection models.

KEYWORDS

agriculture, object detection, pedestrian detection, lightweight, re-parameterization

1 | INTRODUCTION

With the ongoing trend towards intelligence and automation, smart agriculture has emerged as the future trajectory of agricultural advancement[1]. Currently, smart agriculture effectively harnesses diverse advanced technologies to foster sustainable agricultural production[2] and offer solutions for intelligent, automated, and unmanned agricultural practices[3]. Serving as a pivotal component of smart agriculture, intelligent agricultural machinery is capable of performing tasks such as crop harvesting and yield monitoring[4].

Even in the absence of direct human interaction, advanced agricultural machinery poses significant safety risks, especially in the presence of individuals on-site [5]. Therefore, reliable individual detection is crucial for fully automated systems to reduce accidents. Currently, numerous object detection algorithms find extensive application in mainstream obstacle detection domains[6]. Qiu et al.[7] creatively combined the improved YOLOv3 algorithm with the DeepSORT method to effectively detect and track farmers and water buffaloes within paddy fields. However, in orchard environments, Li et al.[8] devised a lightweight network and incorporated a Gaussian model to enhance the detection capabilities regarding common obstacles found in orchards, thereby establishing a basis for obstacle avoidance in intelligent orchard robots. Similarly, Su et al.[9] employed the K-means clustering algorithm and SE attention mechanism to obstacle detection within a semi-structured apple orchards. This was coupled with pruning procedures to guarantee model detection speed, followed by utilizing identified tree trunks for route computation. Additionally, in unstructured orchards, Huang et al.[10] accomplished unified obstacle avoidance and path planning by integrating

depth information with the Dynamic Window Approach (DWA). Despite initial challenges, Despite initial deviations, the robot successfully navigated through real orchards. However, the local convolutional operations of Convolutional Neural Networks (CNNs) result in a limited capacity to perceive long-range pixel relationships, thereby diminishing their efficacy in addressing global issues. Consequently, the introduction of DETR[11] and ViT[12] has garnered attention to Transformer-based object detection approaches. In the same year, Zhu et al.[13] proposed Deformable DETR, aiming to mitigate lengthy training periods and suboptimal detection performance concerning small objects in DETR. And other researchers also made improvements and optimizations to DETR, proposing models such as Conditional DETR[14], DN-DETR[15], DAB-DETR[16]. Additionally, Zhang et al.[17] proposed DETR with improved denoising anchor boxes (DINO), introducing a contrastive denoising training approach and a mixed query selection method based on DETR.

The application of object detection algorithms for pedestrian detection within agricultural environments demands both high accuracy and real-time performance. However, models often struggle with deep network architectures and complex parameter designs, leading to substantial computational resource usage, thereby posing challenges in practical inference processes[18]. This underscores the necessity for additional optimization of current networks while simultaneously enhancing detection accuracy. Consequently, structural re-parameterization methods have been introduced and extensively implemented in models. ACNet[19] is widely regarded as one of the pioneering models to employ structural re-parameterization method, enhancing model performance by employing asymmetric convolution for network fusion without increasing additional computational costs. Ding et al.[20] introduced RepVGG specifically designed for inference hardware chips. It adopts a VGG-like single-branch structure comprising solely 3×3 convolutions and activation functions during the inference process, showcasing a favorable speed-accuracy trade-off. Diverse Branch Block (DBB)[21] adopted the multi-branch architecture from Inception, enabling the model to substitute any $K \times K$ convolution during the training phase, consequently acquiring comprehensive image feature information. Additionally, Ding et al.[22] introduced RepLKNet, which employs large kernel convolutions to effectively expand the receptive field while simultaneously reducing model depth, leading to a notable enhancement in model performance. To mitigate the challenge of heightened computational complexity associated with large kernel convolutions, UniRepLKNet[23] combines large kernel convolutions with dilated convolutions and proposes four architectural guidelines, showcasing superior performance across diverse domains.

Achieving a balance between accuracy and computational efficiency remains a challenging task for conventional orchard obstacle detection approaches in computer vision. Therefore, we propose an object detector based on an improved DINO algorithm, which improves detection accuracy while ensuring model's lightweightness. The main contributions of this study are as follows:

- We propose the ReB-DINO, a lightweight method for pedestrian detection in apple orchards, incorporating a structural re-parameterization method to significantly decrease the number of parameters during inference time.
- We integrate a progressive Bidirectional Feature Pyramid Network (BiFPN) to enhance the multi-scale feature fusion and representation capabilities of the model. The Convolutional Block Attention Module (CBAM) is inserted between the backbone network and feature fusion layers to optimize the extracted feature maps.
- Additionally, we use the Minimum Point Distance Intersection over Union (MPDIoU) loss function instead of the Generalized Intersection over Union (GIoU) loss function. These adjustment ensure that the final predictions of the model closer to the ground-truth, thereby effectively enhancing prediction accuracy.
- Moreover, by conducting training and testing on the NREC agricultural pedestrian detection dataset[5], we attain robust detection performance and establish a benchmark on the dataset.

2 | MATERIALS AND METHODS

2.1 | Overview Framework

The architecture of the ReB-DINO model proposed for pedestrian detection in orchard environments is shown in Figure 1, comprising three main components: a backbone feature extraction module, a neck feature fusion module, and an object detection head with transformer blocks. The backbone feature extraction module extracts feature information from the input image, while the neck module employs lateral connection blocks for multi-scale feature fusion from top to bottom, thereby acquiring high-level semantic information of varying scales. The object detection head is trained using a mixed query selection strategy and
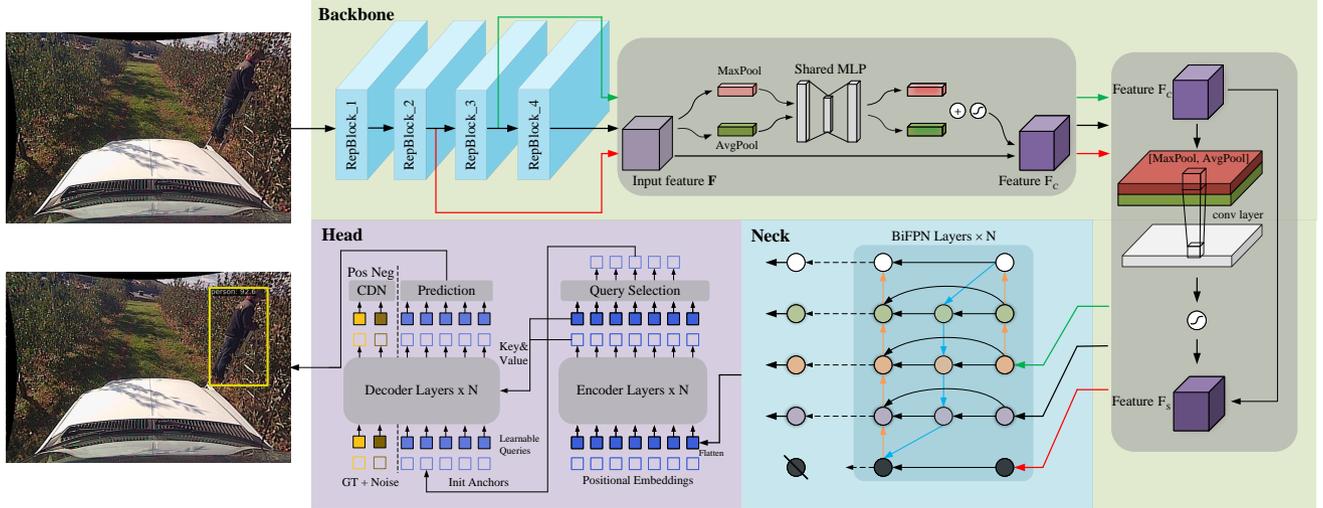
FIGURE 1 The ReB-DINO network architecture for pedestrians detection in orchard.

contrastive denoising transformer blocks to detect objects. The proposed model is optimized for lightweight feature extraction, multi-scale feature fusion and loss function metrics, facilitating rapid and accurate pedestrian detection in orchards.

