MDGL-DETR: An Efficient Method for Detecting Apple Leaf Diseases by Integrating Global and Local Features

Guoli Wang
Anhui Agricultural University
Yong Ye
yeyong@ahau.edu.cn
Anhui Agricultural University
Tao Luo
Anhui Agricultural University

Research Article

Keywords: Object Detection, RT-DETR, Apple Leaf Disease, Attention Mechanism, Feature Fusion

Posted Date: February 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8714923/v1

Additional Declarations: No competing interests reported.

MDGL-DETR: An Efficient Method for Detecting Apple Leaf Diseases by Integrating Global and Local Features

Guoli Wang,[a] Yong Ye,[a] Tao Luo[a]

[a]Anhui Agriculture University, School of Information and Artificial Intelligence, Computer Science and Technology, Hefei 230001, China

Abstract. To address challenges such as diverse apple leaf disease phenotypes, high background similarity, and complex natural environments, this study proposes an improved MDGL-DETR model based on RT-DETR, aiming to enhance both the accuracy and efficiency of apple leaf disease detection. First, a Multi-Scale Dilated Asymmetric structure combined with Channel Reduction Attention is designed to strengthen extraction capabilities for multi-scale and global features while reducing computational costs. Second, a Directional Shift Adaptive Context module is proposed, which utilizes shift convolution and SoftPool to dynamically focus on disease regions and suppress redundant background interference. Finally, a Global-Local Collaborative Fusion module is constructed to facilitate efficient interaction between local texture and global semantic information, thereby reinforcing feature representation capabilities. Experimental results indicate that the MDGL-DETR model achieves an mAP50 of 89.09%, an increase of 3.31% over the original RT-DETR, while reducing the computational load by 4.39%. Comprehensive evaluations show that the model outperforms other object detection models. The proposed MDGL-DETR provides a novel solution for the efficient detection of apple leaf diseases.

Keywords: Object Detection, RT-DETR, Apple Leaf Disease, Attention Mechanism, Feature Fusion.

***Second Author,** E-mail: yeyongahau.edu.cn

1 Introduction

China is the world's largest producer and consumer of apples [1]. As a key pillar of China's agriculture, the apple industry plays a pivotal role in fostering regional economic development and ensuring the sustained growth of farmers' income. Statistics indicate that in recent years, China's apple production has maintained a high volume and exhibited a steady growth trend; this serves as a powerful testament to the diligent labor of countless farmers and the vigorous development of the industry as a whole. However, leaf diseases represent a prevalent issue during the growth cycle,

frequently exerting severe adverse effects on apple yield and quality. This phenomenon not only incurs substantial economic losses for farmers but also has a direct bearing on national food safety and the quality of life for consumers. Consequently, the efficient and accurate identification of apple leaf diseases is of paramount importance [2].

Currently, the discovery and detection of crop leaf diseases in China rely heavily on manual inspection [3]. This approach is not only time-consuming and labor-intensive but also suffers from low identification accuracy, hindering timely intervention and loss mitigation. With the advent of deep learning theories, the utilization of object detection for the automatic identification and diagnosis of crop leaf diseases has gradually evolved into a mature approach. In 2014, the R-CNN algorithm [4] pioneered the integration of deep learning into object detection. Building upon the R-CNN framework, classic algorithms such as Faster R-CNN [5] and Mask R-CNN [6] were subsequently developed. Gong et al. optimized the Faster R-CNN model by introducing the advanced Res2Net and Feature Pyramid Network (FPN) architectures as the feature extraction backbone, combined with the Soft-NMS algorithm, thereby improving the identification accuracy of apple leaf diseases [7]. Hou et al. proposed a deep learning model named FPN-Inception Squeeze-and-Excitation ResNet-Faster R-CNN, which enhanced the recognition performance for apple leaf diseases [8].

The YOLO (You Only Look Once) algorithm represents a highly efficient approach in the field of object detection, with its core advantage lying in its capability for end-to-end real-time detection. Yan et al. proposed an adaptive feature enhancement module based on the YOLOv8s model, which strengthened the feature extraction capability for apple leaf diseases [9]. Xu et al. reduced model complexity while maintaining detection accuracy by incorporating MobileNet-V3s basic modules into the backbone network and integrating the Coordinate Attention (CA) mechanism. This approach enabled the real-time, accurate, and lightweight detection of apple leaf diseases [10]. Yang et al. introduced SPD-Conv to replace traditional strided convolution within the backbone of the YOLOv5 algorithm, providing a high-precision solution for maize leaf disease detection [11]. However, the two-stage architecture of the R-CNN series results in slow inference speeds, high computational resource consumption, and elevated deployment costs. Conversely, the YOLO series algorithms possess limited capability in capturing detailed information and exhibit insufficient global feature modeling abilities in complex backgrounds.

To address the technical bottlenecks inherent in the aforementioned algorithms, the Transformer architecture has achieved new breakthroughs in the field of object detection, leveraging its unique self-attention mechanism and global context modeling capability. The core advantage of the Transformer architecture lies in its ability to transcend the limitations of the local receptive fields characteristic of traditional CNNs. Through the self-attention mechanism, it enables the model to directly capture semantic associations between arbitrary positions within an image. This global modeling capability remedies the inadequacy of YOLO algorithms regarding feature modeling in complex backgrounds, while its end-to-end prediction paradigm circumvents the complicated multi-stage processing workflows associated with the R-CNN series.

Representative of this approach, the DETR (Detection Transformer) [12] algorithm pioneered the application of the Transformer architecture to object detection. By utilizing an encoder-decoder structure to directly predict object sets, it completely eliminated the need for anchor designs. Building upon the original DETR architecture, the Baidu PaddlePaddle team developed the RT-DETR (Real-Time DETR) [13] algorithm, which maintains high accuracy while significantly enhancing real-time processing capabilities. Subsequently, Li et al. introduced the P2 detection head and dynamic convolution, improving the efficiency of litchi leaf disease detection[14]. Liu et al. proposed the CSP-OmniKernel module and the VariableFocalLoss function to optimize RT-DETR, demonstrating excellent performance in the detection of rice pests and diseases [15]. Furthermore, Bao et al. proposed embedding a dynamic multi-feature fusion module into RT-DETR to enhance the model's adaptability to variations in the size of wheat scab lesions [16].

While the aforementioned improvements to RT-DETR have enhanced detection accuracy or reduced model complexity to a certain extent, there remains room for optimization. Moreover, these modifications were not specifically tailored for the scenario of apple leaf disease detection. In the context of detecting apple leaf diseases in natural environments, the diversity of diseases results in significant variations in phenotypic characteristics. Distinct diseases exhibit marked differences in their visual manifestations on leaves, including variations in color, shape, size, and edge texture. Even within the same disease category, morphological presentation varies across different growth stages. Furthermore, the complexity of the natural environment cannot be overlooked; factors such as illumination variations, the randomness of leaf orientation, and background noise interference further exacerbate the difficulty of feature extraction for the model. Addressing these challenges, this study utilizes RT-DETR as the baseline model and proposes

specific improvements, aiming to enhance both the accuracy and efficiency of apple leaf disease detection.

2 Materials and Methods

2.1 Experimental Dataset

The apple leaf disease dataset used in this study was constructed by combining the FGVC8 Plant Pathology 2021 dataset and the AppleLeaf9 public dataset [17]. Following manual screening to remove images with severe overexposure, defocus blur, or other significant quality issues, a total of 5,147 disease images were retained.

