An Optimized Mature State Detection Model for Apple Flowers Under Complex Orchard

Fengping Zhang, Yujing Yang, Li Xu, Weikuan Jia

***Abstract*—Accurate assessment of apple flower maturity plays a pivotal role in ensuring successful pollination and subsequent fruit development. However, the inherently complex and unstructured orchard environment presents substantial challenges to reliable and efficient evaluation, limiting the application of intelligent agricultural technologies. To address this issue, we propose GLD-Net, a customized detection framework built upon RT-DETR, specifically adapted for fine-grained recognition of flower maturity stages. In the feature fusion network, we incorporate a Global-to-Local Spatial Aggregation (GLSA) module to replace conventional lateral convolutions and input projections. This module enhances spatial representation by combining global contextual cues—reflecting overall floral morphology—with fine-grained local focus, which sharpens the delineation of petals and stamens. Additionally, to improve the model's sensitivity to diverse floral structures, we strengthened the standard RepC3 component with a Diversified Branch Block Convolutional (DBBC3) module, utilizing multi-branch convolutions for comprehensive multi-scale feature extraction and deep-level information integration. We further introduce the AppleFlowers dataset, which includes flower images taken under a range of natural lighting and scene variations, enabling robust benchmarking in real orchard environments. Experimental results show that GLD-Net achieves a precision of 88.2%, a recall of 78.2%, and a mAP$_{50}$ of 86.0%, confirming its effectiveness and applicability in precision horticultural systems.**

***Index Terms*—Maturity state, Apple flowers, GLD-Net, Object detection**

I. INTRODUCTION

THE apple flowers mark the inception of the apple production cycle, with anthesis heralding the onset of fruit development; consequently, precision and efficiency in flower maturation assessment are pivotal for securing successful pollination and maximizing final yield [1]. In the pre-anthesis phase, meticulous selection of healthy, as pollen viability and vigor directly dictate fertilization outcomes. Upon reaching maturity-defined by approximately 20% petal opening, bees can be released to ensure effective bee-mediated pollination and enhance fruit-set rates [2]. Precision assessment of apple flower maturity is currently undermined by variability in illumination, weather conditions, and floral posture, resulting in both low accuracy and limited throughput [3]. Moreover, studies explicitly targeting the detection of flower maturity remain scarce, with existing efforts largely confined to flower identification and pollination-status evaluation [4]. Consequently, the development of robust, high-precision methodologies capable of accurately discerning flower maturity under complex and dynamic orchard environments represents an urgent and unresolved challenge. Researchers have made notable progress in enhancing the accuracy of flower recognition and pollination-state assessment. For instance, Zhang et al. [5] introduced a flower classification framework that incorporates spatial and channel attention mechanisms into the Xception network, along with a multi-loss function combining Triplet Loss and Softmax Loss. This design effectively minimizes intra-class variance while maximizing inter-class separability in the feature space, thereby improving classification performance. To address the challenge of flower detection in cluttered backgrounds, Lodh et al. [6] proposed an automated segmentation strategy based on color mean and variance. They extracted features using color moments and GIST descriptors, followed by classification with a support vector machine (SVM), achieving significant improvements in recognition accuracy over traditional methods. Deng et al. [7] developed a Mask R-CNN–based detection network featuring a ResNeXt-50-FPN backbone to enhance multi-scale representation. A class-agnostic mask branch was introduced to simplify the training process, and instance segmentation was employed to accurately detect citrus flowers, overcoming difficulties associated with their small size and dense distribution. Xiao et al. [8] improved the YOLOv5 architecture by deepening feature pyramids and refining path aggregation. Anchor boxes were optimized via K-means++ clustering, and the CBAM attention module was incorporated to better capture features of tiny floral targets like Phalaenopsis buds. Zhang et al. [9], on the other hand, proposed a streamlined detection model by replacing the YOLOv8n backbone with VanillaNet, reducing model complexity while maintaining detection efficacy. The integration of the LSKA module into the neck component of the architecture strengthens the model's ability to distill critical spatial features. Beyond empirical gains in detection accuracy, the proposed framework offers theoretical value for advancing future research in flower maturity analysis.

Manuscript received April 17, 2025; revised July 29, 2025.

This work is supported by National Nature Science Foundation of China (No.:62072289); Young Innovation Team Program of Shandong Provincial University (No.: 2022KJ250); New Twentieth Items of Universities in Jinan (2021GXRC049).

F. Zhang is a postgraduate student of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (e-mail: zfp231823@163.com).

Y. Yang is a postgraduate student of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (e-mail: 2522911363@qq.com).

L. Xu is an associate professor of School of Information Science and Engineering, Zaozhuang University, Zaozhuang 277160, China (Corresponding author, e-mail: xl412@126.com).

W. Jia is an associate professor of School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China (Corresponding author to provide phone: +86-531-86181755; fax: +86-531-86181750; e-mail: jwk_1982@163.com).

Recent progress in object detection technologies—spanning convolutional neural networks (CNNs), model pruning strategies, and clustering algorithms—has markedly enhanced detection accuracy and efficiency across diverse application domains. These approaches have demonstrated considerable success in fruit recognition [10, 11], medical image analysis [12], and intelligent transportation systems [13], and have also inspired advances in apple flower detection. For instance, Dias et al. [14] developed a flower recognition system robust to clutter and illumination variation, employing a pre-trained CNN for feature extraction from fine-tuned apple flowers images, followed by classification via a support vector machine. In subsequent work, Dias et al. [15] applied a fully end-to-end residual CNN for semantic segmentation, fine-tuned for apple flowers to enhance recognition sensitivity, and employed refinement strategies to improve segmentation precision and better delineate individual floral instances. Wu et al. [16] proposed a YOLOv4-based deep learning algorithm with channel pruning, built on the CSPDarknet53 framework. Fine-tuning with a dataset of 2,230 manually annotated apple blossom images enabled rapid and precise real-time detection. Zhang et al. [17] introduced a detection network optimized with generative modules and pruning-based inference, incorporating data augmentation and automatic deactivation of redundant network components. This significantly improved both detection accuracy and inference speed. Khanal et al. [18] utilized K-means clustering to identify centroids of individual flowers—both open and unopened—and associate them with floral clusters, facilitating accurate floral instance detection. Wang et al. [19] proposed YO-AFD, by embedding a novel ISAT attention mechanism into the C2f module to form C2f-IS, the model markedly enhances multi-scale feature representation and integration.

While the aforementioned algorithms have demonstrated promising performance in handling variations in flower shape and size, they often fall short in addressing challenges posed by occlusion and overlapping at flower edges. The introduction of the DETR algorithm [20] marked a paradigm shift in object detection by reframing it as a direct set prediction problem. Leveraging the self-attention mechanism of the Transformer architecture [21], DETR predicts object categories and bounding boxes in a fully end-to-end manner, enabling superior preservation of occluded object boundaries and thereby enhancing detection accuracy. With the growing adoption of DETR, numerous variants have emerged to adapt the framework to more diverse and complex scenarios. For example, Zhang et al. [22] proposed the DINO model, which improves DETR's performance through contrastive denoising training, hybrid query selection, and a dual look-ahead mechanism, achieving more robust end-to-end detection. Ma et al. [23] proposed an enhanced OptiDETR model, integrating a Swin Transformer-based encoder to improve feature extraction and capture both local and global context. The model introduces an IoU-aware query selection mechanism to prioritize object queries based on their localization quality, addressing the challenges of slow convergence and small-object detection in UAV images. Although DETR and its variants have achieved considerable success in general object detection, autonomous driving, video surveillance, and medical image analysis, their limitations in small object detection remain a significant bottleneck for agricultural applications. Nevertheless, the continued evolution of deep learning—particularly the incorporation of Transformer architectures into detection pipelines—has fueled further innovation in this domain. The recently proposed RT-DETR model [24] retains the core advantages of DETR while introducing a high-efficiency hybrid encoder and a minimum uncertainty query selection mechanism. This enables end-to-end real-time object detection without reliance on non-maximum suppression, while also supporting flexible trade-offs between detection speed and model complexity. These improvements render RT-DETR particularly well-suited for agricultural scenarios, notably in the context of precise apple flower detection. To address the specific challenges posed by tiny flower buds, occlusion, and dense overlapping of floral structures, we propose GLD-Net—an enhanced detection framework based on RT-DETR, optimized for assessing the maturity status of apple flower. The main contributions of our method are as follows: pipelines—has fueled further innovation in this domain. The recently proposed RT-DETR model preserves the core strengths of DETR while incorporating a high-efficiency hybrid encoder and a minimum uncertainty query selection strategy, enabling end-to-end real-time detection without reliance on non-maximum suppression. These advances support flexible trade-offs between speed and complexity, making RT-DETR particularly suitable for precision tasks in agriculture, such as apple flower detection. To tackle challenges such as tiny buds, occlusions, and dense floral overlap, we propose GLD-Net—an RT-DETR-based framework optimized for assessing apple flower maturity. The main contributions are as follows:

(1) To better capture both the fine-grained local features of small flower buds and stamens, as well as the global structural characteristics of entire flowers, we introduce the GLSA module. This module facilitates effective feature fusion and enhancement by replacing horizontal convolutions, thereby improving the representation of spatial and channel information and enriching the descriptive power of feature maps. Moreover, by substituting conventional input projection mechanisms, GLSA strengthens the contextual representation of input features, further promoting efficient integration across spatial hierarchies.

(2) To enhance the model's capacity for representing complex scene features, we integrate a Diversified Branch Block (DBB) module as an optimized version of the original RepC3 architecture. This modification deepens the fusion of multi-level features and substantially improves the model's ability to capture intricate visual patterns under diverse and challenging environmental conditions.

(3) To assess the performance of our model for apple flower maturity detection, we construct a new benchmark dataset named AppleFlowers. It consists of high-resolution images captured in complex orchard environments, encompassing a broad range of real-world conditions such as varying lighting, occlusions, and complex backgrounds

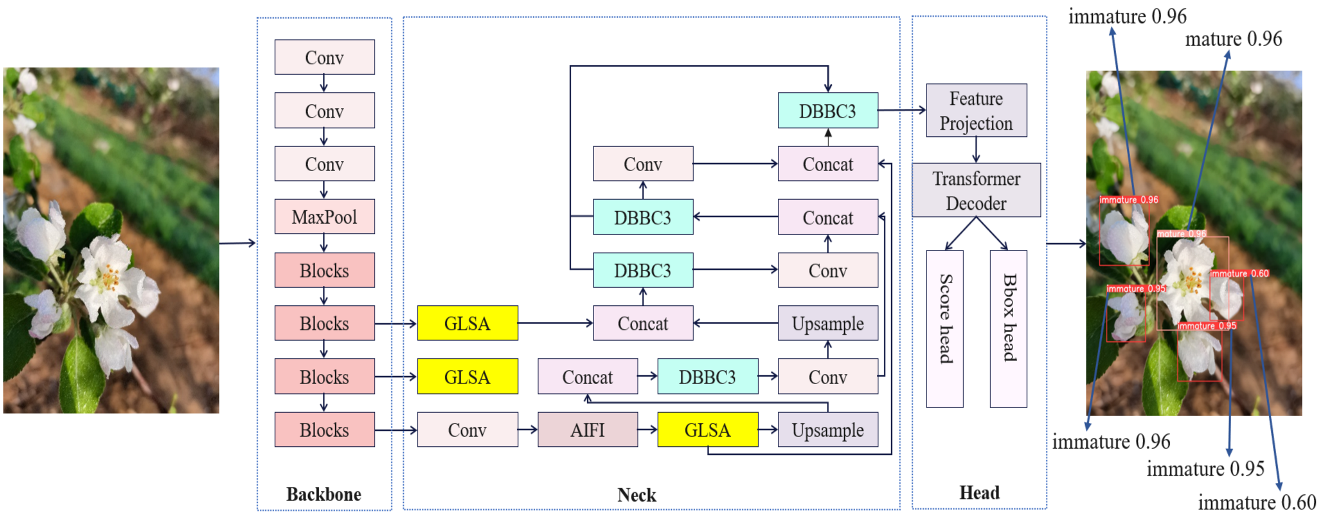


Fig. 1. Overall architecture of apple flower maturity detection model

II. GLD-Net Detection Model

To enhance the accuracy and efficiency of apple flower maturity detection, we propose an improved detection framework built upon the RT-DETR architecture, as illustrated in Fig.1. The overall network is composed of three primary components. First, the backbone module extracts rich semantic and spatial features through a series of convolutional and pooling operations, producing multi-scale feature maps. Next, the neck module refines and aggregates these multi-scale features using upsampling, attention mechanisms, and feature fusion strategies to capture both high-level semantics and fine-grained details. Finally, the detection head employs the RT-DETR decoder to perform end-to-end prediction of object categories and bounding box coordinates, leveraging the integrated features from the neck module to enable precise localization and classification.

To further enhance the model's representational capacity, the conventional horizontal convolution layers within the neck module are replaced with the GLSA module. This substitution significantly improves the encoding of spatial and channel-wise information in the feature maps. In addition, the input projection layers are also replaced with GLSA modules to strengthen contextual information extraction from the input features, thereby promoting more effective semantic integration. Furthermore, the model integrates a DBB module to improve the original RepC3 structure within the neck. Multiple DBB instances are deployed in parallel, each comprising a set of heterogeneous convolutional branches that process the input features through diverse receptive fields. This design facilitates deeper and more comprehensive feature fusion, resulting in enhanced feature representations that are better suited for complex detection scenarios.

A. Global-to-local spatial aggregation module

To enhance the representation of spatial and channel-wise information within feature maps, as well as to improve the contextual encoding of input features, we replace three conventional convolutional layers in the original architecture—including the horizontal convolution and input projection components—with the Global-to-Local Spatial Aggregation (GLSA) module [25]. Unlike standard attention mechanisms, the GLSA module features a more sophisticated design that enables adaptive channel adjustment—a capability rarely observed in conventional implementations.

The GLSA framework comprises two complementary attention units: Global Spatial Attention (GSA) and Local Spatial Attention (LSA). Specifically, the input feature set {Xi | i ∈ {2, 3, 4}} is evenly divided into two subgroups, denoted as X_i^1 and X_i^2, each corresponding to distinct spatial feature dimensions. This division allows the model to independently capture global structural cues and local discriminative details. The first subgroup is processed by the GSA module to extract holistic floral representations, while the second is passed through the LSA module, which focuses on enhancing localized features, such as petal and stamen edge characteristics. The outputs of both branches are subsequently fused and further refined via a 1×1 convolutional layer to ensure effective integration and dimensional alignment, as illustrated in Fig.2.

This is done to achieve feature fusion and dimensional adaptation. The following formula can mathematically express these operations:

$$X_i^1, X_i^2 = Split(X_i) \tag{1}$$

$$X_i' = C_{1\times1}(Concat(G(X_i^1), L(X_i^2))) \tag{2}$$

Where G indicates global attention and L indicates local attention. $X_i' \in R^{\frac{H}{8} X \frac{W}{8} X 32}$ represents the output feature map after processing and fusion.

The GSA and LSA modules are described below, focusing on their roles in processing local and global spatial features.