2.2 | Improvement of backbone network

In the training phase, branch networks are employed to ensure the model achieves high accuracy. We stack multiple network blocks to construct the training-time structure, inspired by ResNet. In the inference phase, the multi-branch structure blocks are equivalently re-parameterized into single-branch structures through two methods: linearization and module sqeezing, thereby ensuring efficient prediction. These network blocks will contain the following transformations:

1. A Conv-BN layer is fused into Conv layer: W and W^{merged} denote the weights of the i-th convolutional kernel in the convolutional layers before and after fusion, respectively. b^{merged} represents the i-th bias term in the fused convolutional layer. The weights and biases of this layer can be expressed as follows:

$$W_i^{merged} = \frac{\gamma_i}{\sqrt{\sigma_i^2 + \epsilon}} W_i \qquad (1)$$

$$b_i^{merged} = -\frac{\mu_i \gamma_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i \qquad (2)$$

where ϵ is set to 1×10^{-4} to maintain stability, and $\mu, \sigma, \gamma, \beta$ denote the accumulated mean, standard deviation, learned scaling factor, and bias of the BN layer following the 3×3 Conv, respectively.

2. 1×1 Conv and residual branches are fused into 3×3 Conv: the residual branch can be regarded as a 1×1 convolution with the identity matrix as the kernel, followed by zero-padding to achieve a 3×3 Conv. Following this, the 1×1 Conv and residual branches can be seen as a Conv-BN layer and then perform transformation 1.

3. Multiple parallel Convs are fused into one single Conv: $W^{inference}$ and $b^{inference}$ denote the weight and bias of the fused convolutional layers during the inference-time, respectively, while W^{merged} and b^{merged} denote the weight and bias of each branch after padding on the parallel branches. Therefore, the formulas for computing the weights and biases of the fused convolutional layer are as follows:

$$W^{inference} = W_1^{merged} + W_2^{merged} + \cdots + W_N^{merged} \qquad (3)$$
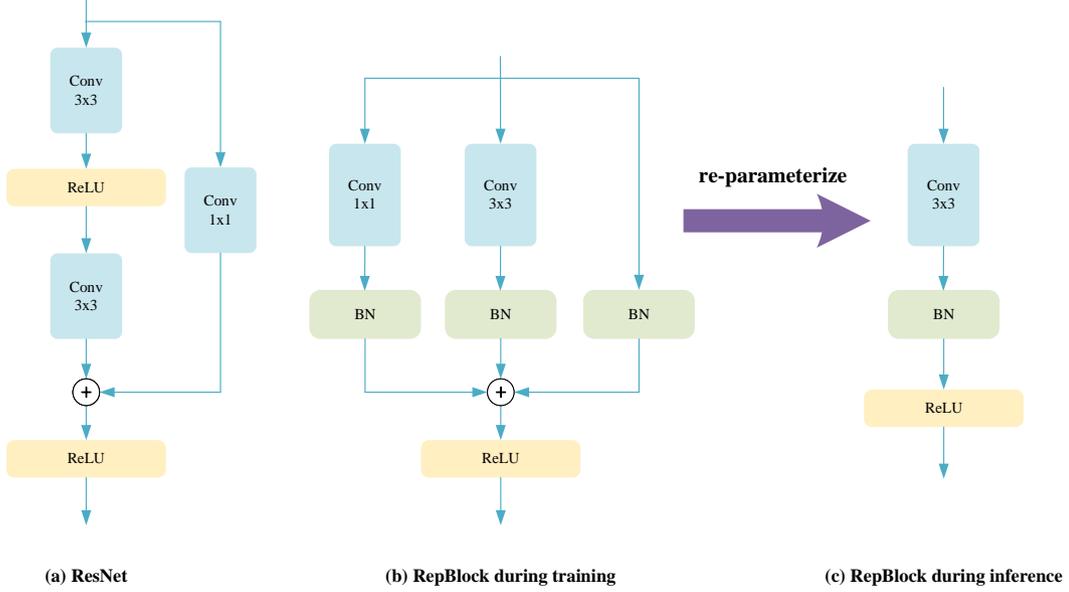
FIGURE 2 (a) Structure of ResNet. (b), (c) Structure of RepBlock in training-time and inference-time.

$$b^{inference} = b_1^{merged} + b_2^{merged} + \cdots + b_N^{merged} \qquad (4)$$

where N denotes the number of parallel branches. In this paper, we set N to 3, indicating the presence of three branches.

Thus, the number of layers in each stage of the backbone network is determined by the following principles: **1.** Given that the input image to the backbone network has the highest resolution and high computational complexity, only one RepBlock is utilized in the first stage to reduce computational losses. **2.** Since the final stage typically requires more channels to maintain rich feature representations, we employ only one RepBlock to preserve the parameters. **3.** The majority of RepBlock layers are allocated to the second last stage, aligning with ResNet and its variants. This enables the model to process feature maps at lower resolutions more effectively and achieve enhanced representation capabilities.

Three branches were employed to construct the structural re-parameterization module. In Figure 2 (b), a RepBlock layer comprises a 3×3 Conv branch, a 1×1 Conv branch, and an identity branch during training-time, with each branch containing a BN layer. Moreover, the multi-branch layer can be fused using the above structural fused methods 1, 2, and 3. Subsequently, it can be transformed into a single branch structure for inference, as shown in Figure 2 (c).

2.3 | Convolutional block attention module

The feature map for orchard pedestrian detection often contains irrelevant information like trees, weeds and ground, thus CBAM[24] was incorporated at the end of the backbone. CBAM consists of channel attention mechanism (CAM) and spatial attention mechanism (SAM) in principle, which can make the model pay more attention to the pedestrian features in the image.

The channel attention mechanism enhances the feature expression of each channel by employing global max pooling and global average pooling to obtain feature vectors for individual channels. Subsequently, attention weights are obtained through fully connected layers to weigh the channels. This process accentuates channels relevant to pedestrian detection in the orchard while suppressing irrelevant ones. Conversely, the spatial attention mechanism emphasizes the positional information of features by highlighting the significance of different positions. It generates spatial attention feature maps by employing average pooling and max pooling along the channel dimension and concatenating them.

Consequently, the integration of CBAM enables the selection of key features relevant to the current task and enhances the representation capacity of CNNs. The comprehensive computational formula for CBAM is as follows:

$$F_C = Att_C(F) \otimes F \tag{5}$$

$$F_S = Att_S(F_C) \otimes F_C \tag{6}$$

where $F \in \mathbb{R}^{C \times H \times W}$ denotes the feature map from the input module, $\otimes$ signifies element-by-element multiplication, and $F_C \in \mathbb{R}^{C \times H \times W}$ and $F_S \in \mathbb{R}^{C \times H \times W}$ denote the feature maps following channel and spatial attention, respectively. $Att_C(F) \in \mathbb{R}^{C \times 1 \times 1}$ denotes the operation of global average pooling and global maximum pooling on the input feature map F, $Att_S(F_C) \in \mathbb{R}^{H \times W}$ denotes the distinct maximum pooling and average pooling operations conducted on the feature map F_C along the channel dimension. The equations for channel attention and spatial attention are given below:

$$
\begin{aligned}
Att_C(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\
&= \sigma(W_1(W_0(F_{avg}^C)) + W_1(W_0(F_{max}^C)))
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
Att_S(F_C) &= \sigma(f^{7 \times 7}([AvgPool(F_C); MaxPool(F_C)])) \\
&= \sigma(f^{7 \times 7}(F_{avg}^S; F_{max}^S))
\end{aligned}
\tag{8}
$$

where σ denotes the Sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$, $W_1 \in \mathbb{R}^{C \times C/r}$, and $f^{7 \times 7}$ corresponds to the convolutional operations employing a 7×7 kernel. F_{avg}^C and F_{max}^C denote global average pooling and maximum pooling in the channel dimension, respectively. Similarly, $F_{avg}^S \in \mathbb{R}^{1 \times H \times W}$ and $F_{max}^S \in \mathbb{R}^{1 \times H \times W}$ indicate the average pooling and maximum pooling operations in the spatial dimension.