The dataset includes 1,202 images of Alternaria leaf spot (*Alternaria alternata f. sp. malicorticis*), 795 images of grey spot (*Phyllosticta solitaria*), 1,146 images of powdery mildew (*Podosphaera leucotricha*), 1,118 images of rust (*Gymnosporangium yamadae*), and 886 images of scab (*Venturia inaequalis*).

All images were resized to 640×640 pixels and annotated using the X-AnyLabeling software (version 2.5.4). An example of the annotation process is shown in Figure 1.

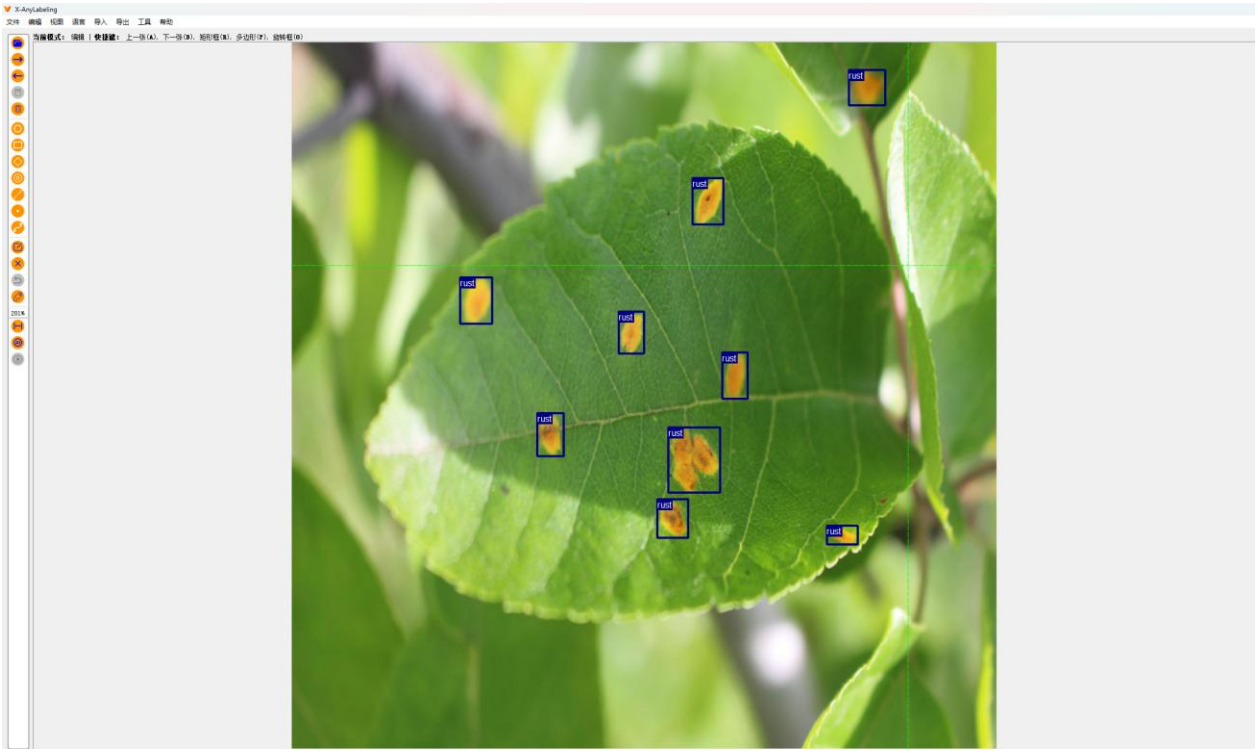

Fig. 1 Annotation of Apple Leaf Diseases Dataset.

To improve model generalization, data augmentation was performed on all images to expand the dataset. Simultaneously, to prevent data leakage, a strategy of "splitting before augmenting" was adopted. To ensure that the number of original images for each disease category in both the validation and test sets exceeded 100—thereby avoiding bias issues and high volatility in evaluation results—the images for each disease were partitioned into training, validation, and test sets according to a 7:1.5:1.5 ratio.

The resulting training set contained 3,601 images (Alternaria leaf spot: 841, Grey spot: 555, Powdery mildew: 802, Rust: 782, Scab: 621); the validation set comprised 772 images (Alternaria leaf spot: 180, Grey spot: 120, Powdery mildew: 172, Rust: 168, Scab: 132); and the test set consisted of 774 images (Alternaria leaf spot: 181, Grey spot: 120, Powdery mildew: 172, Rust: 168, Scab: 133). Subsequently, data augmentation was applied to all images within the training, validation, and test sets. For each image, brightness, saturation, and contrast were first adjusted (increased or decreased), followed by the random addition of Gaussian noise, salt-and-pepper noise, shadows, or occlusions. A comparison before and after augmentation is shown in Figure 2. To prevent overfitting and ensure data consistency, each image was subjected to data augmentation exactly once.

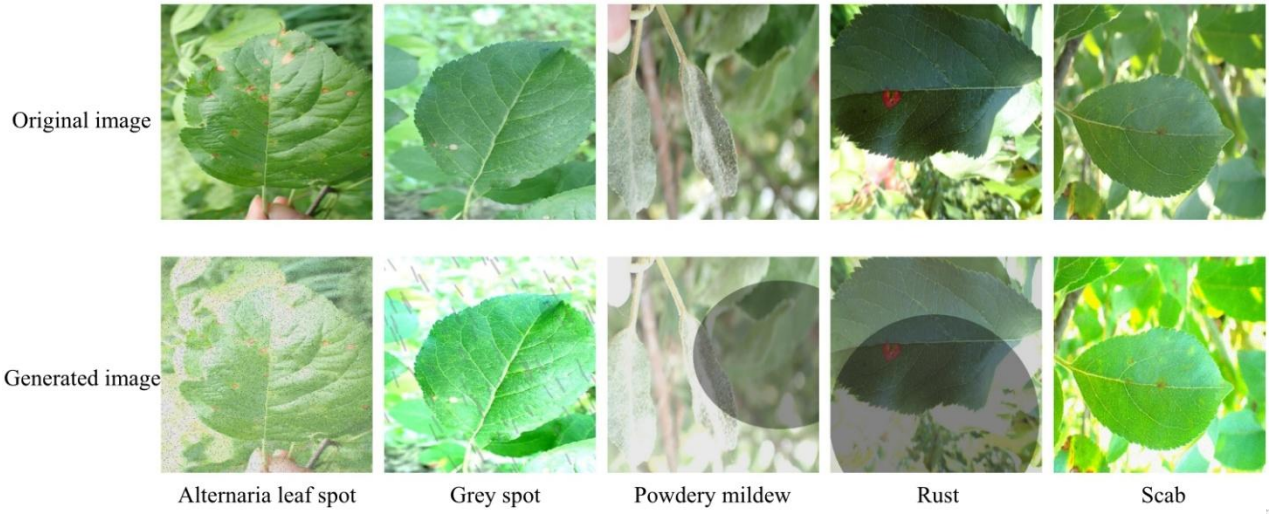

Fig. 2 Comparison of Dataset Images Before and After Data Augmentation.

The resulting final dataset comprises a total of 10,294 images. This includes 7,202 images in the training set (Alternaria leaf spot: 1,682; Grey spot: 1,110; Powdery mildew: 1,604; Rust: 1,564; Scab: 1,242), 1,544 images in the validation set (Alternaria leaf spot: 360; Grey spot: 240; Powdery mildew: 344; Rust: 336; Scab: 264), and 1,548 images in the test set (Alternaria leaf spot: 362; Grey spot: 240; Powdery mildew: 344; Rust: 336; Scab: 266).