(1) GSA Module: The GSA module is designed to enhance long-range spatial dependencies between pixels across the entire image. By capturing such non-local interactions, GSA effectively complements the LSA mechanism, thereby enriching the model's ability to represent complex spatial structures. This extended spatial awareness plays a crucial role in boosting the descriptive power of the learned feature representations. The operation of the GSA module can be formally defined as follows:

$$Att_G(X_i^1) = Softmax((C_{1x1}(X_i^1))) \tag{3}$$

$$G_{sa}(X_i^1) = MLP(Att_G(X_i^1) \otimes X_i^1) + X_i^1 \tag{4}$$

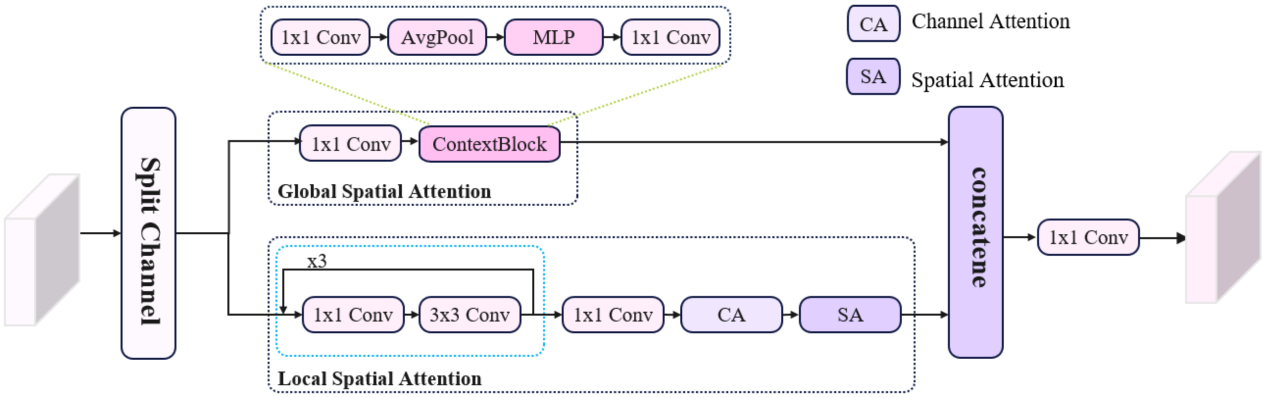


Fig.2. GLSA module structure diagram.

Global attention weights across the feature maps are generated using a 1×1 convolution followed by Softmax normalization, effectively capturing the spatial distribution of salient features. These weights, denoted as $\mathrm{Att_G}(\mathrm{X}_i^1)$, are applied to the input features via element-wise multiplication, enabling the network to selectively emphasize informative regions. The reweighted features are subsequently passed through a multi-layer perceptron (MLP), which consists of two fully connected layers separated by a normalization layer and a ReLU activation. The first layer expands the feature dimensionality by a factor of two to enrich the representation, while the second projects it back to the original dimension to ensure consistency in the output feature space. Finally, a residual connection adds the transformed features back to the original input, facilitating effective information fusion. Through this mechanism, the GSA module constructs an attention map G that not only preserves the intrinsic structure of the input feature map X_i^1, but also integrates global contextual cues highlighted by the attention mechanism. This integration significantly enhances the model's capacity to interpret complex visual scenes and supports more accurate inference in challenging vision tasks.

(2) The LSA module is specifically designed to enhance the detection of small objects by effectively capturing fine-grained local spatial features within the feature maps. To compute the local spatial attention response tensor $\mathrm{L} \in \mathrm{R}^{\frac{H}{8} \mathrm{X} \frac{W}{8} \mathrm{X}\, 32}$, the module takes the input feature map X_i^2 and applies a series of operations tailored to preserve detailed spatial cues.

$$Att_L\,(X_i^2) = \sigma(C_{1\times1}(X_c(X_i^2)) + X_i^2) \quad (5)$$

$$L = Att_L\,(X_i^2) \odot X_i^2 + X_i^2 \quad (6)$$

The component Fc consists of a sequence of three 1×1 convolutional layers followed by three 3×3 depthwise convolutional layers, collectively projecting the feature channels to a fixed dimensionality of 32. Within this framework, $\mathrm{Att_L}$ refers to the local attention mechanism, σ denotes the Sigmoid activation function, and ⊙ signifies element-wise multiplication.

The GLSA module integrates both global and local spatial attention mechanisms to holistically capture diverse and discriminative visual cues across varying spatial scales. This dual-attention design enhances the representational capacity of the model, particularly improving its sensitivity to subtle and small-scale features. In the context of the AppleFlowers dataset, the strengthened local spatial attention mechanism enables the model to better capture fine-grained details and edge structures, which are critical for accurate small object detection. Moreover, by employing a channel-split strategy, the GLSA module effectively balances computational overhead with representational fidelity, facilitating high-precision feature extraction without compromising inference efficiency.

B. Diversified convolutional branch module

To enhance the expressiveness of feature representations, we propose a novel module for the feature pyramid network (FPN), termed the Diversified Branch Block Convolutional (DBBC3) [26], which improves the conventional RepC3 module. The core component DBB, is specifically designed to improve the representational richness of FPNs by introducing a set of heterogeneous convolutional branches. Drawing inspiration from the Inception architecture [27-28], DBB employs a multi-path topology to expand the representational capacity of the feature space, akin to the human visual system's capability to process multi-scale information in parallel. By aggregating convolutional paths of varying receptive fields and computational complexities, DBB strengthens the ability of a single convolutional layer to capture diverse spatial patterns and semantic contexts. The architectural design of DBBC3 is illustrated in Fig.3. Consider an input feature map I composed of C channels. DBB aims to convert this input into an output feature map O while promoting feature diversity. The DBB process can be expressed as:

$$O = \mathcal{F}(I) \quad (7)$$

In this context, F represents a sequence of convolutions and nonlinear operations within DBB.

The DBBC3 module, a vital element of the Feature Pyramid Network, enables the fusion of multi-scale features through repeated applications of DBB. This design includes an input channel C1, an output channel C2, and an intermediate hidden channel C', with e denoting the expansion factor, calculated as follows：

$$C' = int(C_2 \times e) \quad (8)$$

The DBBC3 module contains n DBBs, each designed to progressively refine and integrate feature maps. The operation within the k-th DBB can be described as follows:

$$I_{out}^{(k)} = \mathcal{F}_k(I_{in}^{(k)}) \quad (9)$$

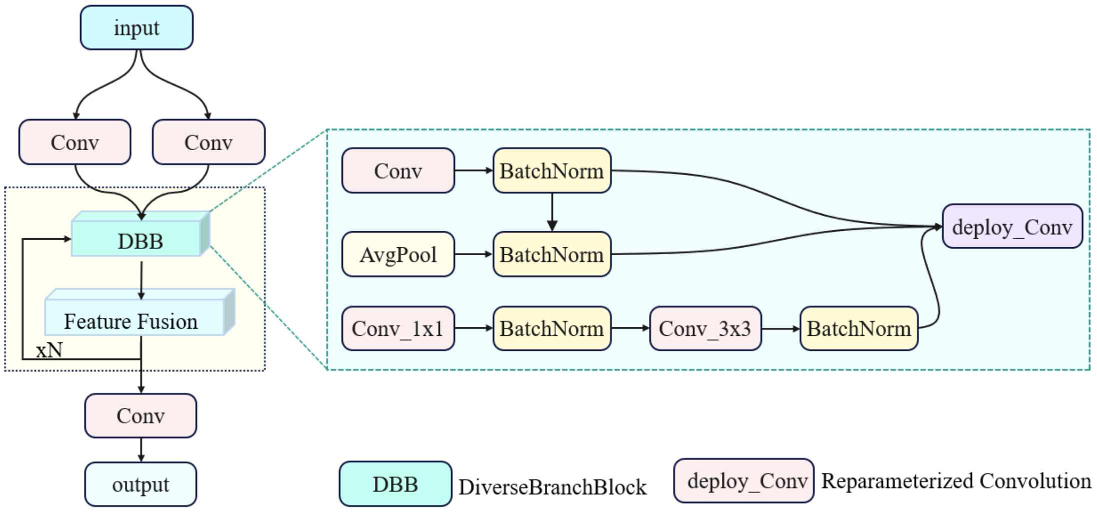


Fig.3. DBBC3 module structure diagram.

