

Article

RT-DETR-DCEA: A Lightweight Citrus Defective Fruit Detection Algorithm for Complex Orchard Environments

Jihui Qiao ^{1,2}, Yuchen Sun ², Binyuan Zhong ^{2,3}, Lun Wang ^{1,2}, Siyu Li ², Hang Liu ^{2,3}, Youqing Chen ^{1,2} and Tong Li ^{2,3,*}

¹ College of Mechanical and Electrical Engineering, Yunnan Agricultural University, Kunming 650201, China

² The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Yunnan Agricultural University, Kunming 650201, China

³ School of Big Data, Yunnan Agricultural University, Kunming 650201, China

* Correspondence: tli@ynu.edu.cn

Abstract

Given the issues in natural orchard environments, such as large-scale variations of defective citrus fruits, weak texture boundaries, strong illumination changes, branch and leaf occlusion, and significant background interference, this paper constructs a lightweight detection model, RT-DETR-DCEA, based on RT-DETR-R18. This model is improved through four aspects: “fine-grained defective feature extraction—multi-scale feature fusion—up-sampling detail recovery—global feature interaction for noise suppression”. First, a Dynamic Hybrid Convolution Module (DIMB) is introduced into the backbone network, drawing on the ideas of Inception-style multi-branch depthwise convolution and MetaFormer residual mixing. It extracts local textures of various forms through square convolution, horizontal strip convolution, and vertical strip convolution, and utilizes dynamic branch weights to enhance the model’s adaptability to irregular defects such as lesions, mildew, and external damage. Second, a Content-Guided Attention Feature Fusion Network (CGAFN) is designed in the neck network, which achieves adaptive fusion of low-level detail features and high-level semantic features through channel attention, spatial attention, and pixel-level fusion weights. Next, a lightweight upsampling enhancement module called EUCB-SC is constructed, which introduces channel rearrangement and Shift spatial offset into the efficient upsampling convolutional structure to enhance the local spatial interaction capability of upsampled features with low parameter overhead. Finally, adaptive sparse self-attention is introduced into the AIFI module to form AIFI-ASSA, which suppresses irrelevant background interactions through a sparse attention branch and retains necessary contextual information through a dense attention branch. The experimental results demonstrate that on a dataset containing four categories of citrus images—healthy, diseased, moldy, and severely externally damaged—RT-DETR-DCEA achieves 92.1% Precision, 86.1% Recall, and 91.8% mAP@50, with a parameter count of 1.477×10^7 and an inference speed of 81 FPS. Compared with the original RT-DETR-R18 and various YOLO series models, this method strikes a favorable balance among detection accuracy, recall capability, and model lightweightness. This paper also discusses limitations such as data scale, ratio of private data, single training result, and insufficient validation on edge devices, providing a basis for subsequent cross-regional data validation and real-world deployment testing.

Keywords: RT-DETR; defective citrus fruit; lightweight detection; dynamic convolution; multi-scale feature fusion; adaptive sparse attention

Academic Editors: Dayang Liu and Yi Shi

Received: 8 June 2026

Revised: 27 June 2026

Accepted: 2 July 2026

Published: 3 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Citrus is a globally important economic fruit, and its appearance quality and defect status directly affect commodity grade, postharvest storage, and circulation safety [1]. During growth, harvesting, and transportation, fruits may develop defects such as lesions, mold, collision-induced dents, peel tears, and insect damage. In early stages, these defects often manifest as small areas, low contrast, and weak boundary textures. Reliance on manual inspection or sorting is not only inefficient but also susceptible to variations in experience, fatigue, and changes in ambient lighting [2]. With the advancement of smart orchard and automated sorting technologies, establishing a lightweight visual model capable of stably identifying multiple categories of citrus defects in complex natural scenes has clear engineering value.

In recent years, the YOLO series, DETR series, and improved models have been widely used for agricultural object detection [3]. YOLO-type models offer advantages such as fast inference speed and mature engineering deployment, but they are still prone to missed detections or false positives on small objects, occluded objects, and backgrounds with similar colors [4–12]. DETR-type models reduce manual design processes such as anchor boxes and non-maximum suppression through set prediction and the Transformer architecture, and possess strong global modeling capabilities [13–19]; among them, RT-DETR further improves the real-time performance of end-to-end detection and has been gradually applied to tasks such as fruit ripeness recognition, crop disease detection, and agricultural object perception in complex scenes. Existing studies have shown that improved RT-DETR can increase mAP and reduce model size or computational complexity in fruit detection. However, most works still focus on single targets, single environments, or obvious differences in target appearance. Their adaptability to scenes with multiple categories of defects, weak-texture lesions, and complex orchard backgrounds still needs to be strengthened [20–27].

For the task in this paper, the original RT-DETR-R18 still has three deficiencies. First, conventional convolutions or fixed-receptive-field convolutions have limited adaptability to variations in defect morphology, making it difficult to simultaneously capture multi-morphological features such as spot-like mildew, flaky lesions, and directional lacerations. Second, if cross-scale fusion in the neck network relies solely on addition, concatenation, or fixed weights, high-level semantics are prone to overshadowing low-level details, causing the edge information of small defects to decay during multi-layer propagation. Third, standard dense self-attention computes relationships between all spatial positions in complex orchard backgrounds, which may introduce noisy interactions from irrelevant regions such as foliage, shadows, highlights, and healthy fruit skin textures. The aforementioned problems collectively limit the detection performance of the model for small-scale, multi-category, and weak-boundary defects under lightweight constraints.

To address these issues, this paper constructs RT-DETR-DCEA. The main objectives of this paper include: (1) improving detection accuracy for small-scale and weak-boundary defects; (2) reducing model parameters while maintaining real-time inference capability; (3) enhancing robustness against occlusion, strong light, and similar-texture interference in complex orchard backgrounds; (4) verifying the independent contributions and combined effects of each module on performance improvement through ablation experiments. The main contributions of this paper are as follows.

1. A Dynamic Inception-style Mixed Convolution Block (DIMB) is proposed. By introducing input-adaptive branch weights into Inception-style multi-pattern depthwise

convolutions, DIMB enhances the backbone network's ability to represent irregular defect textures.

2. A Content-Guided Attention Feature Fusion Network (CGAFN) is designed to dynamically assign fusion weights to low-level detail features and high-level semantic features according to the input content, thereby improving the quality of multi-scale defect feature fusion.
3. A lightweight upsampling enhancement module, EUCB-SC, is constructed. By introducing channel shuffle and Shift-based spatial displacement into the efficient upsampling convolution of EUCB, EUCB-SC enhances local spatial information interaction with a low parameter overhead.
4. Introducing ASSA into the Attention-based Intra-scale Feature Interaction (AIFI) stage of RT-DETR forms AIFI-ASSA, which suppresses interference from low-relevance backgrounds while preserving essential global context.

2. Materials and Methods

2.1. Citrus Data Acquisition

The experimental study area was located at 24.25° N, 99.06° E, supported by the Yunnan Key Laboratory of Crop Production and Smart Agriculture, and combined with the Chu Orange production base in Longling County, Baoshan City, Yunnan Province, China. This study focused on the Chu orange cultivar from Yunnan Chushi Agriculture Co., Ltd., supplemented with a large amount of citrus data collected from other regions to improve the recognition accuracy and generalization capability of the model under different regional and environmental conditions. The images were manually collected using an Intel RealSense D455 camera manufactured by Intel Corporation, Santa Clara, CA, USA. Examples of the collected images are shown in Figure 1.



Figure 1. Images of citrus fruits with different defects and healthy citrus fruits.

Traditional citrus defect classification generally includes healthy fruit and three common defect categories: diseased fruit, moldy fruit, and fruit with severe external damage. Samples with extensive lesions exhibited clear gradient characteristics in lesion coverage. The affected area ranged from approximately one-third of the peel surface to nearly the entire fruit, with color gradually changing from light brown to dark brown. Some samples were also accompanied by peel shrinkage or gum exudation. Moldy samples were mainly characterized by fungal colonies, with mold spots appearing in white, green, or black. Their morphological patterns included punctate aggregation, patchy spreading, and annular diffusion, and some samples showed visible details of hyphae penetrating peel cracks. Samples with severe external damage included typical defects such as impact-induced depressions, peel tearing, and insect-bored holes. The damaged regions commonly showed tissue rupture, browning caused by juice loss, or discolored patches resulting from external contamination. By constructing a sample set containing typical morphological variations, the ability of deep learning models to recognize the spatial distribution

characteristics of lesions can be enhanced, thereby improving the detection and recognition performance for various defective fruits in complex field environments. This study focused on citrus fruits with lesions, mold infection, and severe external damage, with healthy citrus fruits used as controls. Each type of citrus fruit was annotated according to the above method, as shown in Figure 1.

2.2. Overall Architecture of RT-DETR-DCEA

This study adopts RT-DETR-R18 as the baseline model and introduces improvements at four key stages: feature extraction, cross-scale fusion, upsampling enhancement, and global feature interaction. The overall framework is illustrated in Figure 2. Specifically, DIMB is embedded into the backbone network to strengthen the local representation of defect textures with diverse morphological patterns. CGAFN is used to replace or enhance the original multi-scale fusion structure in the neck, allowing the fusion weights of features from different levels to be adaptively determined according to the input content. EUCB-SC is applied to the upsampling path to improve the recovery of local details in upsampled features. AIFI-ASSA is further introduced to optimize intra-scale feature interaction within AIFI, thereby reducing the interference of low-relevance regions in complex backgrounds during defect feature aggregation.

It should be noted that the contributions of the modules in this study mainly lie in their structural adaptation and integrated optimization for citrus defect fruit detection, rather than in reinterpreting existing modules as entirely independent new theories. To avoid overstatement, the relationships between the proposed modules and the relevant reference works are explicitly described in Section 2, and their practical contributions to the present task are further validated through ablation experiments.