2.4 | Improvement of neck network

In the neck section, a progressive bi-directional feature fusion network (BiFPN) is employed for pedestrian detection in orchards. This structure represents an improvement on the feature pyramid (FPN)[25] and path aggregation network (PAN)[26]. FPN represents a traditional top-down feature fusion approach that fuses deep semantic information with shallow texture information. However it struggles to convey the location information of the target. Conversely, PAN supplements FPN with an additional bottom-up fusion path. Despite enhancing the feature characterization, its computational overhead and simplistic structure pose challenges for detecting pedestrians in orchards amid complex environments and low resolutions.

We first use a top-down approach is employed to fuse multi-scale features, while bottom-up paths are added for progressive underlying feature fusion. Given the minimal contribution of a relay node at the edge of modules to the overall network, BiFPN improves the above two feature fusion networks by removing intermediate nodes at the top and bottom of the network structure and eliminating redundant connections between them to reduce the parameters. Additionally, we also establish residual connections between input and output nodes at the same level. This strategy aims to fuse original obstacle features more effectively without escalating computational costs. As shown in Figure 3, each progressive bi-directional path is regarded as a module, with these modules interconnected iteratively three times to achieve a higher level of feature fusion.

Furthermore, the prior feature fusion approach treats feature maps of varying importance equally, which is extremely unreasonable. The contribution to feature fusion should vary based on the distinct poses and movements of target pedestrians in feature maps of different resolutions. Consequently, BiFPN introduces additional weights to each input during the feature fusion process, enhancing the ability of the network to discern the significance of various input features.

$$O = \sum_i \frac{\omega_i}{\varepsilon + \sum_j \omega_i} I_i \tag{9}$$

where O denotes the output features, I_i represents the input features, ω_i and ω_j denote the parameters obtained through network learning, and ϵ is set to 1×10^{-4} to maintain stability in the values.

In Figure 3, the P_6 level feature map serves as an example. The input and output formulas are given by:

$$P_6^{mid} = DSConv\left(\frac{\omega_1 \cdot P_6^{in} + \omega_2 \cdot Resize(P_7^{in})}{\omega_1 + \omega_2 + \varepsilon}\right) \tag{10}$$

$$P_6^{out} = DSConv\left(\frac{\omega_1' \cdot P_6^{in} + \omega_2' \cdot P_6^{mid} + \omega_3' \cdot Resize(P_5^{out})}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \tag{11}$$
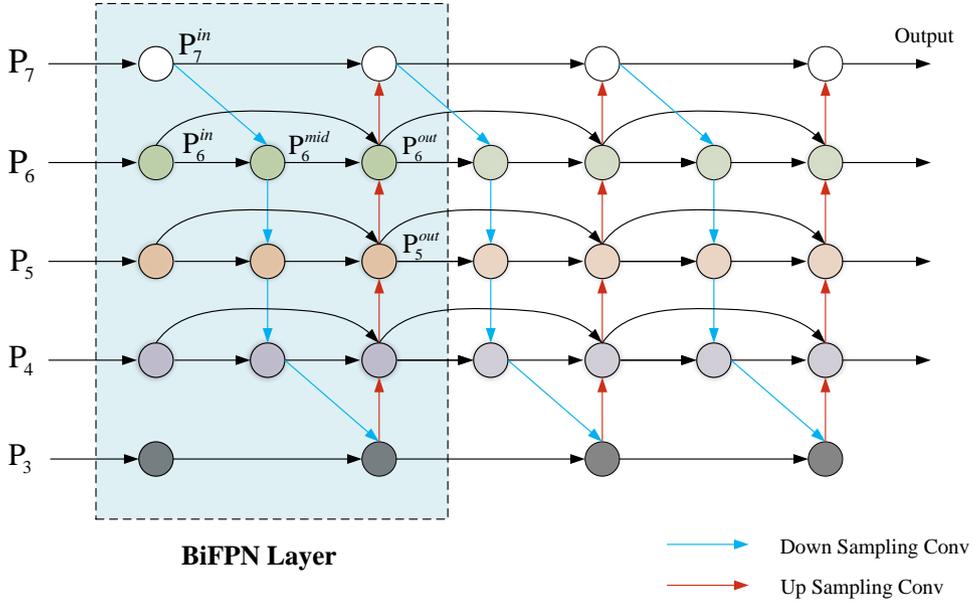
FIGURE 3   The overall structure of BiFPN with the basic layer components in the dashed box.

where P_6^{mid} denotes the relay feature node in the top-down path at the P_6 level, P_6^{out} represents the output feature node in the bottom-up path at the P_6 level, and the *Resize*() function denotes to resize feature maps with different resolutions to the same resolution sized feature maps, and *DSConv* denotes depth separable convolution.

2.5 | Loss function

The generalized intersection over union (GIoU)[27] was employed as the bounding box loss function in the original DINO. While GIoU considers the overlapping area and incorporates a penalty term, it struggles to differentiate between the two when the predicted box is within the ground truth box. To tackle this limitation, we introduce the Minimum Point Distance Intersection over Union (MPDIoU) as the localization loss function for bounding box regression. This loss function aims to minimize the distance between top-left and bottom-right corners to compute the loss. The formula for calculating the squared Euclidean distance between the corner points of the predicted and ground truth boxes is as follows:

$$d_1^2 = (x_1^{gt} - x_1^{prd})^2 + (y_1^{gt} - y_1^{prd})^2 \tag{12}$$

$$d_2^2 = (x_2^{gt} - x_2^{prd})^2 + (y_2^{gt} - y_2^{prd})^2 \tag{13}$$

where (x_1, y_1), (x_2, y_2) denote the coordinates of the upper left and lower right corners, respectively, and d_1, d_2 denote the Euclidean distances between these points. The formula for *MPDIoU* regression loss function is given by:

$$MPDIoU = \frac{\mathcal{B}_{gt} \cap \mathcal{B}_{prd}}{\mathcal{B}_{gt} \cup \mathcal{B}_{prd}} - \frac{d_1^2}{h^2 + w^2} - \frac{d_2^2}{h^2 + w^2} \tag{14}$$

where h and w denote the height and width of the input image, $\mathcal{B}_{gt}$ denotes the ground truth bounding box, while $\mathcal{B}_{prd}$ denotes the predicted bounding box. Specific parameters for the loss function are detailed in Figure 4.