$I_{in}^{(k)}$ denotes the input feature maps to the k-th DBB while $I_{out}^{(k)}$ represents the corresponding output feature maps. The transformation functions within the k-th DBB are represented by F_k, which include both convolutional operations and nonlinear activations. By sequentially applying these transformations, the DBB module converts the input feature maps into their output counterparts.

In the DBBC3 module, feature integration is carried out by successively combining and enhancing the output feature maps from multiple DBBs. These hierarchically organized outputs are subsequently unified to generate a comprehensive and informative feature representation, providing a solid foundation for subsequent apple flower detection tasks. While the integration of DBBs increases model complexity during the training phase, this overhead is mitigated via structural reparameterization, which converts the multi-branch structure into an equivalent single convolutional operation at inference. This approach allows the DBBC3 module to harness diverse and informative features during training, while maintaining high inference efficiency—a vital characteristic for precision agriculture tasks such as apple flower maturity detection.

C. Loss Function

The formulation of an effective loss function is a critical component in achieving accurate detection of apple flower maturity during training. In this study, the total loss function L_{total} comprises three essential components: the bounding box regression loss L_{box}, the classification loss L_{cls}, and the L1 norm loss L_{L1}. These elements collectively guide the model to refine spatial localization, enhance class discrimination, and enforce numerical stability. The complete loss formulation is detailed in Equation (10).

$$L_{total} = L_{box} + L_{cls} + L_{L1} \tag{10}$$

The bounding box regression component adopts the Generalized Intersection over Union (GIoU) loss, which extends beyond conventional overlap metrics by incorporating both the spatial intersection and the geometric alignment—specifically, the relative shape and size—of predicted and ground truth boxes. This design is particularly well-suited for detecting small-scale objects such as apple flowers, which are often embedded within visually complex backgrounds. Owing to its sensitivity to geometric discrepancies, the GIoU loss maintains robust localization performance across varying flower morphologies, ensuring high detection accuracy even under shape deformations. The formulation of the GIoU loss is provided as follows:

$$L_{GIoU} = 1 \text{ - } IoU + \frac{1}{2}\left(\frac{\Delta w}{W+\epsilon} + \frac{\Delta h}{H+\epsilon}\right) \tag{11}$$

In this formulation, IoU represents the Intersection over Union, while Δw and Δh correspond to the differences in width and height between the predicted and ground truth boxes, respectively. The variables W and H denote the width and height of the ground truth box, and ϵ is a small constant introduced to ensure numerical stability.

The detection of apple flower maturity is framed as a binary classification task. To quantify the classification performance, the cross-entropy loss function is employed. A low loss value is achieved when the predicted probability closely aligns with the true label, whereas a significant discrepancy between the predicted and actual labels results in a higher loss. The formula for calculating the classification loss is as follows:

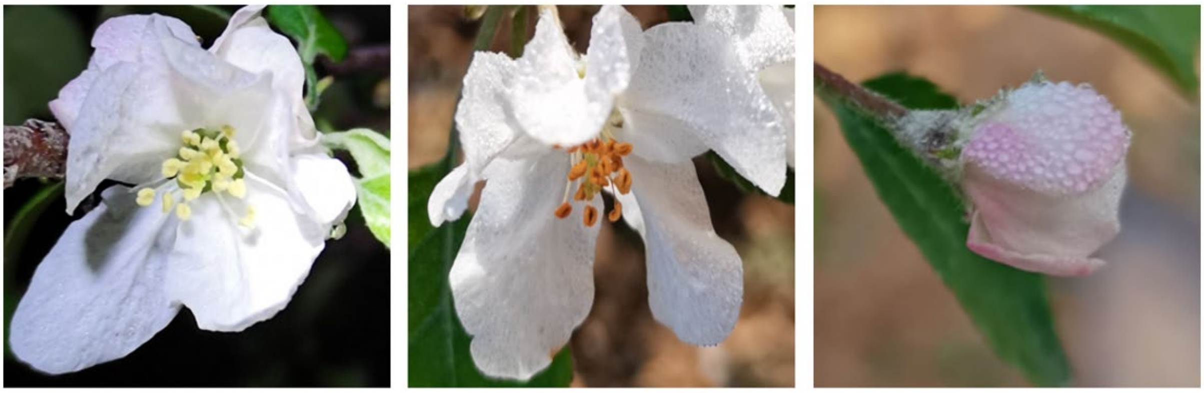
$$L_{cls} = -(y\log(p) + (1-y)\log(1-p)) \tag{12}$$

Here, y represents the true label, typically taking values of 0 or 1. The variable p indicates the probability assigned by the model for the sample belonging to category 1, while 1−p represents the probability of the sample being classified as belonging to category 0.

The L1 norm loss quantifies the absolute difference between the predicted and actual values. In the context of apple flower maturity detection, the L1 loss is employed for bounding box regression, where it aids in predicting the spatial location of the object. The formula for the L1 loss is as follows:

$$L_{L1} = \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{13}$$

The integration of these three loss functions endows the model with the ability to effectively adapt to diverse conditions, maintaining high accuracy and robustness in detecting variations across different lighting environments, background complexities, and flower developmental stages.



(a) Immature apple flower (b) Imature apple flower (c) Apple flower bud

Fig.4. Examples of the maturity of some apple flowers.

TABLE I
SUMMARY OF STATISTICS ON THE NUMBER OF APPLE FLOWERS IN DIFFERENT MATURITY STATES.

Area	Immature apple flowers	Mature apple flowers	All
Training	1320	1469	2789
Val	633	634	1267
Total	1953/48.15%	2103/51.85%	4056

III. Dataset

This study introduces the AppleFlowers dataset, specifically designed for detecting the maturity of apple flower under natural environmental conditions. Serving as a valuable resource for research in intelligent agriculture, the dataset is characterized by its complex and dynamic environments, diverse floral morphologies, and wide pplicability. By utilizing the AppleFlowers dataset in apple flower detection, this work aims to facilitate the development of more advanced intelligent orchard management systems, thereby contributing to the advancement of smart orchard technology.

A. Characteristic analysis of apple flowers

The petals and stamens of apple flowers exhibit distinct differences in both color and texture. Typically, apple flowers consist of five white petals arranged in a plum-flower pattern. In the early stages, the buds are pink or light pink, with the petals remaining closed. During the unripe phase, the anthers appear pale yellow and are clustered together. As the flowers mature, the anthers gradually transition from pale yellow to orange, with their arrangement shifting from a clustered form to a radiating pattern, extending outward in a chuan-like configuration, as depicted in Fig.4.

B. AppleFlowers dataset

To assess the changes in the maturity of apple flowers from a multi-scale and multi-modal perspective and to evaluate the performance of models in detecting and classifying apple flower maturity, we have developed a novel dataset, AppleFlowers. The dataset was constructed using an intelligent monitoring system designed to simulate an orchard environment. Images were captured from multiple perspectives to ensure the dataset's applicability and diversity. The following section provides a detailed description of the image acquisition process and the construction of the AppleFlowers dataset.