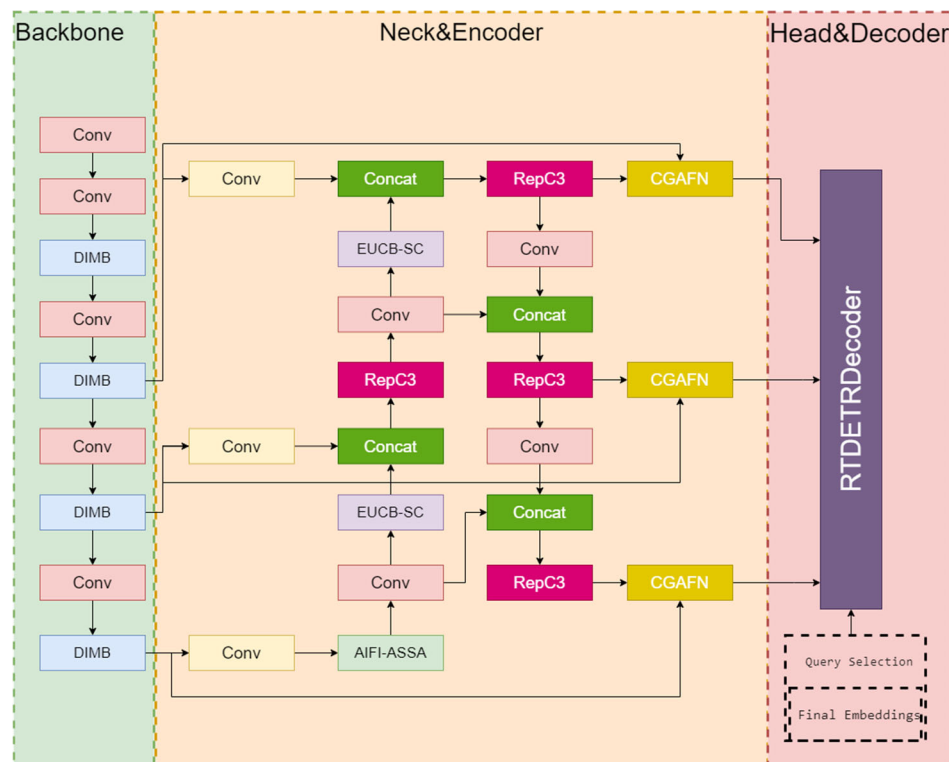


Figure 2. The improved RT-DETR network architecture diagram.

2.2.1. RT-DETR Baseline Model

RT-DETR is an end-to-end real-time object detection framework that extracts multi-scale features using a convolutional backbone network and performs set prediction through an efficient hybrid encoder and a Transformer decoder [28–33]. Compared with conventional anchor-based detectors, RT-DETR reduces the influence of anchor design and NMS post-processing on the detection pipeline, achieving a favorable balance between detection speed and accuracy [34–39]. In RT-DETR, the AIFI module is mainly used to model global interactions among high-level features, whereas the CCFM module is responsible for multi-scale cross-layer feature fusion. In this study, RT-DETR-R18 is selected as the lightweight baseline model because its parameter size and inference speed are more suitable for agricultural detection tasks on mobile or edge devices.

2.2.2. DIMB Dynamic Mixed Convolution Module

Citrus defects exhibit substantial morphological variations. Lesions may appear as spot-like, patch-like, or irregularly spreading regions; cracks and mechanical injuries may present directional edges; and moldy areas are often accompanied by fine-grained texture variations. A single fixed convolution kernel is therefore insufficient to simultaneously capture local spots, elongated textures, and directional edges. To address this issue, inspired by the multi-branch depthwise convolution design in InceptionNeXt and the general structural paradigm of “normalization–feature mixing–channel mapping–residual connection” in MetaFormer [40,41], this study constructs a Dynamic Inception Mixer Block (DIMB) to enhance the backbone network’s representation capability for citrus defect features with diverse morphologies. The structure of DIMB is shown in Figure 3.

The core component of DIMB is DynamicInceptionDWConv2d. This structure consists of three parallel depthwise convolution branches, namely a square depthwise convolution branch, a horizontal strip depthwise convolution branch, and a vertical strip depthwise convolution branch. Specifically, the square convolution branch is used to extract local textures and spot-like defect features; the horizontal strip convolution branch is designed to capture horizontally extended cracks, edges, or texture structures; and the vertical strip convolution branch is employed to extract vertically elongated damage edges and structural information. Let the input feature be defined as:

$$X \in R^{C \times H \times W}, \quad (1)$$

The three depthwise convolution branches can then be formulated as:

$$F_s s = DWConv_{k \times k}(X), \quad (2)$$

$$F_h = DWConv_{1 \times m}(X), \quad (3)$$

$$F_v v = DWConv_{m \times 1}(X), \quad (4)$$

where F_s , F_h and F_v denote the output features of the square convolution branch, horizontal strip convolution branch, and vertical strip convolution branch, respectively. DWConv represents the depthwise convolution operation, while $k \times k$, $1 \times m$ and $m \times 1$ indicate convolution kernels with different shapes.

Different from static multi-branch fusion strategies, DIMB further introduces a dynamic branch-weighting mechanism for convolutional branches. Specifically, global average pooling is first applied to extract global contextual information from the input feature. Then, a 1×1 convolution is used to generate three sets of branch weights, which are normalized by the Softmax function:

$$[\alpha_s, \alpha_h, \alpha_v] = Softmax(Conv_{1 \times 1}(GAP(X))), \quad (5)$$

where α_s , α_h , and α_v denote the dynamic weights corresponding to the three convolution branches, respectively, and $GAP(\cdot)$ represents the global average pooling operation.

The outputs of the three branches are fused through dynamic weighted aggregation:

$$Y = \alpha_s F_s + \alpha_h F_h + \alpha_v F_v, \quad (6)$$

This mechanism enables the model to adaptively adjust the contributions of different convolution branches according to the morphological variations of defect regions in the input image. When defects appear as local spots or small-area lesions, the square convolution branch can provide stronger local texture responses. In contrast, when defects appear as cracks, scratches, or directional edges, the horizontal and vertical strip convolution branches can more effectively capture directional structural information.

In DynamicIncMixerBlock, DynamicInceptionMixer serves as the spatial token mixer to achieve multi-morphological feature mixing along the spatial dimension. A convolutional GLU or 1×1 feed-forward mapping is used as the channel-mixing unit to enhance the nonlinear representation capability in the channel dimension. Meanwhile, this module incorporates residual connections, Layer Scale, and DropPath to improve training stability and reduce the risk of overfitting. The overall computational process can be formulated as:

$$X' = X + \text{DropPath}(\lambda_1 \cdot \text{Mixer}(\text{Norm}(X))), \quad (7)$$

$$Y = X' + \text{DropPath}(\lambda_2 \cdot \text{MLP}(\text{Norm}(X'))), \quad (8)$$

where λ_1 and λ_2 denote the learnable Layer Scale parameters, $\text{Mixer}(\cdot)$ represents the DynamicInceptionMixer, and $\text{MLP}(\cdot)$ denotes the channel-mapping unit.

It should be emphasized that the innovation scope of DIMB does not lie in re-proposing Inception-style depthwise convolution itself. Rather, its contribution lies in combining Inception-style multi-morphology depthwise convolution with an input-adaptive branch-weighting mechanism and adapting this design to the RT-DETR backbone network. This enables the model to better extract fine-grained textures, directional edges, and irregular lesion features in citrus defect fruit detection. In this way, DIMB preserves the efficiency of multi-branch depthwise convolution while improving the model's dynamic perception capability for different defect morphologies.

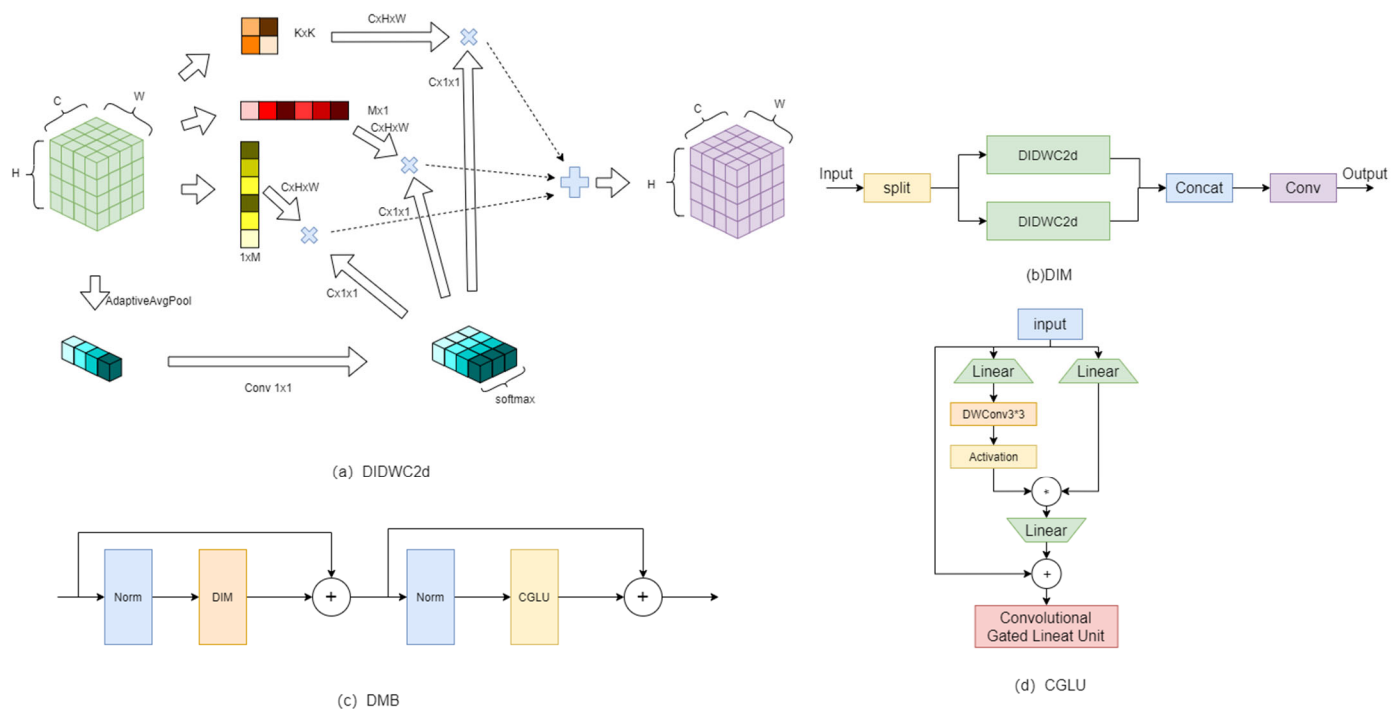


Figure 3. Network architecture diagrams of DIDWC2d, DIM, and DMB.

2.2.3. CGAFN Content-Guided Attention Feature Fusion Module

In complex orchard scenes, features at different levels contain markedly different types of information. Low-level features usually preserve rich edge, color, and texture details, making them sensitive to small lesions, mold boundaries, and mechanical damage edges. In contrast, high-level features contain stronger category semantics and contextual information, which helps suppress interference caused by branch and leaf occlusion, shadows, specular highlights, and complex backgrounds. If multi-scale features are directly fused by simple addition or concatenation, low-level details may be overwhelmed by high-level semantic information, thereby weakening the responses of small-scale defects, weak-texture lesions, and edge-damage regions.

To address the above issue, inspired by the content-guided attention mechanism and CGA-based mixup fusion strategy in DEA-Net [42,43], this study constructs a Content-Guided Attention Feature Network (CGAFN) to achieve content-guided cross-scale feature fusion in the detection neck, as shown in Figure 4. The core idea of this module is to adaptively adjust the contributions of two input features during fusion according to their channel semantics, spatial distributions, and pixel-level responses, rather than using a simple fusion strategy with fixed weights. Let the two features to be fused be denoted as:

$$F_x \in R^{C \times H \times W}, \quad (9)$$

$$F_y \in R^{C \times H \times W}, \quad (10)$$

where F_x and F_y denote features from different scales or different pathways, respectively.