The ratio of the intersection and union of $\mathcal{B}_{gt}$ and $\mathcal{B}_{prd}$ is the formula for the intersection over union. Ordinary *IoU* can only compute the union area of the two bounding boxes, and cannot differentiate between cases where the two boxes do not overlap. When $|\mathcal{B}_{gt} \cap \mathcal{B}_{prd}| = 0$, $IoU(\mathcal{B}_{gt}, \mathcal{B}_{prd}) = 0$, and in this case *IoU* cannot reflect the positional relationship between the two bounding boxes. Additionally, all factors in the existing loss function of the bounding box regression can be determined by the
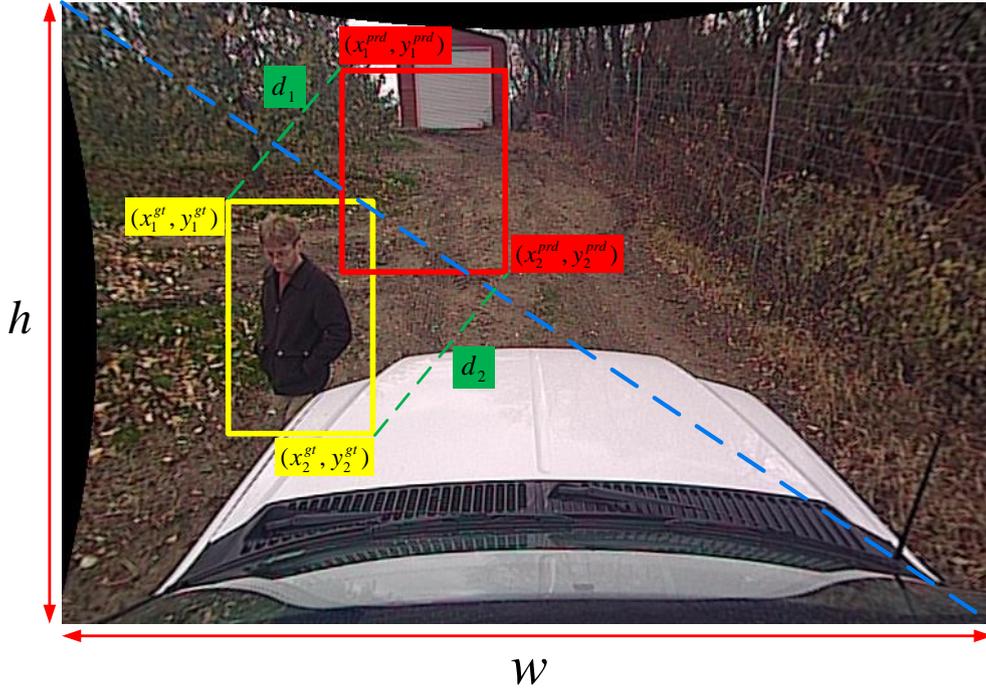
FIGURE 4 The diagram depicts the significance of each parameter in IoU.

four point coordinates, which are calculated as follows:

$$|C| = (max(x_2^{gt}, x_2^{prd}) - min(x_1^{gt}, x_1^{prd})) * (max(y_2^{gt}, y_2^{prd}) - min(y_1^{gt}, y_1^{prd})), \qquad (15)$$

$|C|$ denotes the area of the smallest outer rectangle that covers $\mathcal{B}_{gt}$ and $\mathcal{B}_{prd}$ of the area of the minimum outer rectangle. The formula for the center points of the bounding box is as follows:

$$x_c^{gt} = \frac{x_1^{gt} + x_2^{gt}}{2}, y_c^{gt} = \frac{y_1^{gt} + y_2^{gt}}{2}, x_c^{prd} = \frac{x_1^{prd} + x_2^{prd}}{2}, y_c^{prd} = \frac{y_1^{prd} + y_2^{prd}}{2}, \qquad (16)$$

(x_c^{gt}, y_c^{gt}) and (x_c^{prd}, y_c^{prd}) denote the center coordinates of the ground truth and predicted boxes, respectively.

$$w_{gt} = x_2^{gt} - x_1^{gt}, h_{gt} = y_2^{gt} - y_1^{gt}, w_{prd} = x_2^{prd} - x_1^{prd}, h_{prd} = y_2^{prd} - y_1^{prd}. \qquad (17)$$

where w_{gt} and h_{gt} denote the width and height of the ground truth bounding box, and w_{prd} and h_{prd} denote the width and height of the predicted bounding box.

In Eqs. (15)-(17), all the factors considered by the bounding box loss function can be determined by the coordinates of the two points in the top left and bottom right corners. This approach not only simplifies the calculation process, but also takes into account the existing loss functions $\mathcal{L}_{GIoU}$[27], $\mathcal{L}_{EIoU}$[28], $\mathcal{L}_{CIoU}$[29], $\mathcal{L}_{DIoU}$[29] advantages such as non-overlapping areas, center point distances, and deviation in aspect ratios.

3 | RESULTS

3.1 | Datasets

The study uses the NREC Agricultural Pedestrian Detection Dataset[5], curated by Carnegie Mellon University, for model training, testing, and evaluation. The NREC dataset contains 95,924 images from apple orchards and orange groves, making it a widely

utilized resource for pedestrian detection in agricultural settings. Sorely images from apple orchards were used as the basis for model training. For object detection purposes, the NREC dataset includes two categories, "person" and "person-part". The proportion of positive and negative samples in the dataset is shown in Table 1.

TABLE 1   Training set, validation set, test set proportions and positive and negative sample proportions in the NREC dataset.

NREC Dataset	Training	Validation	Test	Total
positive	15,535	8200	7691	31,426
negative	4570	1981	1949	8500
total	20,105	9781	9640	39,926

The second and third rows of the table indicate the positive and negative samples in the NREC dataset, respectively, while the columns display the counts of positive samples, negative samples, and total images in each set. The training, validation, and test sets consist of 20,105, 9781, and 9640 images, respectively. Additionally, the NREC dataset is standardized to a fixed size of 720×480, which provides a benchmark for the model performance.

3.2 | Evaluation Indicators

The accuracy of bounding box predictions is assessed using the average precision (AP) series of evaluation metrics, a common measure that evaluates the percentage of correct predictions. AP_s, AP_m, and AP_l denote the AP prediction results for objects with bounding boxes areas smaller than 32^2, between 32^2 and 96^2, and larger than 96^2, respectively. For ease of computing AP_{50} and AP_{75}, this study uses IoU as the threshold between the predicted and ground truth bounding box, where AP_{50} and AP_{50} indicate the AP values when IoU exceeds 0.5 and 0.75, respectively. Moreover, model prediction results are evaluated using mAP, which is the average of AP values across all object categories, to evaluate model accuracy. The calculation formulas are as follows:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{18}$$

$$AP = \int_0^1 P(R)\, dR \tag{19}$$

$$P = \frac{TP}{TP + FP} \tag{20}$$

$$R = \frac{TP}{TP + TN} \tag{21}$$

Where N denotes the number of categories, TP and FP denote true positive predictions and false positive predictions, FN denotes true negative predictions. If $TP = FP = 0$, P and R are both 0, indicating that no pedestrians are detected in the image. And a higher AP indicates superior detector performance. Furthermore, we use the number of parameters (Param) and Giga floating-point operations (GFLOPs) to evaluate the lightweight attribute of the model, where smaller GFLOPs means less computational complexity and better inference performance.

3.3 | Experimental Implementation details

In this section, Table 2 show some of the language environments and the software and hardware settings in the experimental process.

To ensure the comparability and fairness, we set the identical hyper-parameters for the same type of experiments. The Adaptive Moment Estimation with Decoupled Weight Decay (AdamW) optimizer was employed, setting the initial learning rate and weight decay set to 0.0001 and 0.001, respectively. Considering hardware limitations, the batch size was set to 2. The Faster R-CNN with ResNet-50 was configured for 100 epochs, Yolox for 200 epochs, while DINO and ReB-DINO followed a 2x

T A B L E 2   Experimental conditions.

Experimental Environment	Details
Programming Language	Python 3.8.10
Operating System	Ubuntu 20.04.3 LTS
Deep Learning Framework	Pytorch 1.10 + CUDA 11.3
CPU	Intel Xeon(R) Platinum 8255C@2.5GHz
GPU	NVIDIA GeForce RTX 2080 Ti

schedule equivalent to 24 epochs; other models were programmed for 50 epochs. All experiments were conducted with three different random seeds, averaging the outcomes to populate the data tables.

3.4 | Comparison with different object detection models

ReB-DINO incorporates novel feature extraction and fusion modules, alongside a new loss function to improve model generalization. We conduct comparative experiments with other models, with results displayed in Table 3.

T A B L E 3   Comparison of different models of orchard pedestrian detection.