The specific parameters for image capture are as follows:

Location: Zhangjiazhuang Village, Yuezhuang Town, Yiyuan County, Zibo City, Shandong Province (118°29'N, 36°23'E)

Apple cultivar: Red Fuji

Image capture period: Late April to early May 2024

Capture hours: 06:00 to 22:00

Capture device: HUAWEI Nova 7 smartphone

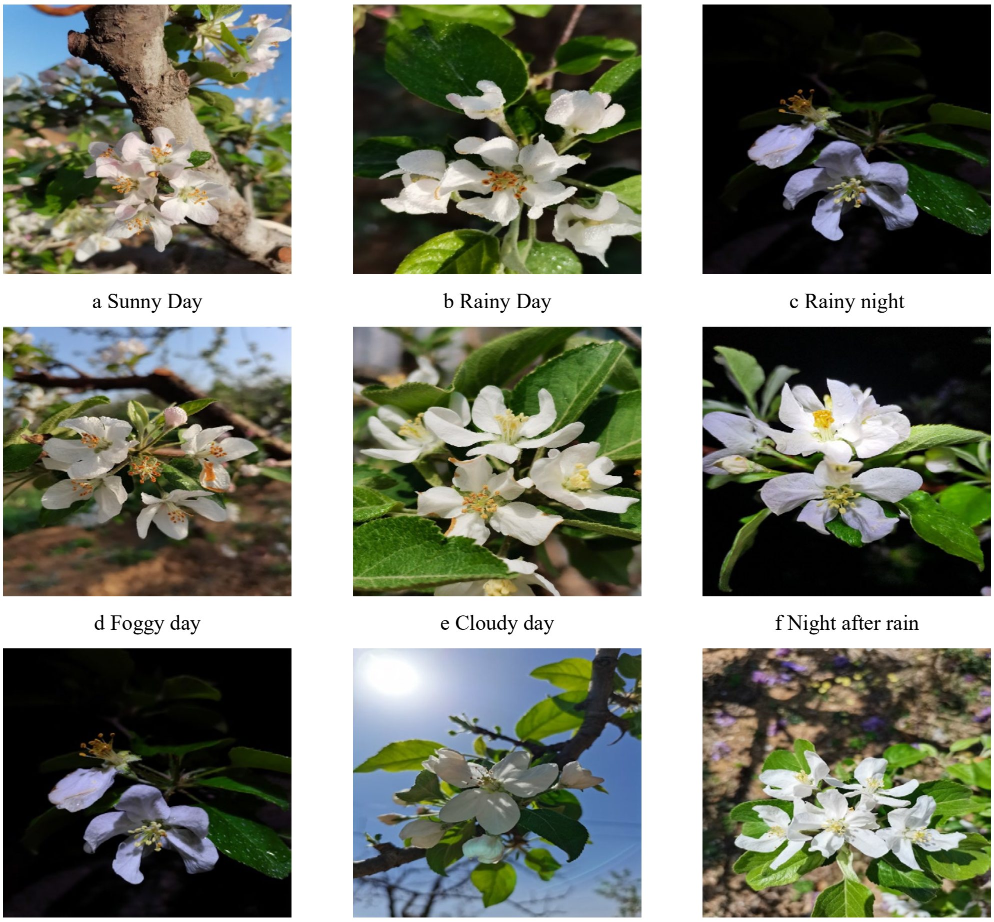
Image resolution: 3456 x 4608 pixels and 4608 x 3456 pixels

Storage format: JPG

All images were captured in the natural environment of the apple orchard, with the surrounding soil, sky, and trees providing the background. A variety of photographic techniques were utilized, including front lighting, backlighting, close-up shots, long-range perspectives, as well as downward and upward angles. The dataset covers a broad spectrum of natural conditions, including clear skies, overcast, fog, rain, post-rain, daytime, and nighttime scenes.

C. Dataset production

A total of 1,286 raw images were initially collected. After a rigorous screening process to remove duplicates and images with excessively blurred petal edges, 1,000 images were retained. These images were annotated using LabelImg software. A selection of original images from the dataset is shown in Fig.5. The dataset was then randomly divided into training and test sets at a 7:3 ratio. The distribution of flowers with different labels in both sets was analyzed. The statistical breakdown of flower labels in the training and test sets is presented in Table 1. The distribution of immature and mature apple flowers in the dataset is 48.15% and 51.85%, respectively.



a Sunny Day b Rainy Day c Rainy night

d Foggy day e Cloudy day f Night after rain

g Clear night h Backlight i Frontlight

Fig.5. Some sample images from AppleFlowers

IV. Experiments and results

To evaluate the effectiveness of RT-DETR in detecting the maturity state of apple flowers, a series of experiments were conducted. We first describe the experimental platform and detail the model's implementation during both training and testing. The model was trained on the AppleFlowers dataset, and the best-performing version was selected for evaluation on the test set, followed by a comprehensive analysis. Finally, under identical conditions, several state-of-the-art detection models were evaluated and compared, with a subsequent analysis of the differences observed in their performance.

A. Experimental platform and evaluation indicators

All experiments in this study were conducted on an identical server system configured with Ubuntu 18.04 (64-bit), an NVIDIA GeForce RTX 3090 24GB GPU, and a CUDA V11.4 environment. The models were developed using Python 3.8 and PyTorch 1.10.0, with the model components built using the MMdetection v3.0.0 library.

The initial learning rate was set to 0.0001, with a weight decay of 0.0001 to mitigate overfitting. To optimize the network parameters and efficiently minimize the loss function, the AdamW optimizer was employed with a momentum of 0.9. The RT-DETR model was trained for 100 epochs. The final precision-recall curve for the model, based on the AppleFlowers dataset, are illustrated in Fig.6.

In this experiment, precision and recall, commonly used metrics for binary classification, were employed to assess the efficiency of the detection model. Precision indicates the proportion of true positives among predicted positives, as defined in Equation (14), while recall measures the proportion of actual positives correctly identified, as described in Equation (15).

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{14}$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{15}$$

TP is the number of true positive examples, FP is the number of false positive examples, FN is the number of false negative examples, and TN is the number of true negative examples.

To comprehensively evaluate the model's detection performance, three key metrics were additionally employed in this study: Average Precision (AP), Average Recall (AR), and mean Average Precision (mAP). These were computed using Equations (16), (17), and (18), respectively, under specified IoU thresholds.

$$AP = \frac{1}{N}\sum_{k=1}^{N} Precision_i \tag{16}$$

$$\mathrm{AR} = \frac{1}{\mathrm{N}}\sum_{k=1}^{101} Recall_i \tag{17}$$

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

Here, N is the number of classes and AP_i denotes the average precision for the i-th class. The mAP is computed as the mean AP across all classes. In this study, N = 2, reflecting the two maturity stages of apple flowers.

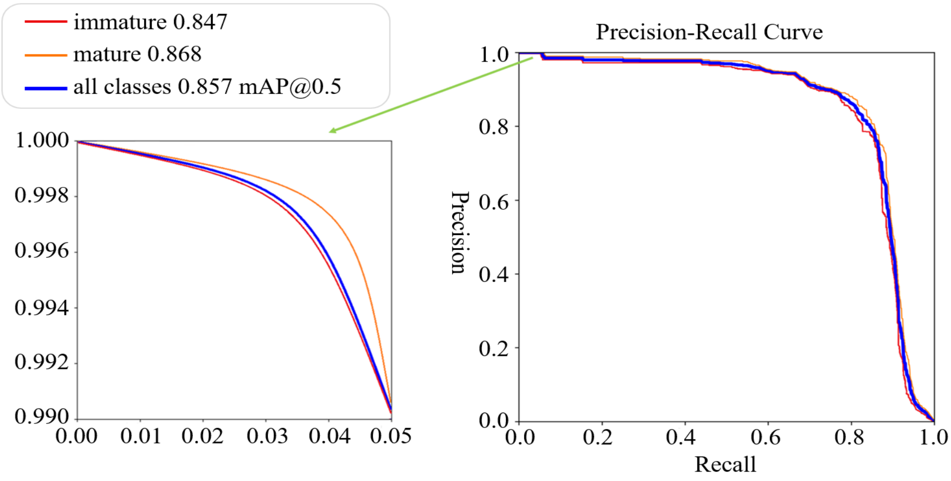


Fig. 6. Precision-recall curve.

TABLE II
EVALUATION RESULTS OF DETECTING TWO TYPES OF MATURE STATES.

Type	Precision	Recall	mAP$_{50}$	mAP$_{50-95}$
Immature	83.7%	82.2%	87.1%	76.7%
Mature	92.8%	74.3%	84.8%	72.3%

A detection is considered valid if the predicted class of the apple flower's maturity state is correct and the IoU exceeds a predefined threshold, which was set to 0.5 in this study. The model's evaluation results for detecting the two distinct maturity stages in the AppleFlowers dataset are summarized in Table 2. The corresponding detection performance for each maturity category is visually illustrated in Fig.7.