The two feature branches are first fused as follows:

$$F_{init} = F_x + F_y, \quad (11)$$

Channel attention and spatial attention are then applied to the initially fused feature F_{init} to extract importance information along the channel and spatial dimensions, respectively:

$$A_c = CA(F_{init}), \quad (12)$$

$$A_s = SA(F_{init}), \quad (13)$$

where $CA(\cdot)$ denotes the channel attention operation, which is used to model the importance of different feature channels, while $SA(\cdot)$ denotes the spatial attention operation, which is employed to locate the response intensity of defect-related regions along the spatial dimension.

After summing the two attention maps, a coarse-grained attention guidance map is obtained:

$$A = A_c + A_s, \quad (14)$$

Based on this, a pixel attention module is further introduced to jointly model the initially fused feature and the coarse-grained attention map, generating fine-grained fusion weights:

$$W = \sigma(PA(F_{init}, A)), \quad (15)$$

where $PA(\cdot)$ denotes the pixel attention operation, $\sigma(\cdot)$ represents the Sigmoid activation function, and W is the final generated adaptive fusion weight. This weight can dynamically regulate the contribution ratio of the two input features across both channel and spatial positions.

The output feature of CGAFN can be formulated as:

$$F_{out} = Conv_{1 \times 1}(F_{init} + W \odot F_x + (1 - W) \odot F_y), \quad (16)$$

where $\odot$ denotes element-wise multiplication, and $Conv_{1 \times 1}$ is used for channel rearrangement and fused-feature projection. This fusion strategy not only preserves the initial information from the two feature branches, but also adaptively enhances the more discriminative feature components through the weight W . When defect boundaries are weak or texture details are prominent, the model can increase the contribution of low-level detailed features. In contrast, when the background is complex or the target is occluded, the model can rely more on high-level semantic features to reduce background interference.

Unlike conventional spatial attention, the fusion weights in CGAFN are not determined by a single spatial weight map alone, but are jointly generated by channel attention, spatial attention, and pixel attention. This design enables the module to address three key questions simultaneously: first, which channels are more important for defect recognition; second, which spatial regions are more likely to contain defect-related information; and third, how the contributions of the two feature branches should be allocated during pixel-level fusion. Therefore, CGAFN can regulate the multi-scale feature fusion process in a more fine-grained manner, thereby improving the representation capability of the neck network for small-scale lesions, mold textures, and mechanical damage boundaries.

In citrus defect fruit detection, defect regions are usually characterized by small areas, blurred boundaries, weak color variations, and strong background interference. Through its content-guided adaptive fusion mechanism, CGAFN helps enhance the feature responses of defect-related regions while reducing interference from irrelevant backgrounds, such as branches and leaves, shadows, specular highlights, and healthy peel textures. It should be emphasized that this study does not claim to propose the CGA mechanism for the first time. Instead, it transfers and adapts the existing content-guided fusion strategy to the neck network of RT-DETR to address the insufficient fusion of multi-scale defect features in agricultural scenarios.

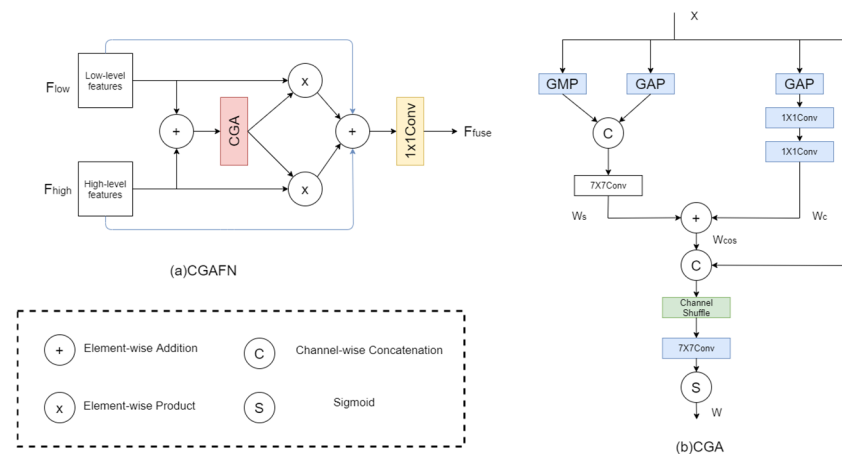


Figure 4. CGAFN Network Architecture Diagram.

2.2.4. EUCB-SC Efficient Upsampling and Shifted Channel Mixing Module

In the detection neck, upsampling is mainly used to restore the spatial resolution of high-level feature maps, enabling cross-scale fusion with low-level detailed features. The upsampling operations in the original neck structure usually rely on nearest-neighbor interpolation or bilinear interpolation. Although these methods can restore the feature map size, they do not inherently provide feature enhancement capability. For citrus defect fruit detection, small lesions, mold boundaries, and mechanical damage regions often exhibit small scales, weak edges, and fragmented textures. If further feature refinement is lacking after upsampling, the responses of defect boundaries may become insufficient, or local details may be lost.

To improve feature recovery during the upsampling stage, this study draws on the design concept of the Efficient Up-Convolution Block (EUCB) in EMCAD [44] and further introduces the Shift Channel Mix operation to construct the EUCB-SC module, as shown in Figure 5. The original EUCB achieves efficient feature recovery through upsampling, depthwise convolution, and 1×1 pointwise convolution, thereby enhancing local spatial representation after upsampling with low computational cost. Considering that depthwise convolution mainly performs spatial modeling within individual channels and has limited capability for inter-channel information interaction, this study further incorporates channel rearrangement and cyclic shift operations to enhance spatial information flow across different channels. Let the input feature be denoted as:

$$X \in R^{C \times H \times W}, \quad (17)$$

EUCB-SC first upsamples the input feature by a factor of 2 and then enhances the local spatial representation of the upsampled feature through depthwise convolution:

$$F_u = DWC(U_p(X)), \quad (18)$$

where $Up(\cdot)$ denotes the upsampling operation, $DWC(\cdot)$ represents the depthwise convolution operation, and F_u denotes the feature obtained after upsampling and depthwise convolution.

The feature is then rearranged along the channel dimension and divided into four sub-feature groups:

$$F_u = [F_1, F_2, F_3, F_4], \quad (19)$$

where F_1 , F_2 , F_3 and F_4 denote the four channel subgroups, respectively.

To enhance information interaction among neighboring spatial positions, the four feature groups are cyclically shifted forward or backward along the height and width dimensions, respectively:

$$F_1' = \text{Roll}(F_1, +s, H), \quad (20)$$

$$F_2' = \text{Roll}(F_2, -s, H), \quad (21)$$

$$F_3' = \text{Roll}(F_3, +s, W), \quad (22)$$

$$F_4' = \text{Roll}(F_4, -s, W), \quad (23)$$

where s denotes the shift step size; H and W represent the height and width dimensions, respectively; and $\text{Roll}(\cdot)$ denotes the cyclic shift operation.

The four shifted feature groups are then concatenated again along the channel dimension, followed by a 1×1 convolution for channel fusion and feature projection:

$$F_{sc} = \text{Concat}(F_1', F_2', F_3', F_4'), \quad (24)$$

$$Y = \text{Conv}_{1 \times 1}(F_{sc}), \quad (25)$$

where $\text{Concat}(\cdot)$ denotes the channel-wise concatenation operation, and Y represents the output feature of EUCB-SC.

The Shift Channel Mix operation does not introduce additional learnable parameters. Instead, it enables feature channels to acquire information from neighboring spatial positions through spatial rearrangement. Compared with standard depthwise convolution, this operation can expand the local information interaction range with extremely low computational overhead. Compared with ordinary upsampling, EUCB-SC not only restores the spatial resolution but also further enhances the local representation capability of the upsampled features. By combining channel rearrangement with 1×1 convolution, this module can compensate for the insufficient channel interaction of depthwise convolution, allowing information from different channels to be further fused.

For citrus defect fruit detection, EUCB-SC helps improve the upsampled representation quality of small defect boundaries, weak-texture lesions, and local mechanical damage regions. Particularly under conditions of fruit occlusion, uneven illumination, and strong interference from background branches and leaves, this module can enhance local detail information with low parameter and computational overhead, thereby providing a more stable feature basis for subsequent cross-scale feature fusion. It should be noted that the contribution of EUCB-SC does not lie in re-proposing upsampling or the shift operation itself. Instead, it combines an efficient up-convolution block with a parameter-free shifted channel mixing mechanism and adapts it to the neck network of RT-DETR, thereby meeting the dual requirements of detail preservation and lightweight inference in citrus defect detection.

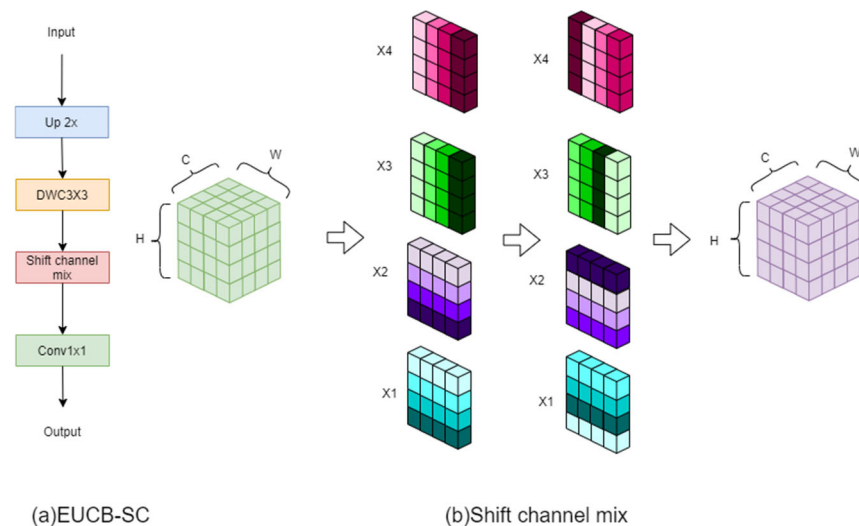


Figure 5. Network structure diagram of EUCB-SC module.

2.2.5. AIFI-ASSA Adaptive Sparse Feature Interaction Module

The AIFI module in RT-DETR is mainly used to enhance global interaction within single-scale features. Through the self-attention mechanism, AIFI can model long-range dependencies among different spatial positions, thereby improving the representation capability of high-level semantic features. However, in citrus defect fruit detection, image backgrounds are usually complex and often contain defect-irrelevant information, such as branch and leaf occlusion, shadows, specular highlights, and natural peel textures. If dense self-attention is directly used to compute the correlations among all spatial positions, low-relevance background regions will also participate in feature aggregation. This may introduce noisy interactions and weaken the responses of small lesions, moldy regions, and weak-texture defects.