Models	Epochs	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l	Recall	Param	Flops
Faster R-CNN [30]	100	31.2	51.6	34.1	21.2	36.6	70.3	35.5	41.353M	197G
Yolox [31]	200	28.3	51.2	27.9	23.3	28.9	50.1	38.2	n/a	n/a
Deformable DETR [13]	50	35.5	57	40.3	28.5	38.3	67.6	52.9	40.099M	184G
Conditional_DETR [14]	50	38.5	57.1	44.3	29.1	43.1	77.2	46.1	43.449M	**95.703G**
DAB_DETR [16]	50	30.2	46.7	35.2	21.2	33.2	66.7	40.7	n/a	n/a
DCN [32]	50	36.4	52.6	45.4	27.2	40.8	74.5	42.6	41.934M	170G
DCNv2 [33]	50	36.9	54	46.8	26.6	42.4	72.7	41.1	149M	224G
DINO [17]	24	39.7	56	**48.4**	27.9	46.2	**79.6**	64.9	47.542M	261G
ours	24	**43.9**	**63.2**	47.2	**36.4**	**47.7**	77.2	**67.4**	**28.392M**	<u>101G</u>

Our model achieves mAP, mAP_{50}, mAP_s, mAP_m, and Recall values of 43.9%, 63.2%, 36.4%, 47.7%, and 67.4%, respectively. Remarkably, our Recall value surpasses the baseline (65.8%) reported in the NREC dataset [5]. We observe a 31.3% reduction in parameters, with a corresponding 12.7% increase in mAP compared to the traditional object detector Faster R-CNN [30]. In comparison to the original DINO model [17], our model achieves a 40.2% reduction in parameters and a 4.2% enhancement in mAP. The advancement is attributed to improvements in certain modules, detailed in the ablation experiments in Section 3.5. Therefore, our model offers high detection accuracy, lightweight design, and robustness for detecting pedestrians in apple orchards.

3.5 | Ablation studies

In this section, we conducted ablation experiments to evaluate the impact of each improved component on the model performance. The results are shown in Table 4, demonstrating the effectiveness of the improvements.

The original DINO model achieves a mAP of 39.7% as shown in Table 4. After improvements to the loss function and feature fusion module, model experienced mAP increases of 0.2% and 1.1%, respectively, which can be attributed to the utilization of the minimum point distance loss function and the unique progressive bi-directional feature fusion of the BiFPN. Furthermore, Models with improvements by incorporating structural re-parameterization and the CBAM attention mechanism in the backbone network, resulting in recognition mAP of 40.9% and 41.7%, respectively. Additionally, the backbone improvement through structural re-parameterization resulted in a 39.4% decrease in the parameter count compared to DINO.

Based on re-parameterization, we improve the model performance by improving the feature fusion network and loss function. Model in last second row achieves detection mAP of 43.7%, however our model achieves a 0.2% increase in mAP from model that without improvement of loss function with a final accuracy of 43.9%. Overall, ReB-DINO, with 40.2% fewer parameters

T A B L E 4 Results of ablation experiments.

RepBlock	CBAM	BiFPN	MPDIoU	mAP	mAP$_{50}$	mAP$_{75}$	mAP$_s$	mAP$_m$	mAP$_l$	Recall	Param
				39.7	56	48.4	27.9	46.2	79.6	64.9	47.542M
			✓	39.9	57.1	**48.5**	31.1	46.5	**79.7**	65.1	47.542M
		✓	✓	41.8	61.2	48.1	33.8	46.7	79.5	65.9	45.783M
✓				40.9	58.7	46.2	33	47	68.7	66.3	28.777M
✓	✓			41.7	61.2	46.4	34.5	47.2	70.9	66.1	28.988M
✓	✓	✓		43.7	63.1	47	35.9	47.6	75.8	67	28.392M
✓	✓	✓	✓	**43.9**	**63.2**	47.2	**36.4**	**47.7**	77.2	**67.4**	**28.392M**

than DINO, improves the model's mAP by 4.2%, ensuring lightweight design, and the final improved model comprises 28.392 M parameters. These comparisons demonstrate the efficacy of various improvements in bolstering detection model performance.

3.6 | Comparison of different attention mechanisms

This section evaluates the impact of various attention mechanisms on model performance. The ReB-DINO model without attention mechanisms serves as a benchmark, while four different types of attention mechanisms are added to the end of the feature extraction layer for experimentation. The experiments primarily compare the impact of different attention mechanisms on the model's performance. Additionally, we assess the performance of the model without attention mechanism. The experimental results are shown in Table 5.

**T A B L E 5** Results of different attention mechanisms experiments.

	mAP	mAP$_{50}$	mAP$_{75}$	mAP$_s$	mAP$_m$	mAP$_l$	Recall	Param
W/o Attention	41.6	59.3	46.5	33.5	46.9	75.3	65.3	28.182M
+SEAM[34]	41.9	61	46.1	32.8	45.4	75.7	64.9	30.1M
+SE[35]	42.7	62.5	**47.8**	34	46.2	76	66.3	28.392M
+CA[36]	42.6	63.1	47.1	33.6	47.3	76.9	63.1	28.498M
ours	**43.9**	**63.2**	47.2	**36.4**	**47.7**	**77.2**	**67.4**	**28.392M**

Table 5 demonstrates that SEAM has a limited impact on model accuracy, resulting in a parameter increase of 1.918M and a marginal 0.3% increase in mAP. The SE and CA modules exhibit comparable effects on the model, with 1.1% and 1.0% increases on mAP and 0.21M and 0.316M increases on parameters, respectively. However, CBAM significantly enhances model accuracy compared to other attention mechanisms, demonstrating a 2.3% improvement in model accuracy with only a 0.74% increase in parameters. Considering the results for Models C and D from Table 4, CBAM effectively integrates channel and spatial information, rendering it more suitable for pedestrian detection in orchards.

3.7 | Comparison of different conditions

In order to further validate the effectiveness of the model under different conditions, this study experiments with pedestrian motion states, poses, and occlusion levels of occlusion as variables for testing. The image examples for each condition in the training, validation, and test sets containing positive samples are shown in Figure 5. The results of the model performance test under each type of condition are shown in Table 6, Table 7 and Table 8.

Figure 5 illustrates the categorization: "Static" and "Moving" represent stationary and continuously moving pedestrians in consecutive frames; "Typical" and "Abnormal" denote the classification of different poses and the specific explanation is given in Section 3.7.2; likewise, "Clear", "Partial" and "Heavy" classify various occlusion levels, which are explained in Section 3.7.3.
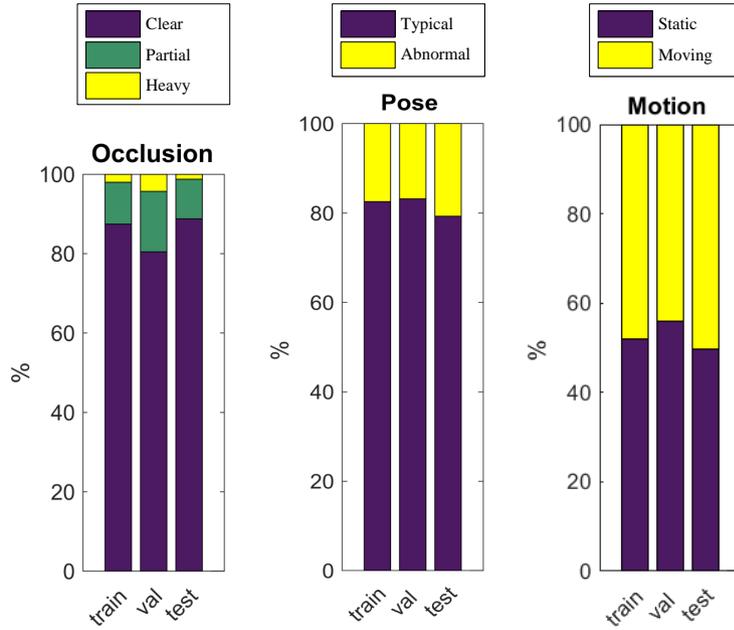
FIGURE 5 Based on different occlusion levels, pedestrian motion states, and various human poses, the data from each domain were divided into subsets.

TABLE 6 Results for people in different motions.

| Motions	mAP	mAP$_{50}$	mAP$_{75}$	mAP$_s$	mAP$_m$	mAP$_l$	Recall
moving	39.9	61.2	48.5	25.5	48.5	79	69.1
static	27.5	45.9	30.5	20	31.6	75.3	61.7

3.7.1 | Experiments in different motions

We segment the test set into different subsets according to two categories: moving and static people. Subsequently, the trained ReB-DINO model was evaluated on these subsets, with experimental results documented in Table 6.