B. Ablation experiment

In order to verify the effectiveness of the GLSA module and the DBBC3 module, ablation experiments were conducted in this section to compare the effects of GLSA and DBBC3 based on the original backbone network, which further verified the effectiveness of the two methods. The results are shown in Table 3.

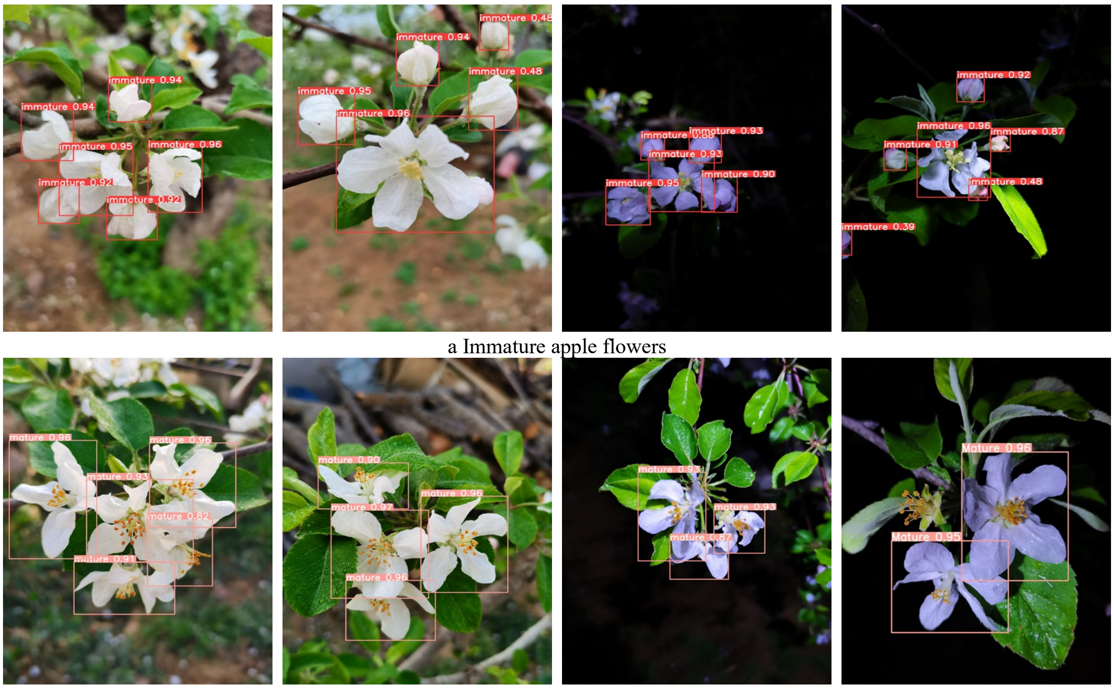
As detailed in Table 3, the baseline RT-DETR model—without any additional architectural modifications—achieved mAP scores of 79.9% at IoU=0.5 and 67.4% across IoU thresholds from 0.5 to 0.95, with a computational complexity of 56.9 GFLOPs. Upon replacing the original convolutional layers in the Neck with the GLSA module, detection performance showed a marked improvement: mAP$_{50}$ increased by 3.1% and mAP$_{50–95}$ by 2.6%. Further performance gains were observed with the addition of the DBBC3 module, yielding a 5.1% rise in mAP$_{50}$ and a 6.7% increase in mAP$_{50–95}$, thereby validating the module's effectiveness in enhancing feature representation. The integration of both GLSA and DBBC3 into the baseline architecture resulted in a substantial boost in detection accuracy, achieving mAP$_{50}$ and mAP$_{50–95}$ scores of 86.0% and 74.5%, respectively. These findings affirm the efficacy of the proposed architectural enhancements and highlight the improved discriminative capacity of the modified RT-DETR model in classifying apple flower maturity states.

In terms of model complexity, the baseline configuration comprised 19.87 million parameters. Incorporating the GLSA module increased the parameter count by 2.08 million and added 6.8 GFLOPs to the computational load, primarily because of the attention mechanisms and graph convolution operations intrinsic to GLSA. These components introduce additional computational steps during both forward and backward propagation to facilitate more abstract and context-aware feature encoding. Notably, the integration of the DBBC3 module incurred no further increase in parameter count or computational complexity. However, its synergy with GLSA contributed to a significant enhancement in model performance, suggesting a complementary interaction that supports more robust object detection under varied and complex orchard conditions.

C. Algorithm comparison

To further assess the detection performance of the improved RT-DETR model, a comprehensive comparative analysis was conducted against a suite of both classical and state-of-the-art object detection algorithms. This evaluation encompassed two-stage detectors—including Faster R-CNN [29], RetinaNet [30], Cascade R-CNN [31], EfficientNet [32], FCOS [33], YOLOF [34], and DDQ [35]—as well as leading one-stage models, namely YOLO v10 [36] and YOLO v11[37]. For the YOLO series, due to their smaller number of layers, the YOLO v10l and YOLO v11l models were selected, enabling a more balanced comparison in terms of parameters and complexity. All models were trained and evaluated under identical experimental conditions and computational environments to ensure consistency and fairness.

As shown in Table 4, the proposed model demonstrates outstanding performance across key evaluation metrics, achieving mAP$_{50}$, mAP$_{75}$, and mAP$_{50–95}$ scores of 86.0%, 79.3%, and 74.5%, respectively, thereby substantially outperforming existing state-of-the-art detectors. Compared to representative two-stage models, GLD-Net surpasses the EfficientNet by 5.4% in mAP$_{50}$ and achieves a notable 15.5% improvement in AP over the anchor-free FCOS model. In addition, relative to the advanced one-stage YOLOv11l.



a Immature apple flowers

b Mature apple flowers

Fig.7. Apple flowers maturity status detection results

TABLE III
THE INFLUENCE OF GLSA AND DBBC3 ON EXPERIMENTAL RESULTS.