To alleviate the above issue, this study draws on the design concept of Adaptive Sparse Self-Attention (ASSA) in the Adaptive Sparse Transformer [45] and introduces it into the AIFI module of RT-DETR, thereby constructing the AIFI-ASSA adaptive sparse feature interaction module, as shown in Figure 6. While preserving global contextual modeling capability, this module suppresses redundant interactions from low-relevance regions through sparse attention, thereby improving the discriminative representation of defect features under complex orchard backgrounds. Let the input feature be denoted as:

$$X \in R^{C \times H \times W}, \quad (26)$$

The input feature is then mapped into the query, key, and value matrices:

$$Q = XW_Q, \quad (27)$$

$$K = XW_K, \quad (28)$$

$$V = XW_V, \quad (29)$$

where W_Q , W_K and W_V denote the linear projection parameters for the query, key, and value, respectively.

The attention scores are calculated based on the correlation between Q and K :

$$S = \frac{QK^T}{\sqrt{d}} + B, \quad (30)$$

where d denotes the feature dimension, B represents the relative positional bias, and S is the attention score matrix. To clarify the dimensions of each tensor, let the spatial size

of the input feature map be $N = H \times W$. The query matrix Q , key matrix K , and value matrix V all have dimensions $\mathbb{R}^{N \times d}$. Accordingly, the attention score matrix S has dimensions $\mathbb{R}^{N \times N}$, and the relative position bias B also has dimensions $\mathbb{R}^{N \times N}$. The two are added element-wise and subsequently fed into the sparse and dense attention branches.

In the dense attention branch, the Softmax function is applied to normalize the attention scores, thereby preserving relatively complete contextual information:

$$DSA = \text{Softmax}(S), \quad (31)$$

Dense attention can sufficiently model global spatial relationships, but it may also introduce redundant interactions from low-relevance background regions. To address this issue, the sparse attention branch employs the squared ReLU function to suppress low-response attention relationships.

$$SSA = \text{ReLU}^2(S), \quad (32)$$

where SSA denotes the sparse self-attention branch, which can reduce the influence of low-relevance query–key relationships and encourage the model to focus more on defect-related regions.

To avoid the loss of useful contextual information caused by using sparse attention alone, AIFI-ASSA further introduces learnable weights to adaptively fuse the sparse attention branch and the dense attention branch. Let a_1 and a_2 denote the learnable parameters; the weights of the two branches can be expressed as:

$$w_i = \frac{\exp(a_i)}{\sum_{j=1}^2 \exp(a_j)}, i = 1, 2, \quad (33)$$

where w_1 and w_2 denote the normalized weights of the sparse attention branch and the dense attention branch, respectively.

The final attention output is formulated as:

$$A = (w_1 \cdot SSA + w_2 \cdot DSA)V, \quad (34)$$

This dual-branch fusion mechanism can achieve a balance between information preservation and noise suppression. The dense attention branch is used to maintain the necessary global contextual modeling capability, preventing information interactions among important regions from being excessively weakened. In contrast, the sparse attention branch is used to reduce the interference of low-relevance background regions, such as branches and leaves, shadows, specular highlights, and healthy peel textures, during defect feature aggregation.

In the specific implementation, AIFI-ASSA retains the basic structure of the Transformer encoder, including residual connections, normalization operations, and the feed-forward network. The input feature is first processed by AdaptiveSparseSA to perform adaptive sparse feature interaction, and is then added to the original input through a residual connection followed by normalization:

$$X' = \text{Norm}(X + \text{ASSA}(X)), \quad (35)$$

Channel mapping and nonlinear transformation are then performed through a feed-forward network composed of 1×1 convolutions:

$$Y = \text{Norm}(X' + \text{FFN}(X')), \quad (36)$$

where $\text{ASSA}(\cdot)$ denotes the adaptive sparse self-attention operation, and $\text{FFN}(\cdot)$ represents the feed-forward network. In this implementation, the feed-forward network consists of a 1×1 convolution, an activation function, and another 1×1 convolution, which are used to perform information mapping along the channel dimension.

It should be noted that this study does not fully introduce all components of the Adaptive Sparse Transformer, such as the FRFN module. Instead, only the ASSA attention mechanism is introduced and adapted to the AIFI feature interaction module of RT-DETR. Therefore, the contribution of AIFI-ASSA lies in the adaptive sparsification of the dense feature interaction mechanism in the original AIFI module for citrus defect fruit detection. This design reduces irrelevant token interactions in complex orchard backgrounds and improves the feature representation capability for small-scale defects and weak-texture regions. This module does not claim to re-propose the sparse attention mechanism; rather, it applies the existing adaptive sparse attention concept to optimize high-level feature interaction in agricultural object detection scenarios.

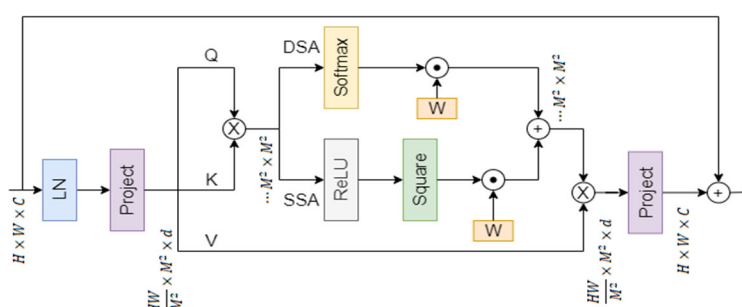


Figure 6. AIFI-ASSA Network Architecture Diagram.

3. Results

3.1. Dataset and Experimental Setup

The citrus fruit disease image dataset used in this study was primarily sourced from the open Kaggle website, with a small portion collected from the Chu Cheng orange base in Longling County, Baoshan City, Yunnan Province, China (24.25° N, 99.06° E), with an approximate ratio of 6:4. The dataset comprises 1146 images of citrus fruits, categorized into healthy and three common defective states. The image backgrounds include both complex natural environments and simple artificial setups, with natural backgrounds presenting challenges such as leaf and branch occlusion, fruit overlapping, strong light reflection, and backlighting. All images maintain a uniform resolution of 640 × 640 pixels. The dataset includes 338 images of healthy citrus, 323 images of diseased citrus, 248 images of moldy citrus, and 237 images of citrus with severe external damage. To improve the recognition accuracy and generalization ability of the enhanced RT-DETR-DCEA model for citrus defective fruits across different environments, a split-before-augmentation strategy was adopted for the dataset images to avoid data leakage caused by near-duplicate samples entering different data subsets. The images were annotated using the Labellmg tool and divided into training, validation, and test sets with a distribution of 7:2:1, after which data augmentation strategies were applied. These strategies featured horizontal and vertical rotations as well as brightness adjustments, ultimately generating a total of 6756 augmented images. This dataset will be used for experiments including model training, parameter optimization, and prediction result comparison to evaluate model performance. The visualization of the citrus dataset is shown in Figure 7. Figure 7A presents the categories and corresponding label information, where healthy citrus is labeled as A, diseased citrus as B, moldy citrus as C, and citrus with severe external damage as D. Figure 7B displays the dimensions of the label boxes, while Figure 7C illustrates the distribution of center point positions. Figure 7D provides relevant information on the size distribution of citrus fruits, and Figure 7E offers a detailed explanation of the label content. As can be observed, the samples include both large-area defects and localized

small lesions and weak-texture targets, and the distribution of target center positions is relatively dispersed, indicating that the model needs to possess both multi-scale feature modeling and complex background suppression capabilities simultaneously.

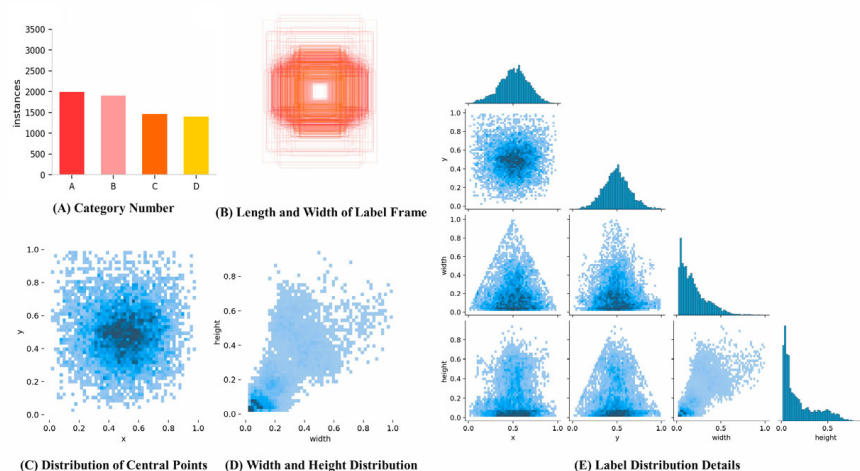


Figure 7. Visualization of the citrus dataset.

To ensure the comparability and reproducibility of the experimental results, all experiments in this study were conducted under a unified hardware and software environment. The experimental platform was based on the Windows 11 operating system and was equipped with an AMD Ryzen 9 7945HX processor, an NVIDIA GeForce RTX 4070 GPU, and 32 GB RAM. The programming language was Python 3.10.10, and the deep learning framework was PyTorch 1.13.1. PyCharm was used as the development environment, and the Compute Unified Device Architecture (CUDA) version was 12.0. All comparative models adopted the same dataset partitioning strategy, image input size, and basic training parameters. The input images were uniformly resized to (640 $\times$ 640), the batch size was set to 4, and the number of training epochs was set to 500. The initial learning rate was set to 0.01, and stochastic gradient descent (SGD) was used as the optimizer, with a momentum of 0.937 and a weight decay of 0.0005. During the first three epochs, a warm-up strategy was adopted, in which the learning rate gradually increased from a small value to the initial learning rate to reduce the influence of unstable parameter updates at the early training stage. Subsequently, the default learning rate scheduling strategy of the framework was used to decay the learning rate progressively during training, thereby improving the convergence stability in the later stage. During training and validation, the random seed was fixed at (seed = 0), and deterministic = True was enabled to reduce random errors caused by model initialization, data loading, data augmentation, and the non-determinism of CUDA operators. All ablation and comparative experiments in this study were conducted under this randomness-control setting. The results reported in the tables correspond to a single training run under a fixed dataset split and a fixed random seed. This setting improves experimental reproducibility; however, it is not equivalent to statistical validation using multiple random seeds. Therefore, the need for repeated experiments and uncertainty analysis is further discussed as a limitation and a direction for future work. No modifications or optimizations are made to the loss function in this work. The loss function, as well as the associated weights and other parameters, remains identical to that of the original RT-DETR.

