Applied Soft Computing Journal 186 (2026) 114137

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

End-to-end plant disease detection using transformers with collaborative hybrid assignment training

Yanfen Li [a], Muhammad Fayaz [b], Sufyan Danish [b], Lilia Tightiz [c], Hanxiang Wang [a], Tan N. Nguyen [d,*], L. Minh Dang [e,f,**]

[a] School of Computer Science, Qufu Normal University, Rizhao, 276826, China
[b] Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea
[c] School of Computing, Gachon University, 1342 Seongnamdaero, Seongnam-si, Gyeonggi-do, 13120, South Korea
[d] Department of Architectural Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, South Korea
[e] The Institute of Research and Development, Duy Tan University, Da Nang, 550000, Viet Nam
[f] Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Viet Nam

HIGHLIGHTS

- A large fruit disease dataset of 6 different diseases containing over 81,000 images.
- An efficient transformer-based fruit disease detection framework.
- Analysis of the disease region using the transformer's feature discriminability scores.
- The proposed model outperformed previous state-of-the-art object detection models.

ARTICLE INFO

Keywords:
Image processing
Transformer
Deep learning
Precision agriculture
Fruit disease

ABSTRACT

Plant diseases pose a significant threat to fruit production and quality if not detected and managed promptly. Precise and efficient recognition of these diseases is critical for ensuring plant health and maximizing fruit production. To tackle this issue, a range of image processing and deep learning techniques have been preferred for plant disease recognition due to their superior performance. This paper proposes an end-to-end transformer-based model that improves both the accuracy and detection rate of fruit diseases. The model is based on a state-of-the-art transformer model and trained using the Collaborative Hybrid Assignment (Co-DETR) scheme. Moreover, several targeted modifications to the original model are conducted to optimize its performance. These modifications enable the model to detect six types of plant diseases with a mean average precision (mAP) of 0.89 while maintaining efficient training times. The proposed model consistently outperforms state-of-the-art detection models. In addition, the model offers interpretability through the visualization of feature discriminability scores to ensure that the prediction process is interpretable and understandable. Finally, the model demonstrates robust performance under challenging environmental conditions, such as poor lighting and image blurring, which are essential for real-world applications in disease management and precision agriculture.

1. Introduction

According to the Food and Agriculture Organization (FAO), global food demand is projected to surge by 70 % by 2050 as the world population surpasses 9.1 billion [1]. Fruits, as critical sources of essential nutrients, play a pivotal role in ensuring food security and combating malnutrition [2]. However, it is increasingly challenging to achieve and sustain high fruit yields due to factors such as limited farmland, climate change, and the devastating impact of pests and diseases [3]. Among these threats, fruit diseases, such as mango scab and citrus thrips, often cause catastrophic yield losses and economic devastation when left undetected.

* Corresponding author.
** Corresponding author at: The Institute of Research and Development, Duy Tan University, Da Nang, 550000, Viet Nam.
Email addresses: tnnguyen@sejong.ac.kr (T.N. Nguyen), danglienminh@duytan.edu.vn (L.M. Dang).

https://doi.org/10.1016/j.asoc.2025.114137
Received 7 May 2025; Received in revised form 3 September 2025; Accepted 27 October 2025
Available online 28 October 2025

Traditional fruit disease detection relied on manual inspection, a labor-intensive and error-prone process with delays in identifying early-stage symptoms [4]. The lag between symptom appearance and detection often results in significant losses. To address these challenges, automated detection systems using machine learning (ML) and deep learning (DL) have emerged as transformative solutions for scalable, accurate, and efficient disease monitoring [5].

While early ML approaches utilized handcrafted features, such as color, texture, and shape, with classifiers like support vector machines (SVMs) [6] and random forests (RFs) [7], their performance was constrained by domain-specific feature engineering and environmental variability [3]. Recent DL advancements, particularly convolutional neural networks (CNNs), have demonstrated superior performance in disease classification [8], segmentation [9], and detection [10,11]. However, CNN-based models often require manual hyperparameter tuning, such as anchors, proposals, and post-processing to reduce redundant predictions [12].

Transformers were initially developed for natural language processing (NLP). Their self-attention mechanisms [13] enable global context modeling, which addresses CNN limitations in capturing long-range dependencies [12]. For instance, Longformer [14] introduced sliding window attention to process long documents efficiently. Reformer [15] reduced computational complexity using locality-sensitive hashing for large-scale NLP tasks. Beyond NLP, the adaptability of transformers was further enhanced by specialized variants customized to domain-specific challenges. For example, in finance, transformer variants have been trained to model temporal patterns and forecast price movements [16,17], while in remote sensing and fault detection, they have enabled precise anomaly identification in high-resolution imagery and industrial systems [18]. In manufacturing, transformers powered quality inspection and predictive maintenance [19]. In protein sequence modeling, Performer with kernel-based attention was introduced to effectively model the scalable protein sequence [20]. However, their application to fruit disease detection remains underexplored, with challenges in convergence, data scarcity, and subtle symptom recognition in complex agricultural environments [21].