The results show that the model demonstrates excellent performance in detecting pedestrians with coherent motion, achieving a mAP of 39.9% and a Recall of 69.1%, surpassing the baseline by 10.1%. For static individuals, the mAP reaches 27.5%, with a recall exceeding the baseline by 18.1%. Therefore, differences in behavioral poses and other aspects among moving individuals enable the model to learn generalized feature states, contributing to the varied detection performance. Conversely, static individuals exhibit minimal motion variation, leading to less distinct target-to-background boundaries and reduced model generalization for static state detection. Figure 6 and Figure 7 illustrate the detection performance of the model for pedestrians with consistent motion and those in static states, respectively.

3.7.2 | Experiments in different poses

In addition, we discover that different pedestrian poses have certain influence on the detection effect of the model. Pedestrian poses are classified into "abnormal poses" and typical poses, in which "abnormal poses" comprise individuals falling, lying under or between rows of fruit trees, and people sitting on the ground in the orchard. The experimental results shown in Table 7, demonstrating the accuracy of ReB-DINO in detecting pedestrians of different poses.

The experimental results reveal that the model achieves a mAP of 37.2% and 25.8% for typical and "abnormal poses", respectively, and Recall values of 68.6% and 45.6%, which are better than baseline. Detection results of the model are shown in Figure 8.
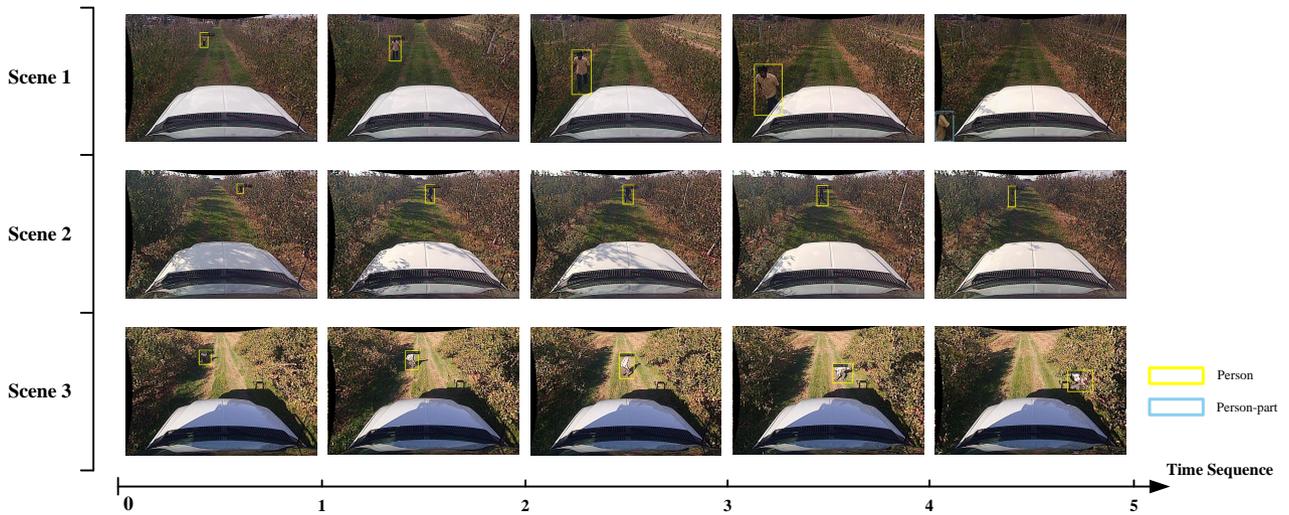
FIGURE 6 Example of detection results of ReB-DINO for pedestrians with coherent motion, where the time sequence is indicated from left to right. "**Scene 1**" denotes pedestrians passing longitudinally through agricultural machine, "**Scene 2**" shows pedestrians crossing transversely through the orchard, and "**Scene 3**" features pedestrians bending over while traversing transversely the orchard.
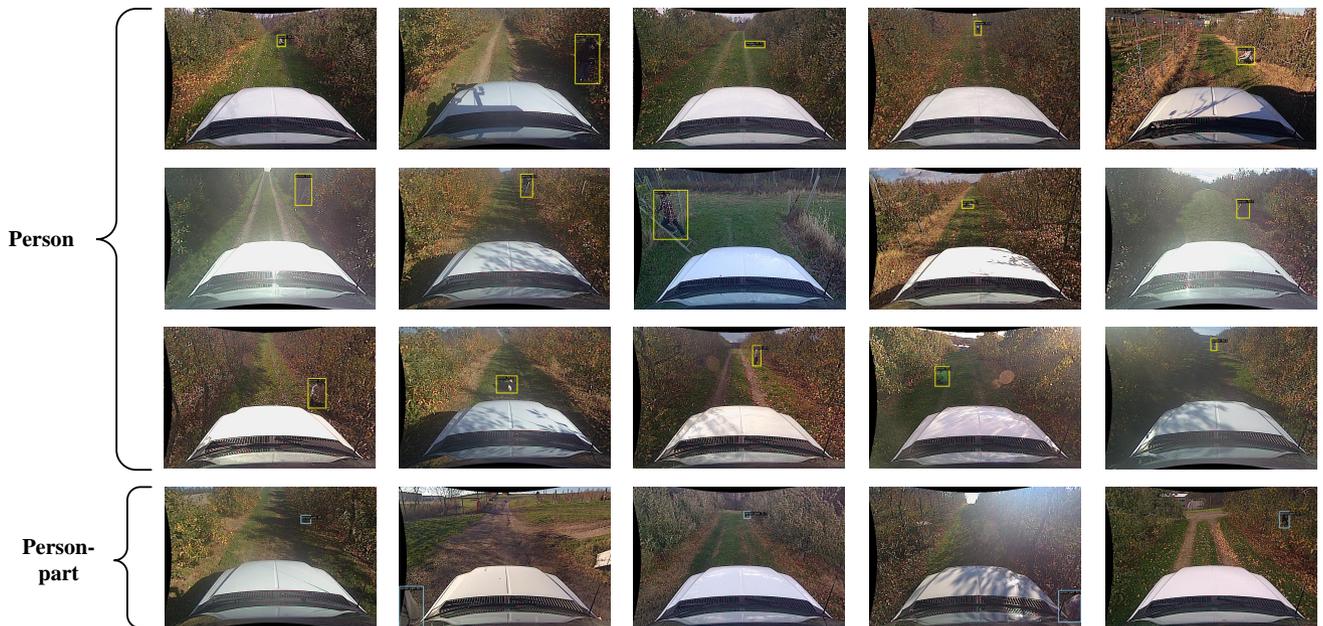

FIGURE 7 The qualitative experimental results for static individuals, where the first three rows indicate the model's predictions labeled "person" in yellow bounding boxes. The last row illustrates detection of heavily occluded individuals, labeled as "person-part" in blue bounding boxes.

3.7.3 | Experiments in different occlusion levels

The experiments on different motions and poses reveal that stationary individuals are often distributed near trees, heavily occluded by leaves, grass, and trunks. Those in continuous motion also susceptible to occlusion by vegetation. Pedestrians in "abnormal poses" blend with the background and suffer significant occlusion influences. Therefore, in order to further evaluate

T A B L E 7 Results for people in different poses.