Furthermore, this paper configures the post-processing pipeline according to the original inference paradigm of each detection framework. RT-DETR is an end-to-end real-time detection framework, and one of its core advantages lies in reducing the reliance on

Non-Maximum Suppression (NMS) through the DETR-style set prediction mechanism, thereby avoiding the negative impact of NMS post-processing on real-time detection speed and stability. The original RT-DETR paper explicitly states that, as a real-time end-to-end detector, it can eliminate the detrimental effects of NMS post-processing on real-time object detection. Therefore, when evaluating RT-DETR and its improved variants, this paper preserves their end-to-end inference paradigm without additionally introducing conventional NMS post-processing. For the YOLO-series comparison models, including YOLOv8m, YOLOv11m, and YOLOv12m, the default Ultralytics inference settings are uniformly adopted: the confidence threshold is set to 0.25, the IoU threshold to 0.70, and the input size to 640×640 . Specifically, the confidence threshold is used to filter out low-confidence predictions, and the IoU threshold is applied in the NMS stage to remove highly overlapping redundant bounding boxes. These settings are consistent with the official Ultralytics default inference configuration, which helps minimize the influence of manual parameter tuning on the comparison results across different models. It should be noted that the post-processing mechanisms of RT-DETR and the YOLO-series models are different. Therefore, in the comparative experiments, this paper maintains the original inference paradigm of each model while unifying the data split, input size, training configuration, and evaluation metrics, ensuring that the comparison results are both consistent with each model's design logic and experimentally comparable.

3.2. Model Evaluation Metrics

This paper selects precision (P), recall (R), average precision (AP), mean average precision (mAP), model parameters, and frames per second (FPS) as core evaluation metrics to comprehensively assess the detection performance and operational efficiency of the model. Precision (P) is defined as the ratio of correctly predicted positive samples to all samples predicted as positive, reflecting the accuracy of prediction results; recall (R) is the ratio of correctly identified positive samples to the total actual positive samples, demonstrating the model's capability to capture positive samples. As key metrics for evaluating object detection performance, mAP@50 uses an Intersection over Union (IoU) threshold of 0.50, whereas mAP@50:95 averages AP over IoU thresholds from 0.50 to 0.95. Therefore, mAP@50 reflects detection performance under a relatively lenient localization criterion, while mAP@50:95 provides a stricter evaluation of localization robustness. The specific calculation is shown in the following formula.

$$P = \frac{TP}{TP + FP}, \quad (37)$$

$$R = \frac{TP}{TP + FN}, \quad (38)$$

$$AP = \int_0^1 P(R) dR, \quad (39)$$

$$mAP = \frac{\sum_{l=1}^C AP(l)}{C}, \quad (40)$$

3.3. Model Performance Analysis and Ablation Study

Based on in-depth optimization and improvement of the RT-DETR model, this paper proposes an enhanced model. The expanded dataset is divided into training, testing, and validation sets for training, with analysis conducted on the results of each improvement. The Grad CAM heatmap results of various enhancement modules are shown in Figure 8, the comparison of effects before and after improvement is illustrated in Figure 9, and the ablation experiments of the improved model are presented in Table 1.

Table 1. Ablation study

DIMB	CGAFN	EUCB-SC	AIFI-ASSA	P (%)	R (%)	mAP@50 (%)	Params (10 ⁷)	FPS (Frame/s)
×	×	×	×	85.7	76.7	85.2	2.008	67
√	×	×	×	87.8	78.9	85.6	1.568	79
×	√	×	×	87.2	81.2	87.9	1.989	85
×	×	√	×	88.6	83.8	88.8	1.921	91
×	×	×	√	87.4	81	86.5	2.002	89
√	√	×	×	90.3	84.4	88.6	1.563	82
√	×	√	×	91.1	83	89	1.514	90
√	×	×	√	89.8	84.7	88.9	1.587	87
√	√	√	×	91.7	85.6	91.2	1.489	84
√	√	√	√	92.1	86.1	91.8	1.477	81

Note: √, uses this module; ×, does not use this module.

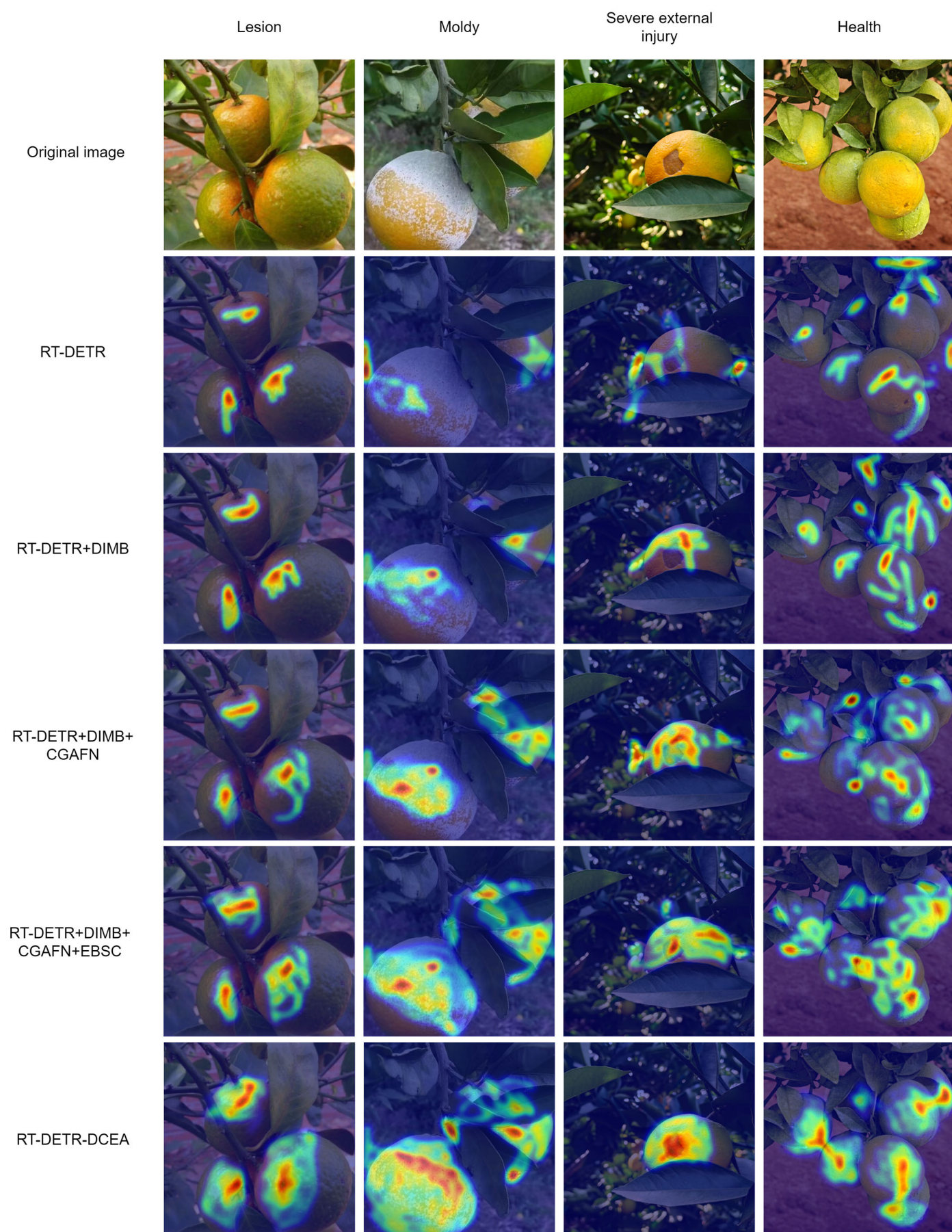


Figure 8. Comparison of heatmaps from different improved models.

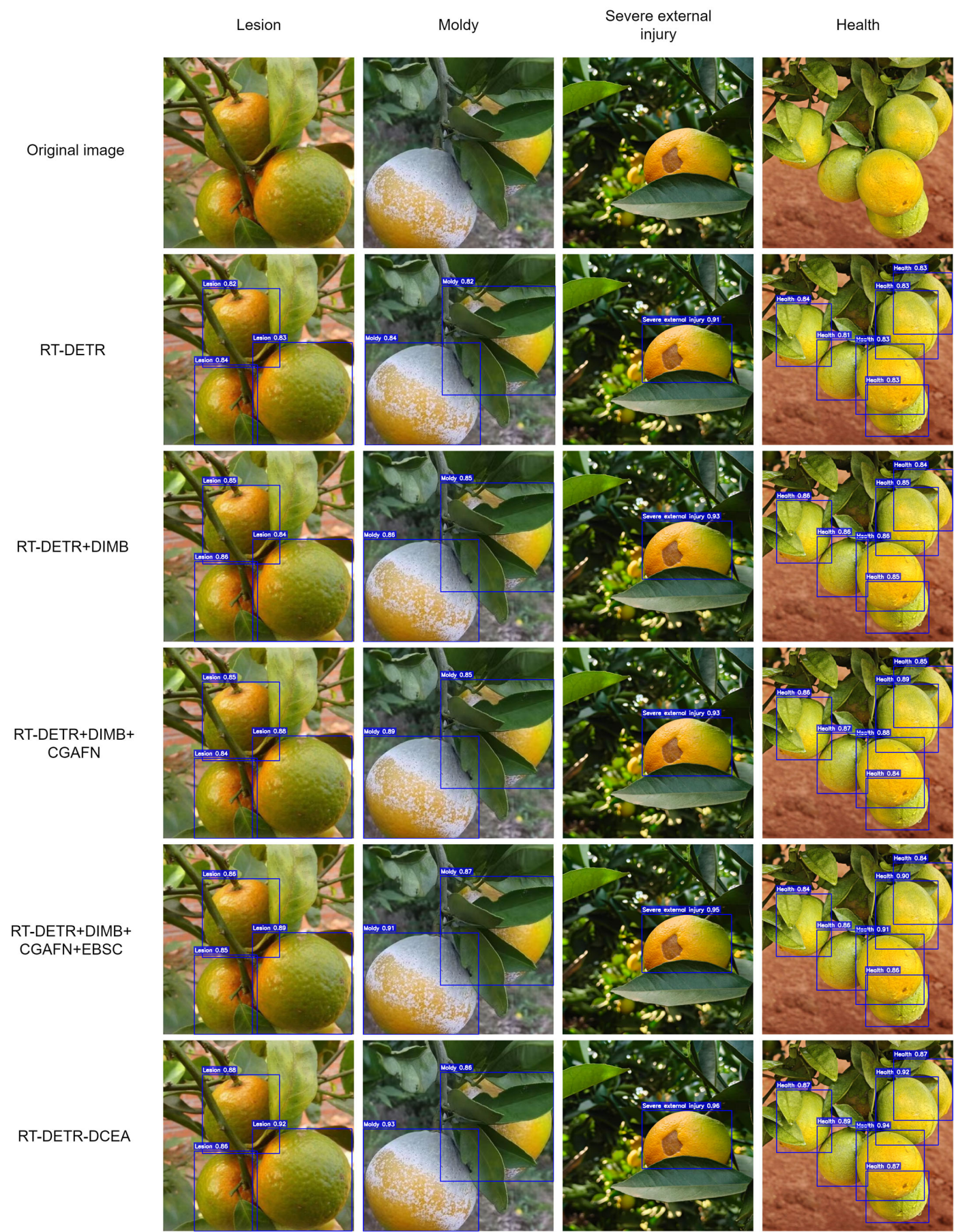


Figure 9. Comparison of effects before and after RT-DETR improvements.