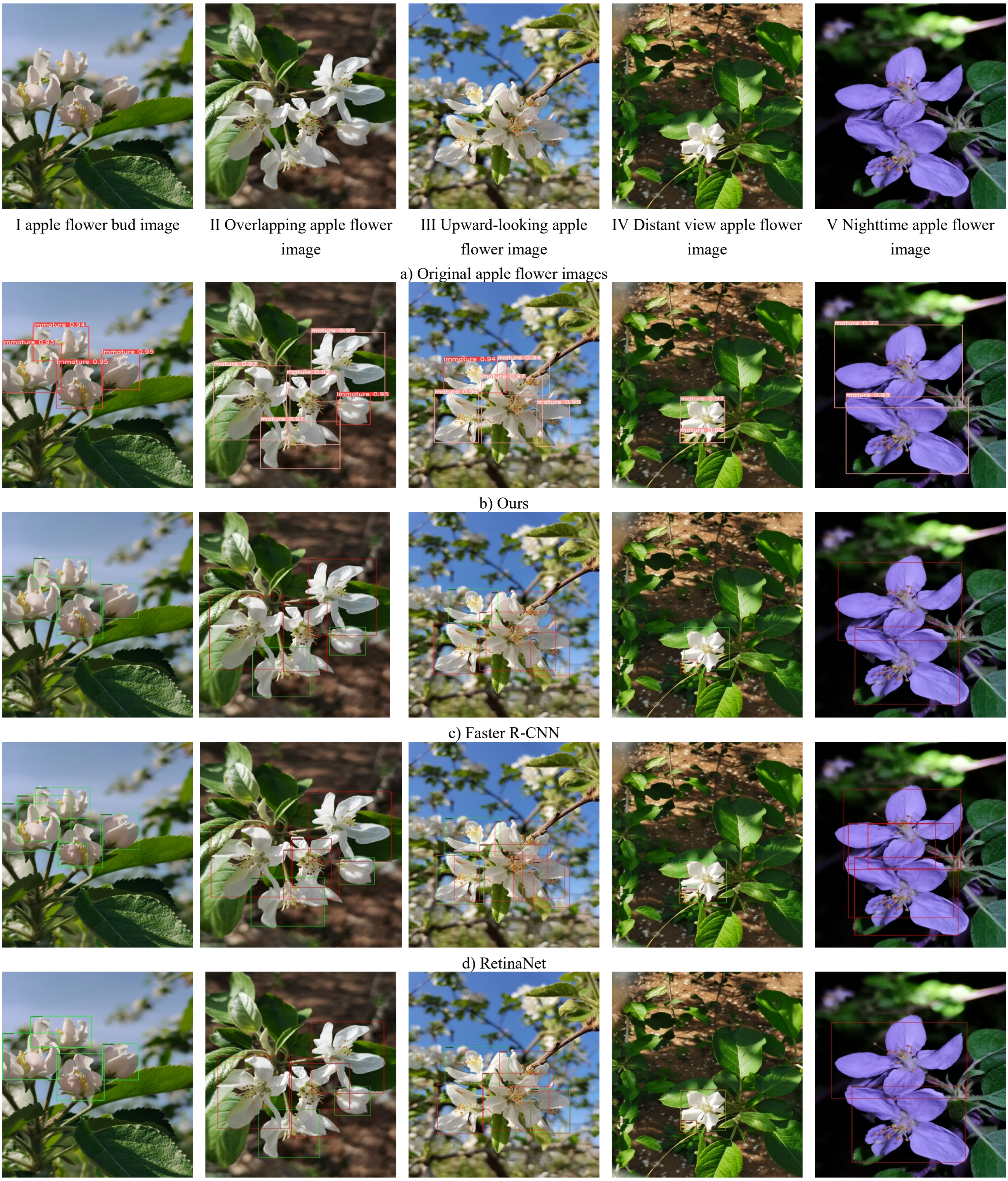
Model	GLSA	DBBC3	Precision	Recall	mAP_{50}	$mAP_{50\text{-}95}$	Params	Flops/GFlpos
RT-DETR	×	×	82.6%	78.8%	79.9%	67.4%	19.87M	56.9
	✓	×	86.1%	78.1%	83.0%	70.0%	21.95M	63.7
	×	✓	82.2%	79.1%	85.0%	74.1%	19.87M	56.9
Ours	✓	✓	88.2%	78.2%	86%	74.5%	21.95M	63.7

TABLE IV
THE DETECTION RESULTS OF EACH DETECTION MODEL

Types	Model	AP(%)	AR(%)	$AR_{max=100}$(%)	mAP_{50}(%)	map_{75}(%)	$mAP_{50\text{-}95}$(%)
Two-stage	Faster R-CNN	68.4	70.6	54.7	72	48	43.9
	RetinaNet	71.5	57.0	63.3	76.7	53.7	48.4
	Cascade R-CNN	71.6	76.2	62.8	79.4	57.1	51.2
	EfficientNet	74.3	68.1	68.1	80.6	60	52.9
	FCOS	72.7	55.9	63.3	78.2	48.1	46.4
	YOLOF	68.1	76.3	65.5	76.5	53.8	49.2
	DDQ	70.6	74.9	**83.6**	76.7	65.5	61.1
One-stage	YOLO v10l	84.4	79.2	79.6	85.9	73.1	66.4
	YOLO v11l	84	**79.7**	82.6	85.5	76.2	71.8
	Ours	88.2	78.2	80.2	86	79.3	74.5

TABLE V
COMPARISON OF NUMBER OF PARAMETERS AND COMPUTATIONAL OVERHEAD

Model	Faster R-CNN	RetinaNet	Cascade R-CNN	EfficientNet	FCOS	YOLOF	DDQ	YOLO v10l	YOLO v11l	Ours
Params/M	41.37	36.35	119	54.4	32.11	42.36	48.3	25.71	25.3	21.95
GFlops	91.47	81.74	69.15	18.36	78.58	39.19	119	126.3	87.3	63.7



I apple flower bud image

II Overlapping apple flower image

III Upward-looking apple flower image

IV Distant view apple flower image

V Nighttime apple flower image

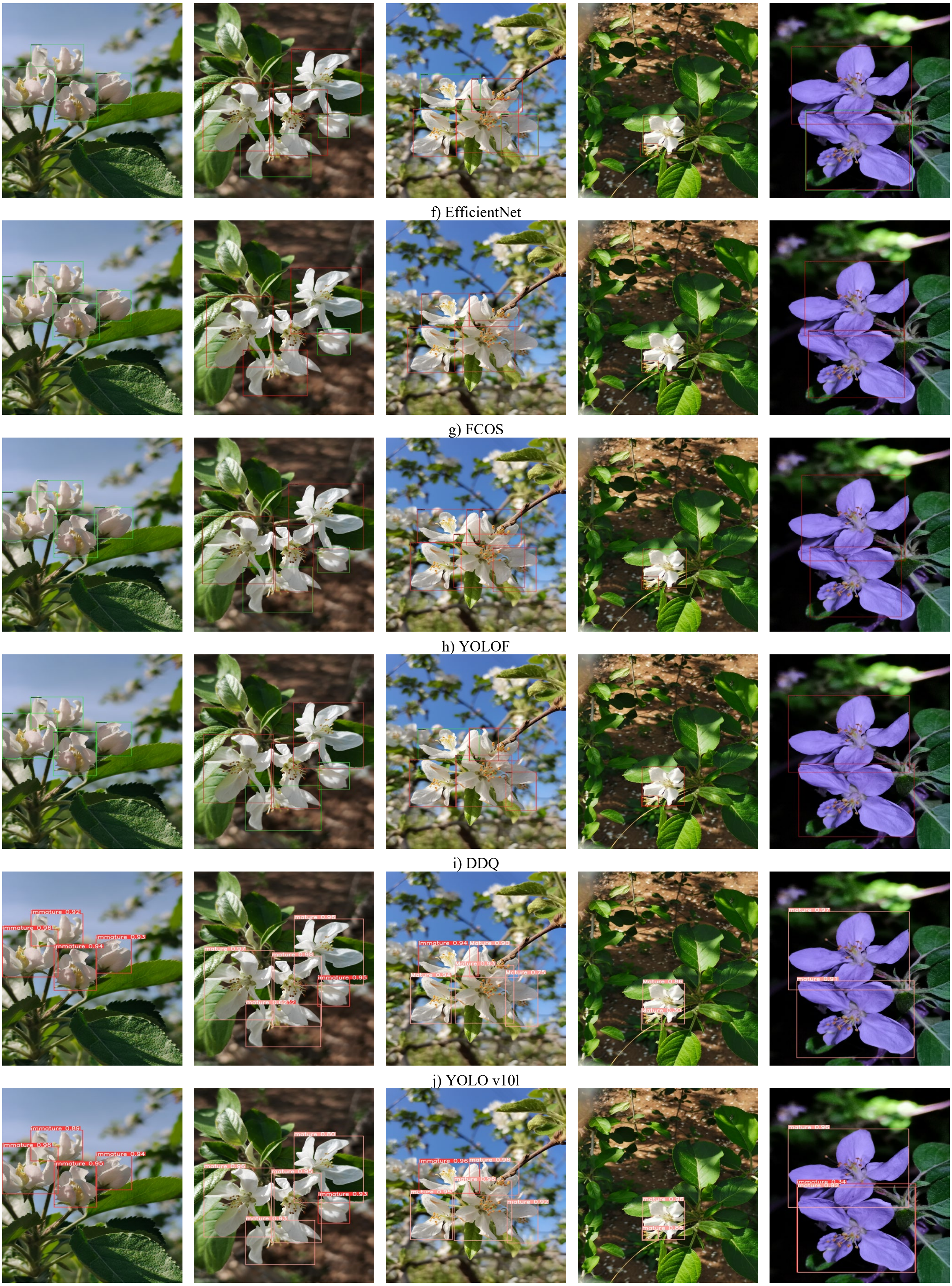
a) Original apple flower images

b) Ours

c) Faster R-CNN

d) RetinaNet

e) Cascade R-CNN



f) EfficientNet

g) FCOS

h) YOLOF

i) DDQ

j) YOLO v10l

k) YOLO v11l

Fig.8. Comparison of detection results on the AppleFlowers dataset.