Regarding annotations, the dataset contains a total of 50,254 bounding boxes. These are distributed as follows: 34,358 in the training set (Alternaria leaf spot: 7,920; Grey spot: 4,280; Powdery mildew: 2,732; Rust: 7,350; Scab: 12,076); 8,884 in the validation set (Alternaria leaf spot: 1,804; Grey spot: 1,180; Powdery mildew: 664; Rust: 1,660; Scab: 3,576); and 7,012 in the test set (Alternaria leaf spot: 1,706; Grey spot: 612; Powdery mildew: 604; Rust: 1,606; Scab: 2,484).

2.2 RT-DETR

The RT-DETR architecture comprises three main components: a Backbone, a Hybrid Encoder, and a Transformer Decoder with auxiliary prediction heads. The Backbone extracts visual features from input images and provides feature representations for subsequent interaction and fusion stages. The Hybrid Encoder consists of two modules: the Attention-based Intra-scale Feature Interaction (AIFI) module and the Cross-scale Feature Fusion Module (CCFM). The AIFI applies a lightweight self-attention mechanism to capture local details while reducing computational redundancy. The CCFM integrates multi-scale features (e.g., S3, S4, and S5) through a fusion block to enable multi-scale feature representation. The Decoder takes the encoded feature sequence as input and produces final detection results through iterative refinement of object queries. Unlike conventional object detection frameworks, the RT-DETR Decoder directly predicts non-overlapping bounding boxes without relying on Non-Maximum Suppression (NMS), thereby simplifying the detection pipeline and reducing inference latency.

2.3 MDGL-DETR

ResNet[18] serves as the default backbone network for RT-DETR. Considering the real-time requirements of apple leaf disease detection and the limited size of the dataset used in this study, employing deeper backbone networks (e.g., ResNet50 or ResNet101) would substantially increase computational cost and potentially increase the risk of overfitting. Therefore, ResNet18 is selected as the backbone network in this work.

Although RT-DETR achieves competitive real-time performance, its detection accuracy remains limited when applied to complex natural scenarios of apple leaf disease detection. To address these limitations, this study introduces several targeted architectural optimizations based on the RT-DETR framework. Specifically, (1) a Multi-Scale Dilated Asymmetric (MSDA) structure combined with a Channel Reduction Attention (CRA)[19] mechanism is integrated into the

BasicBlock of ResNet18; (2) a Directional Shift Adaptive Context (DSAC) module is employed to modify RepC3; and (3) a Global-Local Collaborative Fusion (GLCF) module is constructed to facilitate joint modeling of local and global information. The resulting model is referred to as MDGL-DETR, and its overall architecture is illustrated in Figure 3

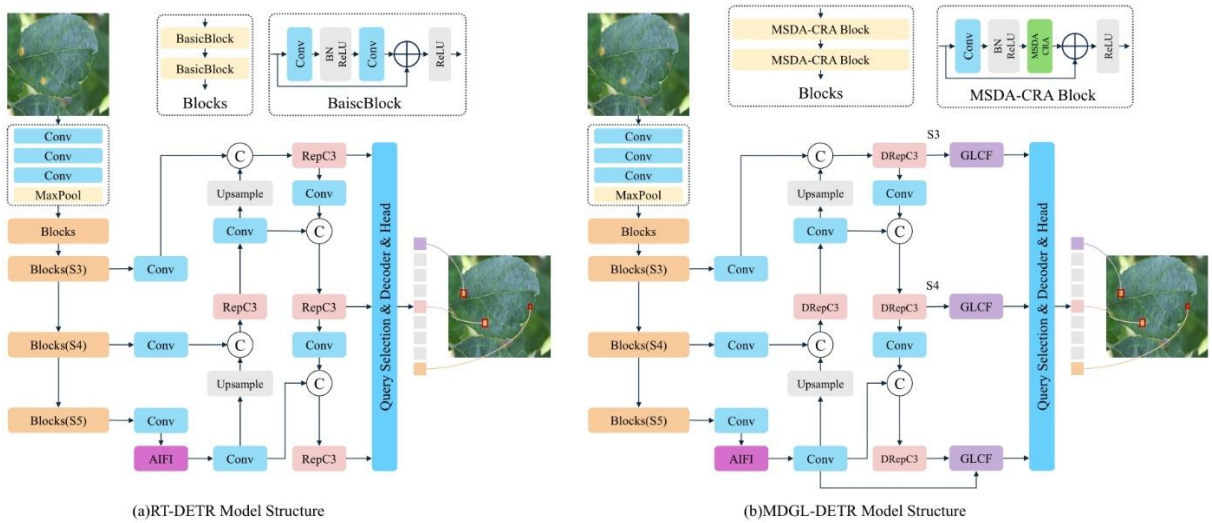

Fig. 3 RT-DETR Model Structure and MDGL-DETR Model Structure.

2.3.1 MSDA-CRA

To address the large variation in the spatial scales of apple leaf diseases—ranging from localized spots to whole-leaf infections—the representational capacity of ResNet18 is limited when modeling features across diverse scales. Conventional approaches often adopt multi-branch convolutional structures with different kernel sizes to enhance multi-scale feature representation; however, such designs typically introduce substantial parameter and computational overhead.

To achieve efficient multi-scale feature extraction with limited computational cost, this study proposes a Multi-Scale Dilated Asymmetric (MSDA) structure and integrates it with a Channel Reduction Attention (CRA) mechanism, forming the MSDA-CRA module. By embedding MSDA-CRA into the BasicBlock of ResNet18, an MSDA-CRA Block is constructed, as illustrated in Figure 4.

The MSDA module employs a lightweight multi-branch architecture for multi-scale feature extraction. Input features are first compressed along the channel dimension using a 1×1 convolution to reduce computational complexity. Four parallel branches are then applied. In each branch, a standard 3×3 convolution is decomposed into sequential 1×3 and 3×1 asymmetric convolutions to reduce computational cost while preserving the effective receptive field. Based on

this structure, dilated convolutions with different dilation rates (1, 2, 4, and 8) are introduced to expand the receptive field without increasing parameter count, enabling the extraction of features with varying contextual ranges. Depthwise separable convolutions are adopted to further reduce computational overhead.