The ablation experiments demonstrate that each proposed module improves the detection performance of the baseline model from different perspectives. When DIMB was added alone, Precision increased from 85.7% to 87.8%, indicating that the dynamic multi-morphological convolution enhances the model's ability to discriminate defect textures and irregular edges. The individual addition of CGAFN raised Recall to 81.2%, suggesting that the content-guided cross-scale feature fusion helps strengthen the complementary relationship between low-level details and high-level semantics, thereby reducing missed detections. Under the single-module setting, EUCB-SC achieved an mAP@50 of 88.8% and an FPS of 91, demonstrating that this module can improve the detail representation of upsampled features with low computational overhead. AIFI-ASSA improved both Recall and FPS, indicating that the sparse-dense attention fusion contributes to suppressing irrelevant feature interactions in complex backgrounds and enhancing the efficiency of information aggregation in high-level features. When DIMB, CGAFN, and EUCB-SC were combined, the model attained an mAP@50 of 91.2%, an improvement of 6.0 percentage points over the baseline. After further incorporating AIFI-ASSA, the final model achieved a Precision of 92.1%, Recall of 86.1%, and mAP@50 of 91.8%. Meanwhile, compared with the baseline model, the final model's parameter count decreased from 2.008×10^7 to 1.477×10^7 , and the FPS increased from 67 to 81, indicating that the proposed improvements achieve accuracy gains not by substantially increasing computational redundancy, but by striking a favorable overall trade-off among detection accuracy, inference speed, and model size. It is noteworthy that the ablation experiment comparing dynamic weighting and static average fusion in DIMB showed that dynamic weighting yielded a 0.7 percentage point improvement in mAP@50 over static averaging, validating the effectiveness of the input-adaptive mechanism (see Table A1 in the Appendix A).

3.4. Comparative Experiments on Performance of Different Network Models

To further evaluate the comprehensive performance of the proposed RT-DETR-DCEA model in citrus defective fruit detection tasks, a comparative analysis was conducted with several mainstream object detection models, including YOLOv8m [46], YOLOv11m [47], YOLOv12m [48], and the original RT-DETR model. All comparative experiments were performed under the same dataset, identical training-validation-test splits, and hyperparameter settings to ensure fairness in comparison. The performance comparison is shown in Table 2.

Table 2. Comparison results of different network models in citrus disease detection.

Model	P (%)	R (%)	mAP@50 (%)	mAP@50:95 (%)	Params (10^7)	FPS (Frame/s)
YOLOv8m	86.1	73.8	84.4	71.8	2.590	84
YOLOv11m	85.1	74.5	83.6	74.7	2.011	82
YOLOv12m	84.7	74.9	83.8	73.6	2.020	66
RT-DETR	85.7	76.7	85.2	74.8	2.008	67
RT-DETR-DCEA	92.1	86.1	91.8	80.1	1.477	81

Analysis of the data in Table 2 shows that RT-DETR-DCEA outperforms all compared models on key evaluation metrics, including Precision, Recall, mAP@50, and mAP@50:90, demonstrating the strong overall performance of the proposed method for on-tree citrus fruit recognition and defective fruit detection. Compared with the original RT-DETR-R18, RT-DETR-DCEA achieves improvements of 6.4 percentage points in Precision, 9.4 percentage points in Recall, and 6.6 percentage points in mAP@50, while the number of parameters is further reduced by approximately 26.4% and the FPS increases to 81. These results indicate that the proposed improvements not only enhance detection accuracy, but also improve the lightweight nature and inference efficiency of the model.

Compared with YOLOv8m, RT-DETR-DCEA has a slightly lower FPS, yet its Precision, Recall, and mAP@50 are higher by 6.0, 12.3, and 7.4 percentage points, respectively. This suggests that the model possesses stronger recognition capability for small-scale defects, weak-texture lesions, and irregular damage regions in complex orchard scenes, and achieves a favorable balance among detection accuracy, model size, and inference speed. As shown in Figure 10. It should be noted that the current comparative experiments mainly cover YOLO-series models and the original RT-DETR-R18, and have not yet included more strong baselines from the DETR family, such as Deformable DETR, Efficient DETR, or other lightweight RT-DETR variants. Therefore, future studies can incorporate additional DETR-type models under the same data split, training epochs, and evaluation scripts to further validate the relative advantages and generalization performance of RT-DETR-DCEA within the end-to-end detection framework.



Figure 10. Performance comparison of different network models.

Figures 11 and 12 shows the changes in training-process metrics for each model under a fixed random seed of (seed = 0) and the same training configuration. It can be observed that the mAP curve of RT-DETR-DCEA remains at a relatively high level in the later training stage, with comparatively small overall fluctuations, indicating that the model exhibits a stable convergence trend under the current dataset split and training settings. Compared with some comparison models, RT-DETR-DCEA achieves higher detection performance in the later training stage, which is generally consistent with the test results reported in Table 2. This suggests that the proposed structural improvements contribute to improved detection performance on the current test set. It should be noted that Figure 11 only reflects the training convergence process under a fixed random seed and the current dataset split. It cannot independently demonstrate the model's generalization capability across regions, seasons, or scenarios. The statistical stability and external generalization ability of the model still need to be further evaluated through repeated experiments with multiple random seeds, external test sets, and cross-regional data validation.

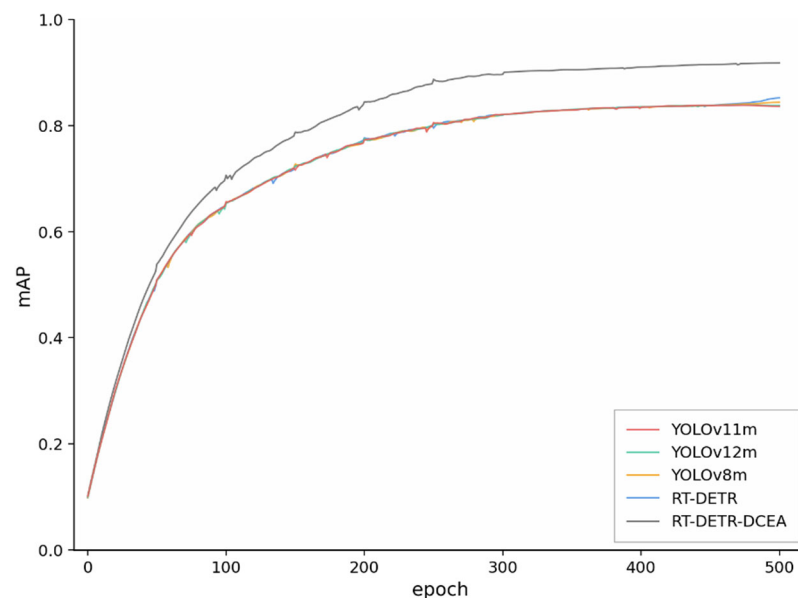


Figure 11. Comparison curves of different models.

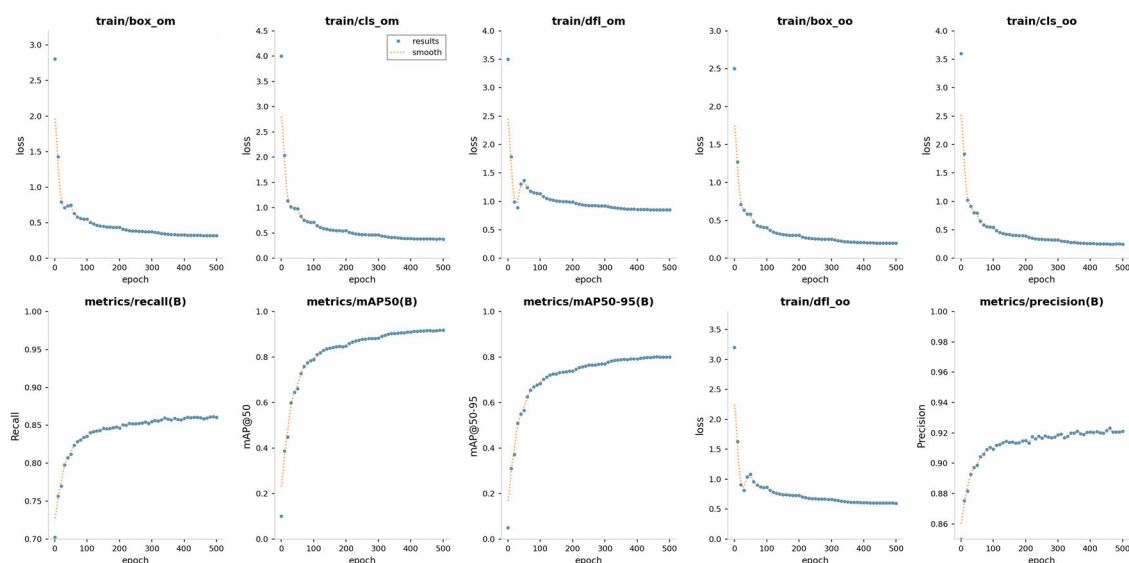


Figure 12. Loss curves.

4. Discussion

The performance improvement of RT-DETR-DCEA stems from the synergistic effects of four enhancement modules: DIMB strengthens the backbone's local representation of multi-morphological defect textures, CGAFN improves the adaptive fusion of cross-scale features in the neck, EUCB-SC enhances detail recovery in the upsampling stage with low parameter overhead, and AIFI-ASSA reduces background noise interference in high-level feature interaction. These four modules target distinct bottlenecks in feature extraction, fusion, recovery, and interaction, respectively, and together yield complementary gains.

From the perspective of computational complexity, the proposed modules improve accuracy without imposing a significant additional computational burden. For an input feature map of size $C \times H \times W$, the input-adaptive dynamic weight generation in DIMB introduces only about $3C$ extra parameters and approximately $3 \cdot C \cdot H \cdot W$ extra FLOPs, the multi-attention mechanism in CGAFN adds about $4C^2$ parameters and $8 \cdot C \cdot H \cdot W$ FLOPs, the Shift Channel Mix operation in EUCB-SC introduces no learnable parameters and only incurs roughly $2 \cdot C \cdot H \cdot W$ shift computations, and the dual-branch attention fusion in AIFI-ASSA adds about $2C^2$ parameters and $4 \cdot N \cdot d$ FLOPs (where $N = H \cdot W$). Overall, RT-DETR-DCEA incurs approximately +12% additional FLOPs compared to the original RT-DETR-R18, while reducing the number of parameters by 26.4% (from 2.008×10^7 to 1.477×10^7). This indicates that the improved performance is not obtained by substantially increasing computational redundancy but rather through precise structural optimization that achieves a favorable balance between accuracy and efficiency.