Note: In the two-stage model images, red boxes indicate mature apple flowers, while green boxes denote immature apple flowers. In the one-stage model image, the pink boxes represent mature apple flowers, while the red boxes represent immature apple flowers.

detector, GLD-Net achieves improvements of 4.2% in AP and 0.5% in mAP$_{50}$. In terms of AR$_{max=100}$, its performance trails the leading DDQ model by a marginal 3.4%.

Although detection accuracy is paramount, model efficiency—characterized by parameter count and computational complexity—plays a crucial role in the practical applicability of detection frameworks. At an input resolution of 640×640, Table 5 summarizes the parameter count and computational overhead for each model. GLD-Net contains only 21.95 million parameters, offering superior parameter efficiency relative to other models. It further maintains a low computational cost of 63.7 GFLOPs, suggesting modest resource demands while preserving high detection accuracy.

The comparative experimental results highlight that our model not only maintains high detection accuracy but also achieves a low parameter count and computational complexity, making it highly suitable for deployment in field environments with constrained computational resources. Fig.8 presents a visual comparison of the detection results produced by various state-of-the-art algorithms on the AppleFlowers dataset, covering challenging scenarios such as occlusion, upward-looking views, long-shot perspectives, and nighttime conditions. While our model demonstrates promising performance in the current experiment, future work will focus on further optimizing the model's speed and efficiency to better accommodate the requirements of real-time or near-real-time detection applications.

V. Conclusion

This study addresses the challenge of detecting the maturity states of apple flower by proposing an optimized RT-DETR-based detection model. By integrating the GLSA and DBBC3 modules, the model significantly enhances both detection accuracy and robustness. The GLSA module enriches the feature maps by improving channel and spatial information representation, while also providing contextual information for the input features. In parallel, the DBBC3 module enhances the representational capacity of individual convolutional layers by integrating multiple convolutional branches with varying scales and complexities. Experimental results on the AppleFlowers dataset demonstrate the model's high precision and recall for detecting apple flower maturity states, achieving AP and mAP50 scores of 88.2% and 86%, respectively. While the model has shown notable improvements in detection accuracy, future work could explore further optimization, focusing on lightweight design and accelerated performance to meet the demands of real-time applications. With ongoing advancements in deep learning and the modernization of agriculture, the proposed model holds substantial promise for advancing precision agriculture and intelligent detection, offering significant potential for future research and deployment.

References

[1] Samnegård U, Hambäck P, Smith H. Pollination treatment affects fruit set and modifies marketable and storable fruit quality of commercial apples. Royal Society open science, 2019, 6(12): 190326.

[2] Houetohossou S, Houndji V, Hounmenou C, et al. Deep learning methods for biotic and abiotic stresses detection and classification in fruits and vegetables: State of the art and perspectives. Artificial Intelligence in Agriculture, 2023, 9: 46-60.

[3] Maupilé L, Chaib J, Boualem A, et al. Parthenocarpy, a pollination-independent fruit set mechanism to ensure yield stability. Trends in Plant Science, 2024, 29(11): 1254-1265.

[4] Földesi R, Howlett BG, Grass I, et al. Larger pollinators deposit more pollen on stigmas across multiple plant species—a meta-analysis. Journal of Applied Ecology, 2021, 58(4): 699-707.

[5] Zhang M, Su H, Wen J. Classification of flower image based on attention mechanism and multi-loss attention network. Computer Communications, 2021, 179: 307-317.

[6] Lodh A, Parekh R. Flower recognition system based on color and GIST features. Devices for Integrated Circuit. IEEE, 2017: 790-794.

[7] Deng Y, Wu H, Zhu H. Recognition and counting of citrus flowers based on instance segmentation. Transactions of the Chinese Society of Agricultural Engineering, 2020, 36(7): 200-207.

[8] Xiao K, Yang H, Su Z, et al. Identification and counting of phalaenopsis flowers based on improved YOLOv5. Journal of Chinese Agricultural Mechanization, 2023, 44(11): 155-161.

[9] Zhang X, Hu G, Li P, et al. Recognizing safflower using improved lightweight YOLOv8n. Transactions of the Chinese Society of Agricultural Engineering, 2024, 40(13): 163-170.

[10] Jia W, Wang Z, Zhang Z, et al. A fast and efficient green apple object detection model based on Foveabox. Journal of King Saud University-Computer and Information Sciences, 2022, 34(8): 5156-5169.

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

[12] Zheng Y, Sui X, Jiang Y, et al. SymReg-GAN: symmetric image registration with generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5631-5646.

[13] Yu Q, Ouyang X, Su B, et al. Vehicle detection algorithm in complex scenes based on improved YOLOv8. IAENG International Journal of Computer Science, vol. 52, no. 4, pp 886-893, 2025.

[14] Dias P A, Tabb A, Medeiros H. Apple flower detection using deep convolutional networks. Computers in Industry, 2018, 99: 17-28.

[15] Dias P A, Tabb A, Medeiros H. Multispecies fruit flower detection using a refined semantic segmentation network. IEEE Robotics and Automation Letters, 2018, 3(4): 3003-3010.

[16] Wu D, Lv S, Jiang M, et al. Using channel pruning-based yolo v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 2020, 178: 105742.

[17] Zhang Y, He S, Wa S, et al. Using generative module and pruning inference for the fast and accurate detection of apple flower in natural environments. Information, 2021, 12(12): 495.

[18] Khanal S R, Sapkota R, Ahmed D, et al. Machine vision system for early-stage apple flowers and flower clusters detection for precision thinning and pollination. IFAC-PapersOnLine, 2023, 56(2): 8914-8919.

[19] Wang D, Song H B, Wang B. YO-AFD: an improved YOLOv8-based deep learning approach for rapid and accurate apple flower detection. Frontiers in plant science, 2025, 16: 1541266.

[20] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2020: 6568–6577.

[21] Han K, Xiao A, Wu E, et al. Transformer in transformer. Advances in Neural Information Processing Systems, 2021, 34: 15908-15919.

[22] Zhang H, Li F, Liu S, et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv: 2203.03605, 2022.

[23] Wenlong Ma, and Weisheng Liu, "Improving UAV Image Target Detection: A Novel Approach Using OptiDETR with Swin Transformer," IAENG International Journal of Computer Science, vol. 52, no. 3, pp771-780, 2025.

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

[25] Tang F, Xu Z, Huang Q, et al. DuAT: Dual-aggregation transformer network for medical image segmentation. Chinese Conference on Pattern Recognition and Computer Vision, 2023: 343-356.

[26] Ding X, Zhang X, Han J, et al. Diverse branch block: Building a convolution as an inception-like unit. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 10886-10895.

[27] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence. 2017, 31(1): 4278-4284.

[28] Si C, Yu W, Zhou P, et al. Inception transformer. Advances in Neural Information Processing Systems, 2022, 35: 23495-23509.

[29] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6): 1137-1149.

[30] Ross T Y, Dollár G. Focal loss for dense object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2980-2988.

[31] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6154-6162.

[32] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning, 2019: 6105-6114.

[33] Detector A F O. Fcos: a simple and strong anchor-free object detector. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(4): 1-12.

[34] Chen Q, Wang Y, Yang T, et al. You only look one-level feature. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13039-13048.

[35] Zhang S, Wang X, Wang J, et al. What are expected queries in end-to-end object detection?. arXiv preprint arXiv: 2206. 01232, 2022.

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

[37] Khanam R, Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements. Computing Research Repository, 2024, abs/2410.177.