Poses	mAP	mAP$_{50}$	mAP$_{75}$	mAP$_s$	mAP$_m$	mAP$_l$	Recall
typical	37.2	57.7	44.9	21.9	48.2	76.9	68.6
abnormal	25.8	46.6	24.5	22	29	66.3	45.6
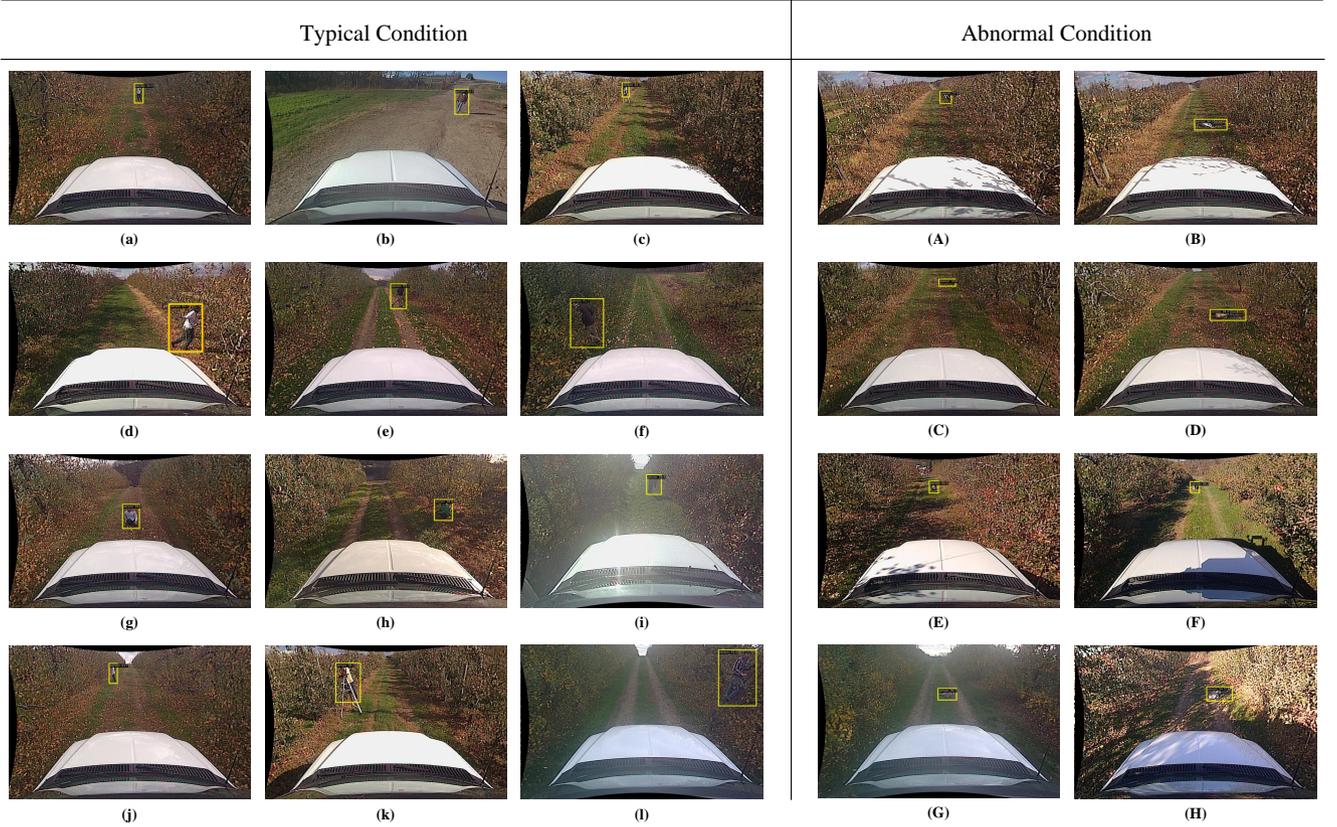
F I G U R E 8 The test results from ReB-DINO illustrate pedestrians in various poses. "Abnormal poses" are categorized into four groups, (**A**), (**B**) for falling people, (**C**), (**D**) for people lying between rows of trees, (**E**), (**F**) for people sitting on the ground, and (**G**), (**H**) for people lying on pathways. Similarly, for typical poses, arranged from top to bottom: (**a**)-(**c**) denote pedestrians facing vehicles, (**d**)-(**f**) show people crossing in an orchard horizontally, (**g**)-(**i**) denote people squatting on the ground or under fruit trees, and (**j**)-(**l**) portray individuals working on ladders.

the model's effectiveness in detecting pedestrians under different occlusion states, inspired by Pezzementi et al.[5], we classify occlusion states into three levels:

1. Clear: Person is more than 70% visible;
2. Partial Occlusion: Person is between 30% to 70% visible;
3. Heavy Occlusion: Person is less than 30% visible. Usually only one body part is visible: an arm, leg, body, or head.

The schematic illustrations of different occlusion scenarios classified in the dataset are depicted in Figure 9.

In Figure 9, "Clear" denotes instances with occlusion less than 30%, as shown in the right image of the left categorization. "Partial occlusion" refers to those with occlusion between 30% and 70%, depicted in the left and center images of the left categorization. "Heavy Occlusion" represents instances with occlusion level exceeding 70%, shown on the right side.
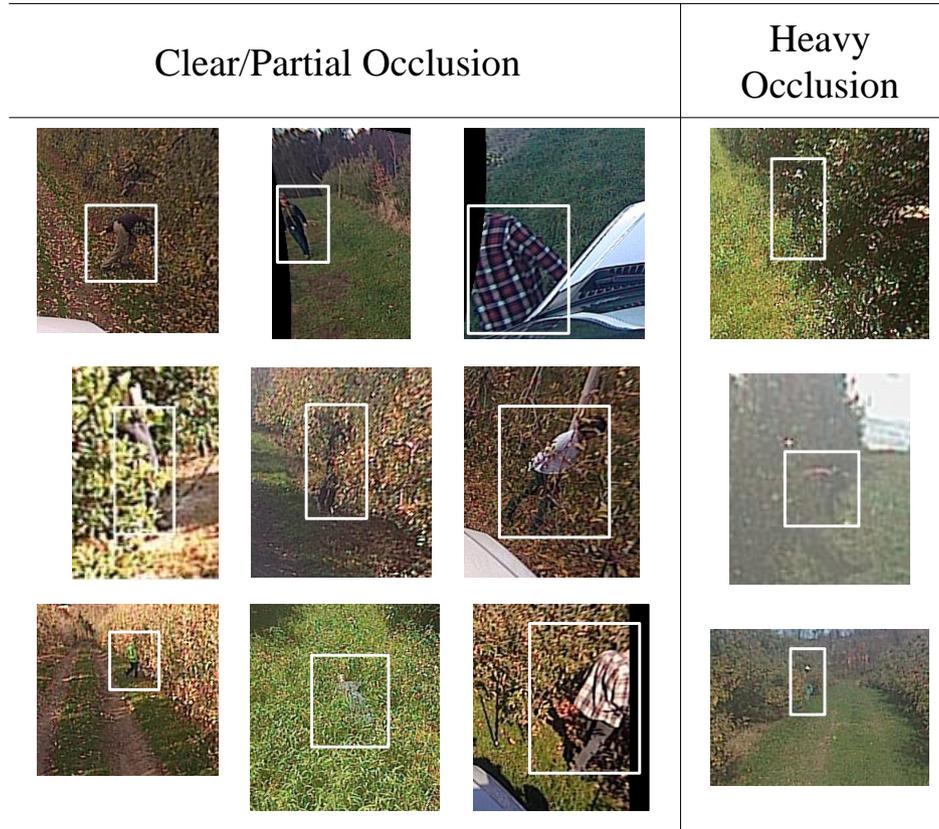
	Clear/Partial Occlusion		Heavy Occlusion

FIGURE 9 The categorization of occlusion levels.

TABLE 8 Results for people in different occlusion levels.

Occlusion levels	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l	Recall
heavy	15	28.8	17	7	28.3	-	54.6
partial	21.9	41.7	20.9	7.4	22.4	41.2	51.3
clear	60.4	93.9	70.6	41.3	67.4	76.7	77.8

- denotes the heavy subset does not contain data of large type.

We find that occlusion usually occurs within a few frames of a person entering or exits the view in the NREC dataset. Figure 9 shows examples of occlusion levels in the dataset across various scenarios. Subsequently, Table 8 shows the detection results of the model under different occlusion levels.