To quantitatively verify the effectiveness of the dynamic weighting mechanism in DIMB over static average fusion, we conducted a supplementary ablation experiment: the adaptive weights in DynamicInceptionDWConv2d were replaced with equal-weight average fusion $Y = (F_s + F_h + F_v)/3$, while all other configurations were kept strictly identical. The results show that the static average fusion achieved an mAP@50 of 84.9%, whereas the proposed dynamic weight fusion reached 85.6%, yielding an absolute improvement of 0.7 percentage points, with negligible additional overhead in parameter count (1.568M vs. 1.567M) and inference speed (79 FPS vs. 80 FPS). This finding confirms that the strong assumption implicit in static averaging—that “features of different morphological defects are equally important”—cannot adapt to the morphological diversity of citrus defects. By generating adaptive weights through global context, DIMB enables the square, horizontal strip, and vertical strip branches to dynamically modulate their contributions according to the local morphology of defects in the input image, thereby more accurately capturing fine-grained differences such as spot-like mold and directional cracks. This ablation experiment provides quantitative support for the input-adaptive mechanism of DIMB and aligns with the aforementioned complexity analysis—achieving a meaningful accuracy gain at a very low additional cost.

Although the overall detection performance is satisfactory, several typical failure modes persist in complex orchard environments. The causes of these failures can be analyzed at both the image feature level and the model mechanism level. As shown in Figure 13, First, false positives induced by illumination interference. Strong light reflection areas and light-colored mold spots exhibit substantial feature overlap in both color and brightness spaces, with closely distributed RGB means and luminance gradients, making it difficult for the model to distinguish them based solely on color and brightness cues, thereby misclassifying highlight regions as defects. Second, false negatives caused by severe occlusion. When foliage occlusion is heavy or only a small portion of the defect is visible, the proportion of valid defect pixels is insufficient, and the local texture features extracted by the backbone become too sparse to support adequate responses during subsequent cross-scale fusion and attention interaction, leading to missed detections. Third, category confusion resulting from texture similarity. The natural texture of healthy fruit skin and

the boundary of early-stage lesions exhibit similar local gradient distributions in regions with gradual color changes. Particularly at the initial stage of a lesion, their edge magnitudes and orientation histograms are highly alike, which tends to cause misclassification. Fourth, localization shift due to overlapping fruits. In scenarios with multiple overlapping fruits, where fruit boundaries occlude each other or the background color resembles the skin color, edge responses of different fruits interfere spatially, making it difficult for the bounding box regression branch to accurately delineate object boundaries and thus producing slight offsets. The above phenomena indicate that citrus defect detection performance is influenced not only by model architecture but also jointly by multiple factors such as illumination conditions, occlusion severity, defect development stage, skin texture variation, and background complexity. In essence, they reflect a considerable “domain gap” between image features and defect semantics, which cannot be fully bridged by network structure improvements alone. A coordinated approach encompassing data augmentation, feature learning, and post-processing strategies is required. To address these issues, future work may introduce illumination-invariant feature extraction, occlusion-aware training strategies, and boundary refinement modules during the training phase.

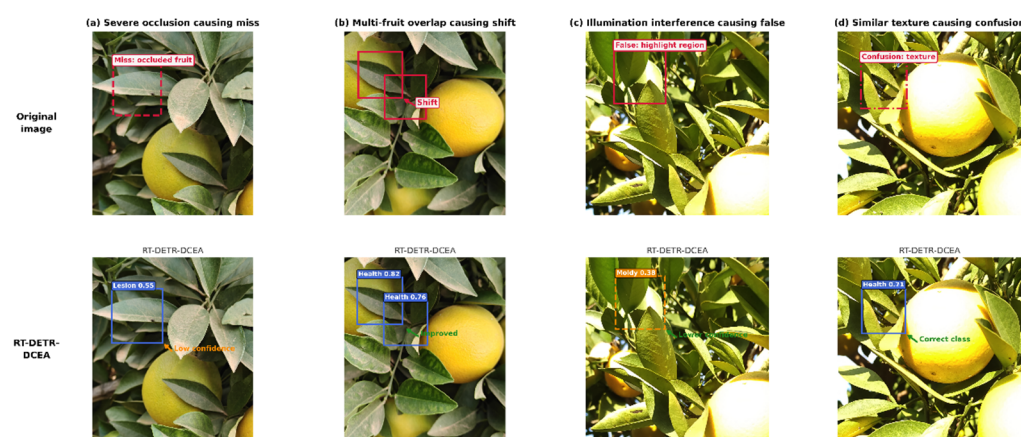


Figure 13. Failure case analysis figure.

This study has the following limitations. First, the dataset contains only 1146 original images, representing a limited data scale. Data augmentation cannot be equated with independently collected large-scale real samples, and the model’s generalization ability needs to be validated across more regions, cultivars, and seasonal conditions. Second, part of the self-collected data has not been made publicly available due to licensing restrictions of the collection sites, which hinders full reproducibility by third parties. Third, the current experiments are based on a fixed data split and a single training run; the mean, standard deviation, and confidence intervals over multiple random seeds have not been systematically reported. Fourth, on-device latency and power consumption tests have not yet been conducted on edge platforms such as Jetson or Ascend, and the model’s edge deployment performance awaits verification. Fifth, the comparative experiments have not sufficiently included additional strong baselines from the DETR family, such as Deformable DETR and Efficient DETR; the comparison scope should be expanded in the future.

Future research will proceed along the following directions. First, construct a larger and more diverse dataset of citrus defective fruit, covering multiple production regions, cultivars, maturity stages, and illumination conditions. Second, improve the experimental evaluation framework by adding multi-seed repeated experiments, per-class AP, mAP@75, precision–recall curves, confusion matrices, and hard-sample analysis. Third, subject to obtaining full data authorization, further provide an image-based analysis of typical failure cases with geographical annotations to support more transparent model

robustness assessment. Fourth, combine deployment techniques such as pruning, knowledge distillation, quantization, and TensorRT acceleration to deploy the model on orchard inspection platforms, edge terminals, or automated sorting lines, and evaluate detection stability, inference latency, and power consumption under long-term operating conditions [49–51].

5. Conclusions

This paper proposes a lightweight detection model, RT-DETR-DCEA, based on RT-DETR-R18, for on-tree citrus fruit recognition and defective fruit detection in complex orchard environments. To address the challenges of citrus defects, such as small scales, weak boundaries, fine and fragmented textures, large morphological variations, and strong background interference, the original RT-DETR-R18 is improved from four aspects: backbone feature extraction, neck feature fusion, upsampling feature enhancement, and global feature interaction. Specifically, the Dynamic Mixed Convolution Block (DIMB) is introduced to enhance the modeling of local defect textures and directional edges, the Content-Guided Attention Feature Fusion Network (CGAFN) is employed to improve the adaptive fusion between low-level details and high-level semantics, the Efficient Upsampling Block with Shift Channel Mix (EUCB-SC) is designed to strengthen the recovery of fine defect details during upsampling, and the AIFI-ASSA (Adaptive Sparse Feature Interaction) module is utilized to reduce the interference of irrelevant background noise in defect feature representation during high-level feature interaction.

Experimental results on a dataset comprising four categories—healthy, diseased, moldy, and severely externally damaged fruit—show that RT-DETR-DCEA achieves favorable detection performance. The final model attains a Precision of 92.1%, Recall of 86.1%, and mAP@50 of 91.8%, with 1.477×10^7 parameters and an FPS of 81. Compared with the original RT-DETR-R18, RT-DETR-DCEA improves Precision by 6.4 percentage points, Recall by 9.4 percentage points, and mAP@50 by 6.6 percentage points, while reducing the parameter count by approximately 26.4% and further increasing inference speed. Compared with competing models such as YOLOv8m, YOLOv11m, and YOLOv12m, the proposed method demonstrates better detection accuracy and recall capability on the current dataset, indicating that it can more effectively identify small-scale lesions, weak-texture moldy spots, and irregular damage regions in complex orchard scenes. Ablation studies validate the effectiveness of each module, confirming that the performance improvement from combining multiple modules originates from their synergistic optimization across different stages.

It should be noted that the conclusions of this paper are primarily based on the current citrus defective fruit dataset and a desktop GPU experimental platform. Due to the still limited number of original samples and the fact that part of the data was collected in self-built scenarios, the model's generalization ability across regions, seasons, cultivars, and different acquisition devices requires further validation. Moreover, although the parameter count and FPS indicate a certain potential for real-time detection, systematic on-device tests for latency, power consumption, and long-term stability on actual edge devices have not yet been completed. Therefore, RT-DETR-DCEA currently serves as a feasible lightweight solution for real-time citrus defect detection in complex orchard environments, providing a model foundation for subsequent intelligent orchard inspection, pre-harvest defect screening, and automated grading systems. Future research will further integrate cross-domain data augmentation, model compression, and edge deployment optimization to promote the application of this method in practical agricultural production scenarios.

Author Contributions: Conceptualization, J.Q., Y.S. and L.W.; methodology, J.Q., B.Z., Y.C. and S.L.; software, J.Q. and H.L.; validation, J.Q.; formal analysis, J.Q.; investigation, J.Q.; resources, J.Q.; data curation, J.Q.; writing—original draft preparation, J.Q.; writing—review and editing, T.L.; visualization, J.Q.; supervision, T.L.; project administration, T.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Central Fund for Guiding Development of Local Science and Technology under Grant No. 202407AB110010, the Yunnan Science and Technology Major Project under Grant No. 202302AE090020, and the Yunnan Provincial Science and Technology Talents and Platforms Program under Grant No. 202305AF150126.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the data that support the findings of this study are available from the corresponding author upon reasonable request. The data consist of publicly available image resources and data collected by the research team. Publicly available data can be obtained according to the requirements of the original platform. However, the self-collected data are not publicly available due to restrictions on collection site permissions and data usage.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

To verify the effectiveness of the dynamic weighting mechanism in the DIMB module, we replaced the adaptive weights in DynamicInceptionDWConv2d with static average fusion (i.e., forcing $\alpha_s = \alpha_h = \alpha_v = 1/3$) under the same experimental settings, while keeping all other structures unchanged. The results are presented in the table below.

Table A1. Ablation comparison between dynamic weighting and static average fusion.