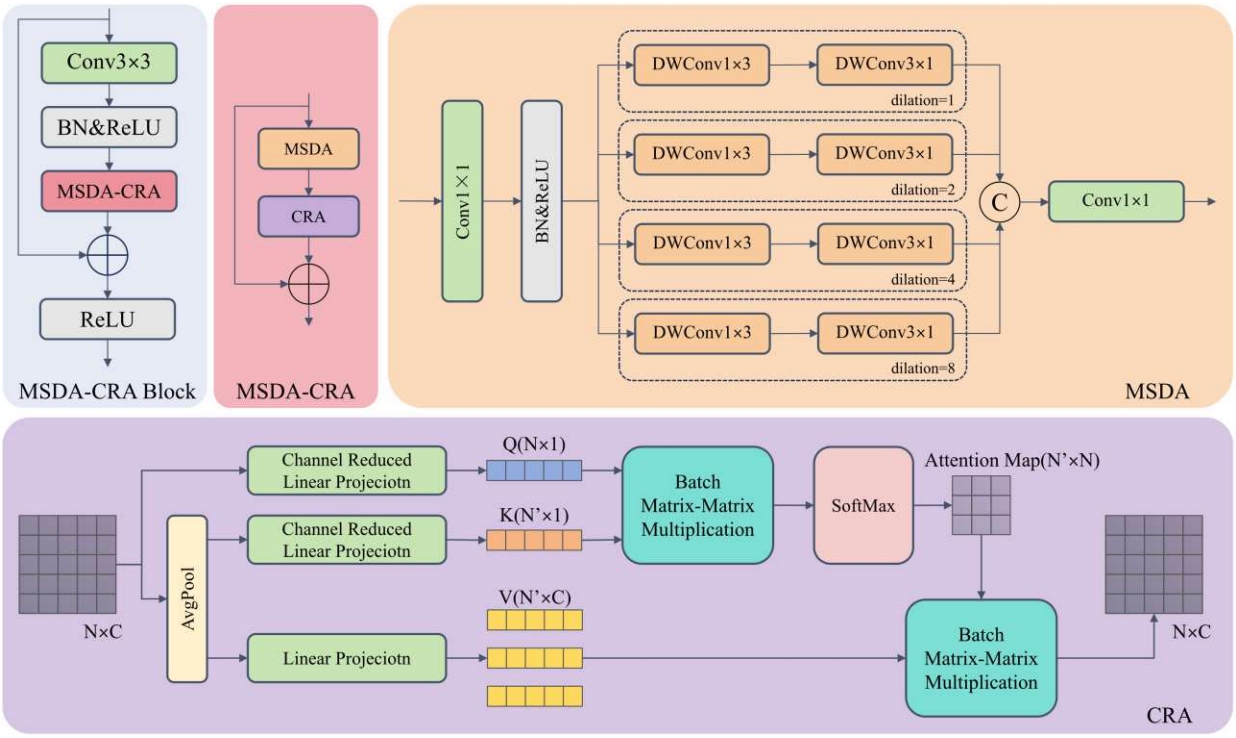

Fig. 4 MSDA-CRA Block Structure.

After multi-branch feature extraction, the outputs are concatenated and combined with a shortcut projection implemented via a 1×1 convolution. The CRA module is then applied to recalibrate channel-wise feature responses. CRA follows a Query-Key-Value formulation, where the Query is derived from spatial feature representations and the Key and Value are obtained from globally pooled channel features. To reduce computational cost, the channel dimensions of Query and Key are projected into a lower-dimensional space, allowing efficient modeling of channel-wise dependencies. Finally, a residual connection is employed to integrate the recalibrated features with the input features.

2.3.2 DSAC

The RepC3 component within RT-DETR applies uniform convolution operations across every position of the input feature map, thereby lacking the capability to discriminate the importance of different regions. In the context of apple leaf disease detection, disease lesions often occupy only

a minor fraction of the image. However, RepC3 treats features from disease regions and background regions indiscriminately, failing to dynamically adjust weights according to regional importance. This results in the features of small disease spots being overshadowed by background noise, while redundant background features, such as leaf texture, are erroneously amplified. To address this issue, this study proposes the Directional Shift Adaptive Context (DSAC) module to improve RepC3, creating the DRepC3 module.

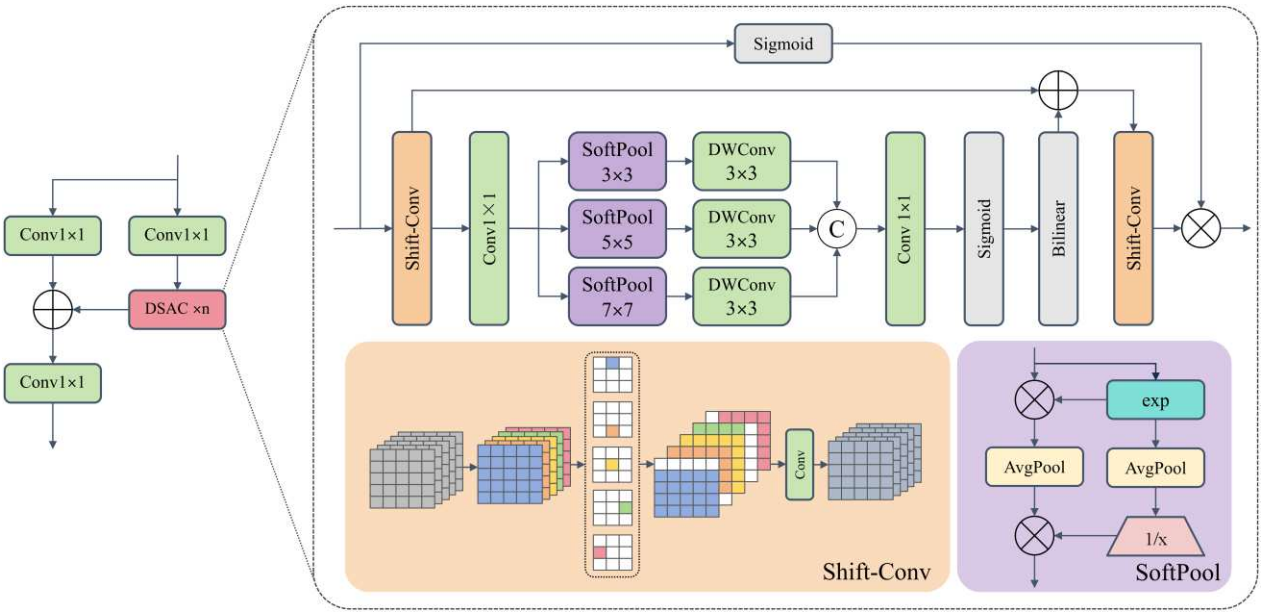

Fig. 5 DRepC3 Structure.

As illustrated in Figure 5 Shift Convolution [20] is introduced within the DSAC module. Compared to traditional convolution methods, Shift Convolution realizes the convolution process through strategic shift operations, enabling the simulation of spatial relationship modeling for features. Specifically, for an input feature map $X \in \mathbb{R}^{H\times W\times C}$ (where C denotes the channel count ， and H and W denote height and width, respectively), Shift Convolution divides the channels C evenly into 5 groups. Each group contains g=C/5 channels. A shift operation is performed separately on each of the 5 feature map groups $X_i \in \mathbb{R}^{H\times W\times g}(i = 1,2,3,4,5)$. The pixel offset for the shift operation is implemented via a predefined 3×3 sparse convolution kernel with an offset magnitude of 1 pixel. For each feature map group X_i , the convolution kernel is set to 1 only at the single position corresponding to the specific offset direction (left, right, up, down, or identity), while all other positions are set to 0. This sparse kernel renders the convolution operation equivalent to directly copying pixel values from neighboring positions in the feature map, thereby simulating a pixel translation effect.

$$
\begin{aligned}
X_1^{left} &= shift(X_1, direction = left) \\
X_2^{right} &= shift(X_2, direction = right) \\
X_3^{up} &= shift(X_3, direction = up) \\
X_4^{down} &= shift(X_4, direction = down) \\
X_5^{identity} &= X_5
\end{aligned} \tag{1}
$$

These shifted feature maps are concatenated along the channel dimension to obtain $X_{shifted} \in \mathbb{R}^{H\times W\times C}$. Finally, a 1×1 convolution is applied to $X_{shifted}$ for channel fusion, resulting in the output feature map $X_S \in \mathbb{R}^{H\times W\times C_{out}}$ (where Cout is the number of output channels). This shift operation is due to the relative positional relationships between different regions in the feature map, thereby better adapting to the feature discrepancies between disease regions and other regions. SoftPool [21] calculates the relative importance of each pixel within a local region via Softmax weighting. It computes the importance value of a pixel x within its surrounding area according to Equation (2), enabling the model to automatically learn the importance of every position in the feature map.

$$
I(X)|_x = \sum_{k\in U} \sum_{i\in U_k} \frac{e^{x_i}}{\sum_{j\in U_k} e^{x_j}} \cdot x_i \tag{2}
$$

Where U is the neighborhood centered at x, x_i is the value of the i-th pixel within region U, x_j represents the indices of all pixels in region U, and $I(X)|_x$ is the importance value of pixel x within neighborhood U.