Table 8 illustrates the detection accuracy of the model across different occlusion levels, in which the model performs best in detecting clear individuals, achieving a mAP_{50} of 93.9%, and an overall mAP of 60.4%. Notably, the Recall values for heavy and partial occlusion conditions surpass baseline values at 45.6% and 18.9%, respectively, which indicates that the model's robustness across diverse occlusion conditions.

The experimental results demonstrate that the model proposed in this paper outperforms the baseline under different conditions. Compared to other object detection models, our model is better suited for pedestrian detection in real apple orchards, achieving accurate and rapid detection with strong generalization and robustness.

4 | DISCUSSION

Although the model outperforms similar object detection models in detecting pedestrians under various orchard conditions with improved accuracy and maintained lightweight, the study still identifies two main limitations. Firstly, the detection accuracy for pedestrians with "abnormal pose" and "heavy occlusion" is unsatisfactory, which is attributed to these pedestrians typically blending into the environment with unclear class contour differences in the dataset. However, our model still achieves a Recall of 54.6% for "heavy occlusion" and 45.6% for "abnormal pose" despite these challenges, respectively. The study notes that "heavy occlusion" pedestrians have only 30% visibility, indicating that occluded object detection remains a significant challenge in pedestrian detection. In future work, we will focus on addressing the bounding box regression problem of occluded objects by introducing repulsive loss. In addition, we will also use human body part segmentation method to detect occluded pedestrians.

Another limitation is that this study only focuses on obstacle detection in apple orchard scenes. In the future, we plan to expand our research to detect multiple types of obstacles in different agricultural environments, including cornfields and wheat fields, and apply findings in practical scenarios. In addition, enhancing the image resolution in the dataset will also improve the detection accuracy of the model. In the NREC dataset, the resolution of the image is 720×480, for pedestrians under the condition of "heavy occlusion", the visible pixel size is only a few dozen pixel blocks, which is extremely demanding on the model performance. Therefore, in our future work, we will focus on constructing a multi-obstacle dataset in agriculture to establish a benchmark for agricultural obstacle detection.

5 | CONCLUSIONS

This research aims to develop a lightweight obstacle detection model for real apple orchards. In order to achieve this, we propose ReB-DINO, an improved CNN-Transformer hybrid deep learning model. Our approach utilizes an improved DINO network as the object detection model. By using RepBlock with structural re-parameterization significantly reduce the model's parameters without compromising accuracy. And the model characterization is enhanced through BiFPN fusion of pedestrian multi-scale features. Additionally, we adopt the MPDIoU loss function to enhance model robustness by replacing the GIoU loss function in the original DINO model. The results of this study demonstrate that our model's ability to accurately detect people in various occlusion conditions with a 4.2% mAP improvement and a 61.3% reduction in parameters. Furthermore, the model performs better than the baseline across different environments, motion states, and human poses. Therefore, our model proposed in this work surpasses other similar deep learning models in terms of lightweight attributes, recognition accuracy, and generalization across scenarios, promising advancements in agricultural mechanization and unmanned applications. In the future, we will continue to improve pedestrian detection technology in agriculture environments and update the detection method to develop intelligent agricultural machine.

AUTHOR CONTRIBUTIONS
The authors contributed equally to this work.

CONFLICT OF INTEREST
The authors declare no potential conflict of interests.

REFERENCES
1. Su Y, Wang X. Innovation of agricultural economic management in the process of constructing smart agriculture by big data. *Sustainable Computing: Informatics and Systems.* 2021;31:100579.
2. Maddikunta PKR, Hakak S, Alazab M, Bhattacharya S, Gadekallu TR, Khan WZ, Pham Q-V. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal.* 2021;21(16):17608-17619.
3. Yang X, Shu L, Chen J, Ferrag MA, Wu J, Nurellari E, Huang K. A survey on smart agriculture: Development modes, technologies, and security and privacy challenges. *IEEE/CAA Journal of Automatica Sinica.* 2021;8(2):273-302.
4. Idoje G, Dagiuklas T, Iqbal M. Survey for smart farming technologies: Challenges and issues. *Computers & Electrical Engineering.* 2021;92:107104.

5. Pezzementi Z, Tabor T, Hu P, Chang JK, Ramanan D, Wellington C, Babu BPW, Herman H. Comparing apples and oranges: Off-road pedestrian detection on the nrec agricultural person-detection dataset. arXiv preprint arXiv:1707.07169. 2017.

6. Wang T, Chen B, Zhang Z, Li H, Zhang M. Applications of machine vision in agricultural robot navigation: A review. *Computers and Electronics in Agriculture.* 2022;198:107085.

7. Qiu Z, Zhao N, Zhou L, Wang M, Yang L, Fang H, He Y, Liu Y. Vision-based moving obstacle detection and tracking in paddy field using improved yolov3 and deep SORT. *Sensors.* 2020;20(15):4082.

8. Li Y, Li M, Qi J, Zhou D, Zou Z, Liu K. Detection of typical obstacles in orchards based on deep convolutional neural network. *Computers and Electronics in Agriculture.* 2021;181:105932.

9. Su F, Zhao Y, Shi Y, Zhao D, Wang G, Yan Y, Zu L, Chang S. Tree trunk and obstacle detection in apple orchard based on improved YOLOv5s model. *Agronomy.* 2022;12(10):2427.

10. Huang P, Huang P, Wang Z, Wu X, Liu J, Zhu L. Deep-Learning-Based Trunk Perception with Depth Estimation and DWA for Robust Navigation of Robotics in Orchards. *Agronomy.* 2023;13(4):1084.

11. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. Paper presented at: European conference on computer vision, 2020.

12. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. 2020.

13. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159. 2020.

14. Meng D, Chen X, Fan Z, Zeng G, Li H, Yuan Y, Sun L, Wang J. Conditional detr for fast training convergence. Paper presented at: Proceedings of the IEEE/CVF international conference on computer vision, 2021.

15. Li F, Zhang H, Liu S, Guo J, Ni LM, Zhang L. Dn-detr: Accelerate detr training by introducing query denoising. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

16. Liu S, Li F, Zhang H, Yang X, Qi X, Su H, Zhu J, Zhang L. Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329. 2022.

17. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum H-Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605. 2022.

18. Xie T, Yin M, Zhu X, Sun J, Meng C, Bei S. A Fast and Robust Lane Detection via Online Re-Parameterization and Hybrid Attention. *Sensors.* 2023;23(19):8285.

19. Ding X, Guo Y, Ding G, Han J. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. Paper presented at: Proceedings of the IEEE/CVF international conference on computer vision, 2019.

20. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J. Repvgg: Making vgg-style convnets great again. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.

21. Ding X, Zhang X, Han J, Ding G. Diverse branch block: Building a convolution as an inception-like unit. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.

22. Ding X, Zhang X, Han J, Ding G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

23. Ding X, Zhang Y, Ge Y, Zhao S, Song L, Yue X, Shan Y. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. arXiv preprint arXiv:2311.15599. 2023.

24. Woo S, Park J, Lee J-Y, Kweon IS. Cbam: Convolutional block attention module. Paper presented at: Proceedings of the European conference on computer vision (ECCV), 2018.

25. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

26. Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

27. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S. Generalized intersection over union: A metric and a loss for bounding box regression. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

28. Zhang Y-F, Ren W, Zhang Z, Jia Z, Wang L, Tan T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing.* 2022;506:146-157.

29. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU loss: Faster and better learning for bounding box regression. Paper presented at: Proceedings of the AAAI conference on artificial intelligence, 2020.

30. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems.* 2015;28.

31. Ge Z, Liu S, Wang F, Li Z, Sun J. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430. 2021.

32. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y. Deformable convolutional networks. Paper presented at: Proceedings of the IEEE international conference on computer vision, 2017.

33. Zhu X, Hu H, Lin S, Dai J. Deformable convnets v2: More deformable, better results. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

34. Wang Y, Zhang J, Kan M, Shan S, Chen X. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020.

35. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Paper presented at: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

36. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design. Paper presented at: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.