Integration Method	P (%)	R (%)	mAP@50 (%)	Params ($\times 10^7$)	FPS
Static average fusion $Y = (F_s + F_h + F_v)/3$	86.5	77.2	84.9	1.567	80
Dynamic Weight Fusion $Y = \alpha_s F_s + \alpha_h F_h + \alpha_v F_v$	87.8	78.9	85.6	1.568	79

References

- Wu, M.; Qiu, Y.; Wang, W.; Su, X.; Cao, Y.; Bai, Y. Improved RT-DETR and its application to fruit ripeness detection. *Front. Plant Sci.* **2025**, *16*, 1423682.
- Sun, J.; Shen, X.; Huang, H.; Wang, Q.; Zhao, H. RT-SLAM: A real-time visual SLAM system integrating enhanced RT-DETR and optical flow techniques. *IEEE Internet Things J.* **2024**, *12*, 12803–12814.
- Madan, M.; Reich, C. strengthening small object detection in adapted RT-DETR through robust enhancements. *Electronics* **2025**, *14*, 3830.
- Zhang, H.; Ma, Z.; Li, X. Rs-detr: An improved remote sensing object detection model based on rt-detr. *Appl. Sci.* **2024**, *14*, 10331.
- Kong, Y.; Shang, X.; Jia, S. Drone-DETR: Efficient small object detection for remote sensing image using enhanced RT-DETR model. *Sensors* **2024**, *24*, 5496.
- Sapkota, R.; Flores-Calero, M.; Qureshi, R.; Badgujar, C.; Nepal, U.; Poullose, A.; Zeno, P.; Vaddevolu, U.B.P.; Khan, S.; Shoman, M.; et al. YOLO advances to its genesis: A decadal and comprehensive review of the You Only Look Once (YOLO) series. *Artif. Intell. Rev.* **2025**, *58*, 274.
- Zhou, Y. A YOLO-NL object detector for real-time detection. *Expert Syst. Appl.* **2024**, *238*, 122256.
- Oreski, G. YOLO* C—Adding context improves YOLO performance. *Neurocomputing* **2023**, *555*, 126655.
- Chen, C.; Zheng, Z.; Xu, T.; Guo, S.; Feng, S.; Yao, W.; Lan, Y. Yolo-based uav technology: A review of the research and its applications. *Drones* **2023**, *7*, 190.
- Gündüz, M.Ş.; Işık, G. A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models. *J. Real-Time Image Process.* **2023**, *20*, 5.
- Vijayakumar, A.; Vairavasundaram, S. Yolo-based object detection models: A review and its applications. *Multimed. Tools Appl.* **2024**, *83*, 83535–83574.

12. Huangfu, Y.; Huang, Z.; Yang, X.; Zhang, Y.; Li, W.; Shi, J.; Yang, L. HHS-RT-DETR: A method for the detection of citrus green-ing disease. *Agronomy* **2024**, *14*, 2900.
13. Ye, R.; Shao, G.; Gao, Q.; Zhang, H.; Li, T. CR-YOLOv9: Improved YOLOv9 multi-stage strawberry fruit maturity detection application integrated with CRNET. *Foods* **2024**, *13*, 2571.
14. Geng, X.; Han, X.; Cao, X.; Su, Y.; & Shu, D. YOLOV9-CBM: An improved fire detection algorithm based on YOLOV9. *IEEE Access* **2025**, *13*, 19612–19623.
15. Yang, R.; Yuan, D.; Zhao, M.; Zhao, Z.; Zhang, L.; Fan, Y.; Liang, G.; Zhou, Y. Camellia oleifera Tree Detection and Counting Based on UAV RGB Image and YOLOv8. *Agriculture* **2024**, *14*, 1789.
16. Ragab, M.G.; Abdulkadir, S.J.; Muneer, A.; Alqushaibi, A.; Sumiea, E.H.; Qureshi, R.; Al-Selwi, S.M.; Alhussian, H. A compre-hensive systematic review of YOLO for medical object detection (2018 to 2023). *IEEE Access* **2024**, *12*, 57815–57836.
17. Zhao, C.; Shu, X.; Yan, X.; Zuo, X.; Zhu, F. RDD-YOLO: A modified YOLO for detection of steel surface defects. *Measurement* **2023**, *214*, 112776.
18. Yin, Y.; Li, H.; Fu, W. Faster-YOLO: An accurate and faster object detection method. *Digit. Signal Process.* **2020**, *102*, 102756.
19. Badgujar, C.M.; Poulouse, A.; Gan, H. Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Comput. Electron. Agric.* **2024**, *223*, 109090.
20. Huang, Z.; Wang, J.; Fu, X.; Yu, T.; Guo, Y.; Wang, R. DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Inf. Sci.* **2020**, *522*, 241–258.
21. Lee, J.; Hwang, K. YOLO with adaptive frame control for real-time object detection applications. *Multimed. Tools Appl.* **2022**, *81*, 36375–36396.
22. Prinzi, F.; Insalaco, M.; Orlando, A.; Gaglio, S.; Vitabile, S. A yolo-based model for breast cancer detection in mammograms. *Cogn. Comput.* **2024**, *16*, 107–120.
23. Zhou, J.; Zhang, B.; Yuan, X.; Lian, C.; Ji, L.; Zhang, Q.; Yue, J. YOLO-CIR: The network based on YOLO and ConvNeXt for infrared object detection. *Infrared Phys. Technol.* **2023**, *131*, 104703.
24. Zhou, C.; He, H.; Zhou, H.; Ge, F.; Yu, P. MSRT-DETR: A novel RT-DETR model with multi-scale feature sequence for cell detection. *Biomed. Signal Process. Control* **2025**, *103*, 107378.
25. Wang, S.; Jiang, H.; Yang, J.; Ma, X.; Chen, J.; Li, Z.; Tang, X. Lightweight tomato ripeness detection algorithm based on the improved RT-DETR. *Front. Plant Sci.* **2024**, *15*, 1415297.
26. Wu, D.; Wu, R.; Wang, H.; Cheng, Z.; To, S. Real-time detection of blade surface defects based on the improved RT-DETR. *J. Intell. Manuf.* **2026**, *37*, 313–325.
27. Zhang, M.; Wei, X.; Liu, G.; Chen, M.; Zhao, C.; Liu, Y.; Bao, Z.; Guo, Y.; An, R.; Zhao, P. Balancing complexity and accuracy for defect detection on filters with an improved RT-DETR. *Sci. Rep.* **2025**, *15*, 29720.
28. Liu, G.; Yang, D.; Ye, J.; Lu, H.; Wang, Z.; Zhao, Y. A real-time welding defect detection framework based on RT-DETR deep neural network. *Adv. Eng. Inform.* **2025**, *65*, 103318.
29. Jin, M.; Zhang, J. Research on microscale vehicle logo detection based on real-time detection transformer (RT-DETR). *Sensors* **2024**, *24*, 6987.
30. Luo, X.; Liu, R.; He, X.; Wang, H. A road damage detection model based on improved RT-DETR for complex environments. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 5037413.
31. Wu, Y.; Yuan, S.; Tang, L. Plant recognition of maize seedling stage in UAV remote sensing images based on H-RT-DETR. *Plant Methods* **2025**, *21*, 60.
32. Hao, Z.; Liu, B.; Xu, B. A lightweight rt-detr model for metal surface defect detection using multi-scale network and additive attention mechanism. *J. Nondestruct. Eval.* **2025**, *44*, 107.
33. Suvittawat, N.; Kurniawan, C.; Datephanyawat, J.; Tay, J.; Liu, Z.; Soh, D.W.; Ribeiro, N.A. Advances in aircraft skin defect detection using computer vision: A survey and comparison of YOLOv9 and RT-DETR performance. *Aerospace* **2025**, *12*, 356.
34. Suchý, I.; Turčaník, M. Review of large YOLOv8 and RT-DETR energy efficiency on edge devices for real-time detection. *Sci. Rep.* **2026**, *16*, 10908.
35. Liu, L.; Yuan, R.; Lv, Y.; Bo, W.; Hu, X.; Gedikli, E.D. An efficient high-resolution remote sensing object detection approach using multi-level-optimized rt-detr model. *IEEE Sens. J.* **2025**, *25*, 42306–42315.
36. Lv, Z.; Dong, S.; Xia, Z.; He, J.; Zhang, J. Enhanced real-time detection transformer (RT-DETR) for robotic inspection of under-water bridge pier cracks. *Autom. Constr.* **2025**, *170*, 105921.
37. Liu, C.; Zhang, Y.; Shen, J.; Liu, F. Improved RT-DETR for infrared ship detection based on multi-attention and feature fusion. *J. Mar. Sci. Eng.* **2024**, *12*, 2130.

38. Mao, H.; Gong, Y. Steel surface defect detection based on the lightweight improved RT-DETR algorithm. *J. Real-Time Image Process.* **2025**, *22*, 28.
39. Liu, B.; Jin, J.; Zhang, Y.; Sun, C. WRRT-DETR: Weather-robust RT-DETR for drone-view object detection in adverse weather. *Drones* **2025**, *9*, 369.
40. Yu, W.; Zhou, P.; Yan, S.; Wang, X. Inceptionnext: When inception meets convnext. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–24 June 2024; pp. 5672–5683.
41. Yu, W.; Si, C.; Zhou, P.; Luo, M.; Zhou, Y.; Feng, J.; Yan, S.; Wang, X. Metaformer baselines for vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 896–912.
42. Wang, L.; Ye, R.; Chen, Y.; Li, T. YOLOv10-LGDA: An Improved Algorithm for Defect Detection in Citrus Fruits Across Diverse Backgrounds. *Plants* **2025**, *14*, 1990.
43. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015.
44. Rahman, M.M.; Munir, M.; Marculescu, R. Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–24 June 2024; pp. 11769–11779.
45. Zhou, S.; Chen, D.; Pan, J.; Shi, J.; Yang, J. Adapt or perish: Adaptive sparse transformer with attentive feature refinement for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–24 June 2024; pp. 2952–2963.
46. Su, Z.; Shao, Y.; Li, P.; Zhang, X.; Zhang, H. Improved RT-DETR network for high-quality defect detection on digital printing fabric. *J. Nat. Fibers* **2025**, *22*, 2476634.
47. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. Yolo-facev2: A scale and occlusion aware face detector. *Pattern Recognit.* **2024**, *155*, 110714.
48. Hsu, W.Y.; Lin, W.Y. Ratio-and-scale-aware YOLO for pedestrian detection. *IEEE Trans. Image Process.* **2020**, *30*, 934–947.
49. Zhou, X.; Jiang, L.; Hu, C.; Lei, S.; Zhang, T.; Mou, X. YOLO-SASE: An improved YOLO algorithm for the small targets detection in complex backgrounds. *Sensors* **2022**, *22*, 4600.
50. Wu, T.; Dong, Y. YOLO-SE: Improved YOLOv8 for remote sensing object detection and recognition. *Appl. Sci.* **2023**, *13*, 12977.
51. Tian, Y.; Wang, S.; Li, E.; Yang, G.; Liang, Z.; Tan, M. MD-YOLO: Multi-scale Dense YOLO for small target pest detection. *Comput. Electron. Agric.* **2023**, *213*, 108233.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.