After the input feature map undergoes preliminary processing via Shift Convolution, the channel count is reduced using a 1×1 convolution. Subsequently, global pooled features are acquired through the parallel application of SoftPool operations at different scales and depthwise separable convolutions. These are then fused to obtain a multi-scale importance map:

$$
\begin{aligned}
X_{S_1} &= S_{conv}(X) \\
I_{ms}(X_{S_1}) &= F\left(I_{3\times3}(X_{S_1}), I_{5\times5}(X_{S_1}), I_{7\times7}(X_{S_1})\right)
\end{aligned} \tag{3}
$$

Where X is the input feature map, S_{conv} represents the shift convolution operation, X_{S_1} is the feature map concatenated after shifting, F denotes the fusion operation, and $I_{ms}(X_{S_1})$ is the multi-scale importance map. Through the Sigmoid activation function and bilinear interpolation, the multi-scale importance map is normalized into a probability distribution and restored to the same spatial dimensions as X_{S_1}, thereby yielding the attention weight map:

$$
\mathcal{A}_{map} = \psi\left(\sigma\left(I_{ms}(X_{S_1})\right)\right) \tag{4}
$$

Where $\mathcal{A}_{map}$ is the attention weight map, $\sigma(\cdot)$ is the Sigmoid activation function, and $\psi(\cdot)$ denotes bilinear interpolation. The attention weight map reflects the importance of each position in the feature map; features in disease regions are enhanced due to higher weights, while features in other regions are suppressed due to lower weights. Simultaneously, a residual connection is employed to prevent the loss of critical disease features during the extraction process, facilitating better learning of subtle variations in disease characteristics. Finally, the output feature map is obtained after a second shift convolution. To avoid artifacts introduced by shift convolution and bilinear interpolation, the Sigmoid function is used to process the input feature map, generating a gating signal to further isolate disease-related features and reduce interference from irrelevant information:

$$Y = S_{conv}\left(\gamma\left(\mathcal{A}_{map} + X_{S_1}\right)\right) \odot \sigma(X) \tag{5}$$

Where Y is the output feature map, and γ is the ReLU activation function.

2.3.3 GLCF

The detection of apple leaf diseases requires the simultaneous consideration of local texture features (e.g., the shape and color of disease spots) and global semantic information (e.g., disease distribution and the proportion of infection across the whole leaf). However, the existing RT-DETR model exhibits deficiencies in collaboratively modeling local texture and global semantic information. Furthermore, feature maps processed by the MSDA-CRA Block and DRepC3 module lack an explicit interaction mechanism, which may result in feature redundancy or the overshadowing of critical information. To address these issues, this study proposes the Global-Local Collaborative Fusion (GLCF) module, designed to achieve the efficient fusion of local information and global semantics, thereby fully leveraging information across different feature hierarchies.

As illustrated in Figure 6, the GLCF module first utilizes a feature transformation layer, employing a 1×1 convolution to normalize the input feature maps along the channel dimension; this ensures that features from diverse sources are mapped into a unified feature space. Specifically, input feature X is derived from the MSDA-CRA Block, representing global features, while input feature Y originates from the DRepC3 module, representing local features. Following the 1×1 convolution processing, input features X and Y are transformed into X_i and Y_i, respectively:

$$X_i = Conv_{1\times1}(X)$$
$$Y_i = Conv_{1\times1}(Y) \quad (6)$$

Where $X \in \mathbb{R}^{H\times W\times C_1}$, $Y \in \mathbb{R}^{H\times W\times C_2}$, $X_i \in \mathbb{R}^{H\times W\times C}$, and $Y_i \in \mathbb{R}^{H\times W\times C}$ (with C_1, C_2, and C denoting channel counts, and H and W representing height and width, respectively). $Conv_{1\times1}$ denotes the 1×1 convolution operation. The Local Spatial Attention (LSA) mechanism [22] is applied to X_i to obtain the spatially-enhanced feature X_{LSA}. Subsequently, X_{LSA} is added to Y_i to yield the new local feature F_L:

$$X_{LSA} = f_{LSA}(X_i)$$
$$F_L = X_{LSA} + Y_i \quad (7)$$

Where $F_L \in \mathbb{R}^{H\times W\times C}$, and f_{LSA} represents the application of the LSA mechanism. This operation leverages the spatial information of global features to guide local features in focusing on critical regions, thereby enhancing spatial consistency. Subsequently, global average pooling and global max pooling are employed on X_i to aggregate features, capturing inter-channel dependencies and integrating global information. Softmax normalization is then utilized to generate channel weights, resulting in features X_a and X_m:

$$X_a = Softmax(Avg(X_i))$$
$$X_m = Softmax(Max(X_i)) \quad (8)$$

Where $X_a \in \mathbb{R}^{1\times1\times C}$, $X_m \in \mathbb{R}^{1\times1\times C}$; Avg denotes global average pooling, Max denotes global max pooling, and Softmax denotes Softmax normalization. The sum of X_a and X_m is applied to F_L, using the semantic importance of global features to dynamically adjust the channel weights of local features. This highlights disease-related features and suppresses irrelevant noise, yielding F_1:

$$F_1 = F_L \odot (X_a + X_m) \quad (9)$$

Where $F_1 \in \mathbb{R}^{H\times W\times C}$. Additionally, the High Frequency Enhancement (HFE) mechanism is applied to F_L to reinforce high-frequency information. This result is added to X_i to obtain F_2, mitigating the detail blurring inherent in global features and improving the discriminability between background and disease regions:

$$F_2 = f_{HFE}(F_L) + X_i \quad (10)$$

Where f_{HFE} represents the application of the HFE mechanism. Finally, F_1 and F_2 are concatenated and processed through a 3×3 convolution to obtain the output feature F:

$$F = Conv_{3\times3}(Concat(F_1, F_2)) \quad (11)$$

Where $F \in \mathbb{R}^{H\times W\times C}$, *Concat* denotes the concatenation operation, and $Conv_{3\times 3}$ denotes the 3×3 convolution. By integrating LSA, HFE, and pooling operations, the GLCF module achieves a synergistic fusion where global features provide spatial context guidance while local features contribute disease details, enabling the model to simultaneously attend to local nuances and global semantic information.

The LSA and HFE mechanisms act as critical components of the GLCF module. The LSA mechanism extracts local spatial features via depthwise separable convolutions, reorganizes features through cascaded convolution operations, maintains information integrity using residual connections, and finally enhances key local regions via a spatial attention mask. Specifically, it sequentially employs 1×1 convolutions and 3×3 depthwise separable convolutions three times: the first iteration performs preliminary extraction of local spatial features; the second captures more complex local features; and the third generates an adaptive local attention mask to focus on critical regions. Subsequently, a 1×1 convolution maps the features back to the original channel count, and the input features are weighted by the mask to efficiently aggregate local spatial information.

The primary objective of the HFE mechanism is to reinforce the high-frequency information of the feature maps. It first employs global average pooling to perform a smoothing operation, yielding the low-frequency information. Next, the high-frequency information is obtained by subtracting the smoothed feature map from the original input feature map. Finally, convolution operations and the Sigmoid activation function are utilized to further amplify this high-frequency information.

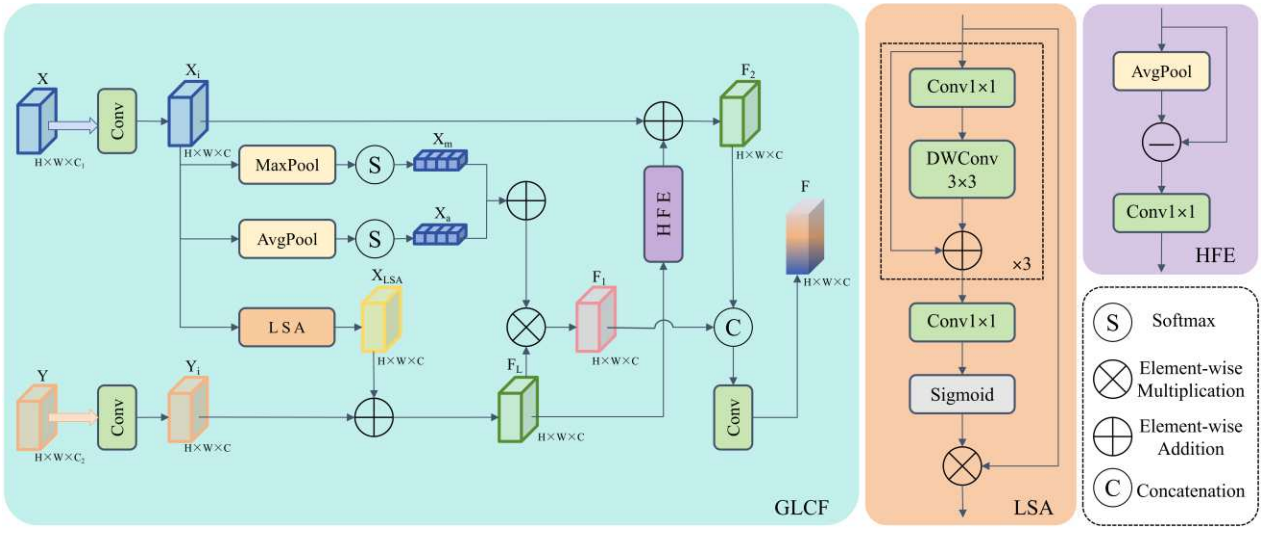

Fig. 6 GLCF Structure.

3 Results and Analysis

3.1 Experimental Platform, Parameters, and Evaluation Metrics

The experiments in this study were conducted on a cloud server platform. The hardware environment consisted of a 16-core CPU (Xeon(R) Platinum 8352V), 120 GB of RAM, and an NVIDIA RTX 4090 GPU with 24 GB of video memory. The software environment included the Ubuntu 22.04 operating system, Python 3.8.16, CUDA 11.7, and PyTorch 1.13.1.

The training parameters were configured as follows: the input resolution was set to 640×640 pixels, the number of epochs was 300, the batch size was 4, and the learning rate was set to 1e-4. The AdamW optimizer was employed for training. To prevent overfitting and reduce training time, an early stopping mechanism was implemented to automatically terminate the training process if the model evaluation metrics showed no improvement over 50 consecutive epochs.

To assess model performance, standard evaluation metrics for object detection were adopted, including Precision (P), Recall (R), mean Average Precision (mAP), the number of Parameters (Params), and Floating Point Operations (FLOPs).

3.2 Ablation Studies

This study evaluates the optimization effects of the proposed improvement strategies on the RT-DETR model through systematic ablation experiments. The experimental design adopts a multi-stage verification scheme: initially, the original, unmodified RT-DETR model serves as the baseline control group. Subsequently, the individual contribution of each of the three improvement strategies is verified sequentially. Following this, the strategies are integrated in a stepwise manner until all three are incorporated into the RT-DETR model. The cumulative effects of different strategy combinations on model performance are then analyzed based on various evaluation metrics. Through this controlled experimental approach, the study systematically reveals both the independent efficacy and the synergistic mechanisms of each improvement strategy.

As presented in Table 1, A denotes the replacement of the BasicBlock with the MSDA-CRA Block; B represents the improvement of RepC3 using the DSAC module; and C indicates the introduction of the GLCF module into the model.

Experimental results demonstrate that, compared to the baseline model, the independent utilization of the MSDA-CRA module resulted in a 1.09% increase in mAP50, while the number of

parameters and FLOPs decreased by 14.60% and 12.98%, respectively. This improvement is attributable to the MSDA-CRA module's ability to enhance multi-scale feature extraction while simultaneously maintaining computational efficiency. When both the MSDA-CRA and DSAC modules were employed, mAP50 increased by 1.85%, demonstrating that the DSAC module effectively improves the model's detection accuracy.

With the simultaneous integration of all three improvement strategies, the global features extracted by the MSDA-CRA module provided richer disease representations for the DSAC module, while the GLCF module reinforced semantic associations between features by coordinating global and local information. Consequently, the model's Precision, Recall, mAP50, and mAP50:95 increased by 3.75%, 2.99%, 3.31%, and 2.76%, respectively. Furthermore, the parameter count and FLOPs were reduced by 4.74% and 4.39%, respectively, indicating that the optimized model comprehensively outperforms the baseline across all metrics.

Tab. 1 Results of Ablation Experiments

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	Params(M)	FLOPs(G)
RT-DETR	85.10	79.42	85.78	57.86	19.878180	57.0
+A	86.69	80.69	86.87	58.85	16.976676	49.6
+B	86.49	81.32	86.88	59.17	20.928804	57.0
+C	86.67	80.30	87.42	59.15	20.787530	61.8
+A+B	87.60	81.81	87.63	59.48	18.027300	49.7
+A+B+C	88.85	82.41	89.09	60.62	18.936650	54.5

3.3 Heatmap Visualization

Heatmaps visualize the specific regions a model focuses on during image processing, with redder areas indicating higher attention levels. To facilitate a clearer comparison of the accuracy between the original RT-DETR model and the MDGL-DETR model in detecting apple leaf diseases, this study utilized Grad-CAM++ [23] to generate heatmaps for both models.

As illustrated in Figure 7, the original RT-DETR model exhibits insufficient precision in attending to disease regions. It suffers from issues such as relatively low heatmap activation in certain disease areas and the diffusion of heat density into non-diseased leaf regions, which obscures the precise boundaries of the lesions. In contrast, the improved model demonstrates significantly more focused attention on critical disease features.

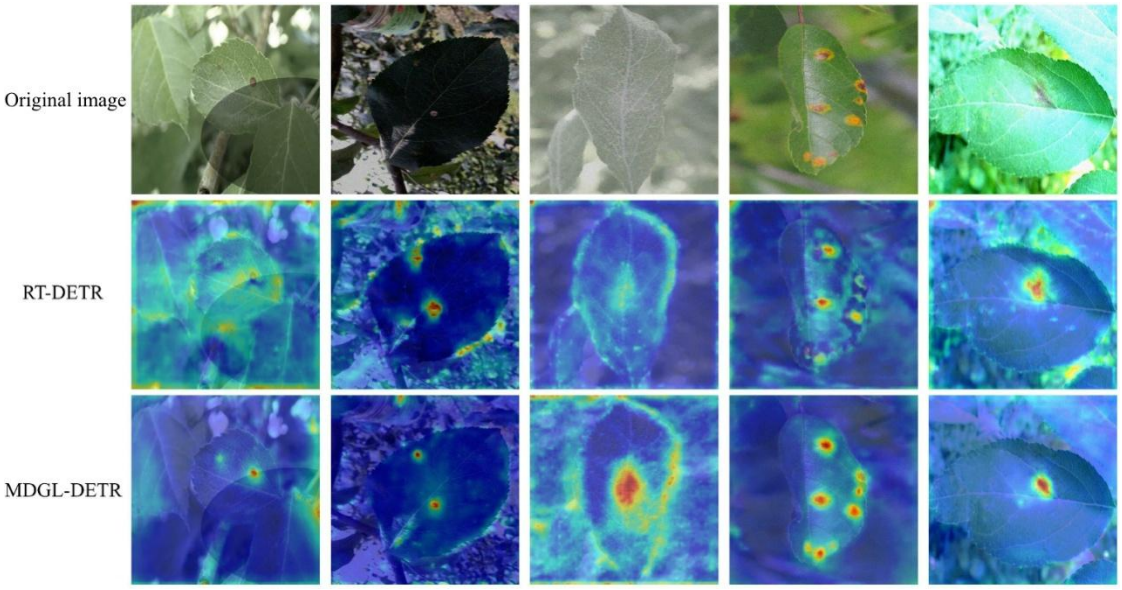

Fig. 7 Results of The Heatmap Comparison.

3.4 Comparative Experiment of Backbone Networks

To validate the efficacy of the MSDA-CRA module, this study conducted training by replacing the ResNet18 backbone of the RT-DETR model with EfficientViT [24], FasterNet [25], LSKNet [26], RepViT [27], and SwinTransformer [28], respectively.

As presented in Table 2, the model in which the BasicBlock was replaced by the MSDA-CRA Block achieved the optimal comprehensive performance. Specifically, while models utilizing EfficientViT, FasterNet, LSKNet, and RepViT as backbones exhibited lower parameter counts and FLOPs, their accuracy metrics declined across the board compared to the default ResNet18 backbone. Conversely, the model employing SwinTransformer as the backbone resulted in a substantial increase in parameters and FLOPs yet yielded no significant improvement in accuracy metrics. In contrast, the model integrating the MSDA-CRA Block not only reduced parameters and FLOPs compared to ResNet18 but also demonstrated significant improvements across all accuracy metrics.

Tab. 2 Results of The Backbone Comparison Experiment

Backbone	P(%)	R(%)	mAP50(%)	mAP50:95(%)	Params(M)	FLOPs(G)
ResNet18	85.10	79.42	85.78	57.86	19.878180	57.0
EfficientViT	82.25	73.76	80.65	53.17	10.707748	27.2
FasterNet	85.63	78.62	85.37	56.77	10.813224	28.5
LSKNet	84.91	79.58	85.12	56.93	12.566834	37.6
RepViT	84.35	77.40	83.09	55.18	13.307852	36.3
SwinTransformer	85.02	79.21	86.62	58.19	36.318526	97.0
Ours	86.69	80.69	86.87	58.85	16.976676	49.6

3.5 Comparative Experiment of Object Detection Models

To demonstrate the advanced performance of the proposed algorithm, it was evaluated against several state-of-the-art object detection models, including RetinaNet [29], EfficientDet [30], YOLOv5m, YOLOv8m, YOLOv10m [31], YOLO11m, and YOLOv12m [32].

As presented in Table 3, although MDGL-DETR did not achieve the lowest parameter count and FLOPs—specifically, its parameter count exceeds that of RetinaNet and YOLOv10m, and its FLOPs surpass those of RetinaNet, EfficientDet, and YOLOv5m—it demonstrates a commanding advantage in Precision, Recall, mAP50, and mAP50:95. Furthermore, when compared with YOLOv8m, YOLO11m, and YOLOv12m, MDGL-DETR not only requires lower computational costs but also achieves superior accuracy metrics.

Tab. 3 Results of The Object Detection Model Comparison Experiment

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	Params(M)	FLOPs(G)
RT-DETR	85.10	79.42	85.78	57.86	19.878180	57.0
RetinaNet	64.83	65.63	65.33	39.20	12.868190	39.0
EfficientDet	57.64	51.90	52.13	29.20	20.551550	41.1
YOLOv5m	84.47	81.43	86.08	57.86	20.869098	47.9
YOLOv8m	87.27	78.26	86.93	59.16	25.842655	78.7
YOLOv10m	86.96	79.02	86.97	59.73	15.316063	58.9
YOLO11m	86.14	80.19	87.84	59.81	20.033887	67.7
YOLOv12m	85.55	81.26	87.68	59.26	20.108767	67.1
MDGL-DETR	88.85	82.41	89.09	60.62	18.936650	54.5

To provide a more intuitive visualization of the performance differences between the improved RT-DETR model and other comparative models in apple leaf disease detection, a subset of detection results from the test set was selected for analysis. The confidence threshold was set to 0.40 to mitigate false detections. The detection results of the various models are illustrated in Figure 8.

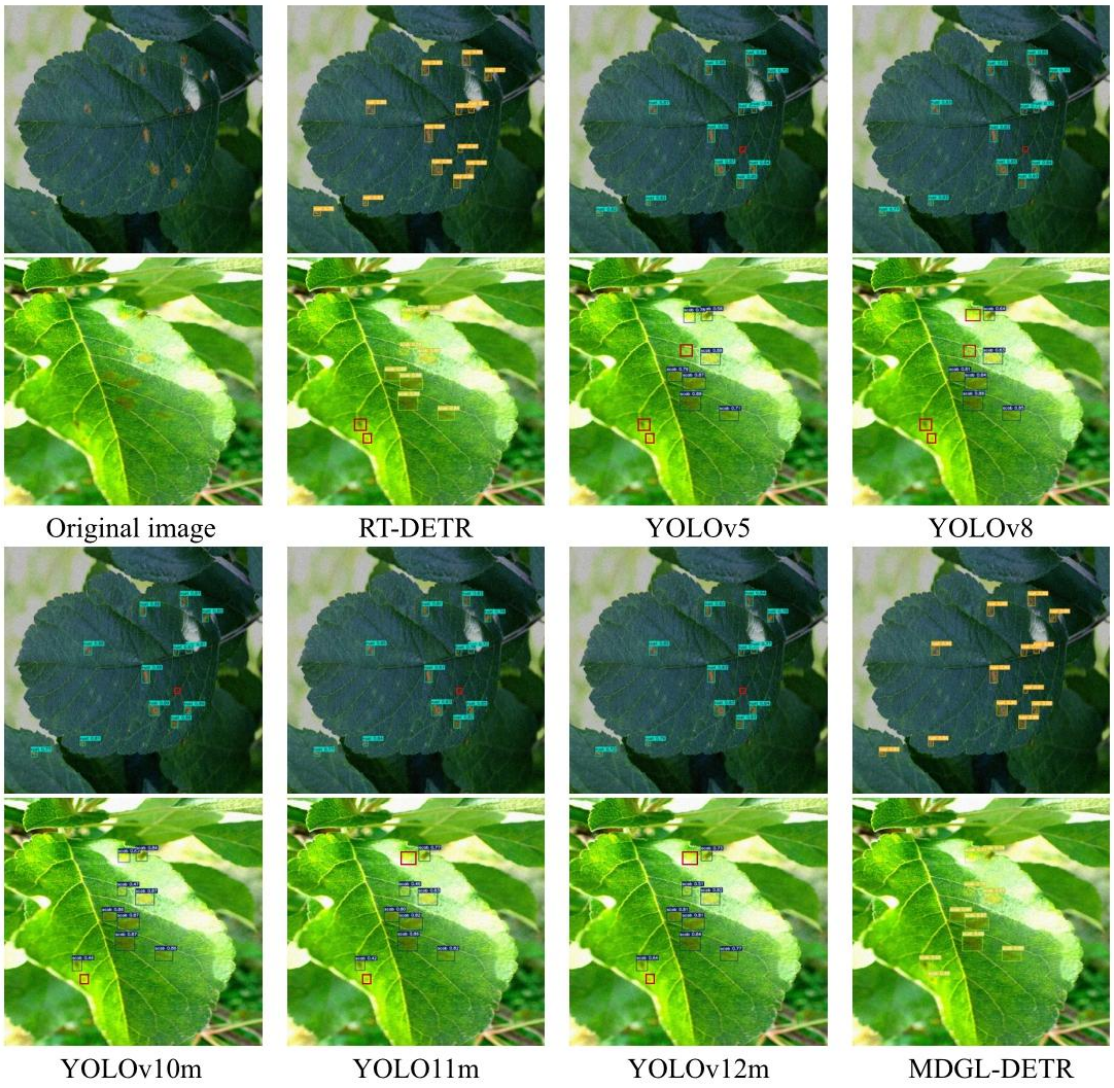

Fig. 8 Comparison of Detection Performance across Different Models.

4 Conclusion

To address the challenges associated with apple leaf disease detection in natural environments—specifically, the diversity of disease phenotypes, the difficulty of multi-scale feature extraction, and strong background interference—this study proposes three improvement strategies based on the RT-DETR model. First, the MSDA-CRA module was designed to enhance multi-scale feature representation through a multi-scale dilated asymmetric structure combined with channel reduction attention. Second, the DSAC module was introduced, utilizing shift convolution and SoftPool to dynamically focus on disease region features. Third, the GLCF module was constructed to achieve the cross-path fusion of local texture and global semantic information.

The results indicate that the synergistic interaction of these three improvements enables the model to achieve an mAP50 of 89.09% (an increase of 3.31% over the original RT-DETR), while reducing the parameter count to approximately 18.94 M (a decrease of 4.74% compared to the original RT-DETR). Compared with mainstream object detection models such as RetinaNet and YOLOv8m, MDGL-DETR exhibits significant advantages across all accuracy metrics and achieves optimal comprehensive performance, providing a novel solution for the efficient detection of apple leaf diseases. This study offers valuable references for the real-world deployment of disease identification technologies in smart agriculture. Future work will further explore lightweight deployment and the generalization capabilities of the model for detecting diseases in other crops.

References

1. Li, D., Zeng, X., & Liu, Y. (2022). Apple leaf disease identification model by coupling global and patch features. *Trans. Chin. Soc. Agric. Eng*, *38*, 207-214.
2. Khan, A. I., Quadri, S. M. K., Banday, S., & Shah, J. L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. computers and Electronics in Agriculture, 198, 107093.
3. Cheng, H., & Li, H. (2023). Identification of apple leaf disease via novel attention mechanism based convolutional neural network. Frontiers in Plant Science, 14, 1274231.
4. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
5. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
6. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
7. Gong, X., & Zhang, S. (2023). A high-precision detection method of apple leaf diseases using improved faster R-CNN. *Agriculture*, *13*(2), 240.
8. Hou, J., Yang, C., He, Y., & Hou, B. (2023). Detecting diseases in apple tree leaves using FPN–ISResNet–Faster RCNN. *European Journal of Remote Sensing*, *56*(1), 2186955.
9. Yan, C., & Yang, K. (2024). FSM-YOLO: Apple leaf disease detection network based on adaptive feature capture and spatial context awareness. *Digital Signal Processing,* 155, 104770.

10. Xu, W., & Wang, R. (2023). ALAD-YOLO: An lightweight and accurate detector for apple leaf diseases. *Frontiers in plant science*, *14*, 1204569.
11. Yang, H., Sheng, S., Jiang, F., Zhang, T., Wang, S., Xiao, J., ... & Wang, Q. (2024). YOLO-SDW: A method for detecting infection in corn leaves. *Energy Reports*, 12, 6102-6111.
12. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213-229). Cham: Springer International Publishing.
13. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024). Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 16965-16974).
14. Li, Z., Shen, Y., Tang, J., Zhao, J., Chen, Q., Zou, H., & Kuang, Y. (2025). IMLL-DETR: An intelligent model for detecting multi-scale litchi leaf diseases and pests in complex agricultural environments. *Expert Systems with Applications*, *273*, 126816.
15. Liu, J., Zhou, C., Zhu, Y., Yang, B., Liu, G., & Xiong, Y. (2025). RicePest-DETR: A transformer-based model for accurately identifying small rice pest by end-to-end detection mechanism. Computers and Electronics in Agriculture, 235, 110373.
16. Bao, W., Yang, Z., Zhang, P., Hu, F., Hu, G., Huang, L., & Yang, X. (2025). A domain adaptive wheat scab detection method for UAV images. *Computers and Electronics in Agriculture*, *233*, 110081.
17. Yang, Q., Duan, S., & Wang, L. (2022). Efficient identification of apple leaf diseases in the wild using convolutional neural networks. *Agronomy*, *12*(11), 2784.
18. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
19. Kang, B., Moon, S., Cho, Y., Yu, H., & Kang, S. J. (2024). Metaseg: Metaformer-based global contexts-aware network for efficient semantic segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 434-443).

20. Zhang, X., Zeng, H., Guo, S., & Zhang, L. (2022, October). Efficient long-range attention network for image super-resolution. In *European conference on computer vision* (pp. 649-667). Cham: Springer Nature Switzerland.
21. Stergiou, A., Poppe, R., & Kalliatakis, G. (2021). Refining activation downsampling with SoftPool. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10357-10366).
22. Tang, F., Xu, Z., Huang, Q., Wang, J., Hou, X., Su, J., & Liu, J. (2023, October). DuAT: Dual-aggregation transformer network for medical image segmentation. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)* (pp. 343-356). Singapore: Springer Nature Singapore.
23. Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018, March). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 839-847). IEEE.
24. Liu, X., Peng, H., Zheng, N., Yang, Y., Hu, H., & Yuan, Y. (2023). Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14420-14430).
25. Chen, J., Kao, S. H., He, H., Zhuo, W., Wen, S., Lee, C. H., & Chan, S. H. G. (2023). Run, don't walk: chasing higher FLOPS for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12021-12031).
26. Li, Y., Hou, Q., Zheng, Z., Cheng, M. M., Yang, J., & Li, X. (2023). Large selective kernel network for remote sensing object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16794-16805).
27. Wang, A., Chen, H., Lin, Z., Han, J., & Ding, G. (2024). Repvit: Revisiting mobile cnn from vit perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15909-15920).
28. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012-10022).
29. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

30. Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781-10790).

31. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., & Han, J. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, *37*, 107984-108011.

32. Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.