To bridge this gap, this study introduces FD-TR, a modified transformer-based fruit disease detection model based on Co-DETR training scheme [22]. The key contributions of this study are:

- The proposed model was trained on a large-scale fruit disease dataset of 81,000 high-resolution images.
- Key modules (e.g., loss, optimizer) of Co-DETR were replaced, and hyperparameters were fine-tuned to address the unique challenges of fruit disease detection.
- Introduction of a feature discriminability score visualization method to enhance model interpretability for real-world deployment.
- The model demonstrated its robustness through systematic evaluation across four benchmark datasets and an additional healthy fruit subset.

The outline of the manuscript is as follows. Section 3 provides a comprehensive description of the fruit disease dataset used in this study. Section 4 discusses in detail each component of the proposed fruit disease detection framework based on DINO with a Co-DETR training scheme. The results of various experiments conducted to evaluate the model's performance are reported in Section 5. Section 6 discusses the main contributions and experimental results of this study. Finally, Section 7 provides conclusions and outlines future research directions.

2. Related work

Table 1 provides an overview of recent fruit disease detection studies. It highlights the diversity of models used, ranging from CNN models to hybrid and transformer-based architectures, applied to various fruit types. While most models achieved high accuracy on their respective datasets, the majority were limited by small sample sizes, limited disease coverage, and lack of real-world deployment validation. These limitations emphasize the need for a more generalized, scalable, and interpretable solution.

2.1. Traditional machine learning approaches

Early efforts in fruit disease detection focused on ML models using handcrafted features. For instance, SVMs trained on color and texture features achieved moderate success in classifying diseases on fruits [6]. RFs were employed to distinguish apple fruit diseases based on color and texture descriptors [7]. However, these methods struggled with environmental variability and required extensive domain expertise for feature design [3].

2.2. Deep learning-based approaches

The developments of CNNs revolutionized fruit disease detection. One-stage detectors like You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD), and two-stage frameworks such as Region-based CNN (R-CNN), were progressively adopted for precise recognition of fruit diseases [3]. For example, Sun et al. [11] introduced an innovative method for identifying fruit diseases in natural orchard settings using a combination of binocular cameras and DL techniques. They implemented a Unimatch stereo-matching algorithm to generate depth maps that focused detection on leaves and proposed a lightweight disease detection model based on YOLOv5-augmented with shuffle-channel blocks and attention modules. The experimental results revealed that it outperformed the YOLOv5-s architecture with 0.93 mean average precision (mAP). Syed et al. [25] presented a two-stage CNN for citrus disease detection. Firstly, the model employed a region proposal network to identify potential diseased areas on citrus leaves. After that, it classified these regions into specific disease categories using a classifier. The model demonstrated a high detection accuracy of 94.37 % for citrus black spot, citrus bacterial canker, and Huanglongbing. In another study, Xie et al. [23] addressed real-time detection of common grape leaf diseases using a customized Faster R-CNN with Inception-v1, Inception-ResNet0-v2, and SE-block. The model achieved a mAP of 0.81 at a real-time detection speed of 15.01 frames per second (FPS). Although these DL models enabled early and accurate disease detection, they still required manually fine-tuned hyperparameters like anchors and proposals during training and additional post-processing algorithms to reduce duplicate predictions [12].

2.3. Transformer-based approaches

Transformers have introduced paradigm shifts in object detection. Vision Transformers (ViTs) effectively processed entire images as sequences of patches, which enhanced global context modeling and motivated researchers to extend their use to more complex tasks such as object detection [31]. For example, Carion et al.[32] proposed Detection Transformer (DETR), an end-to-end object detector that directly predicted bounding boxes (BB) and classes via learned object queries. DETR did not require extensive manual tuning and was proven to handle varying object sizes and overlapping objects more effectively. Subsequent extensions, such as Deformable DETR [33], DN-DETR [34], and DAB-DETR [35], aimed to improve DETR's convergence and performance. While these extensions showed better detection performance, they still performed worse than the CNN counterparts [12]. The recent introduction of a collaborative hybrid assignments training scheme for DETR (Co-DETR) [22] addressed the issue of sparse supervision in DETR models by utilizing multiple auxiliary heads with one-to-many label assignments to enhance the learning of both the encoder and decoder. Co-DETR improved the training efficiency and discriminative feature learning of DETR-based detectors without adding any extra computational cost or parameters during inference. The experimental results

Table 1
Summary of recent fruit disease detection studies (2020–2025).

Author(s) & Year	Method	Dataset	Main findings	Limitations
Xie et al. (2020) [23]	Inception + SE-block	Grape leaf images (4449 images)	0.81 mAP at 15 FPS	Computationally intensive architecture
You et al. (2022) [24]	YOLO + Deep Metric Learning	Strawberry dataset (7230 images)	97.8 % overall accuracy	Complex architecture; lab-based dataset
Syed et al. (2022) [25]	Two-stage CNN	Citrus leaf images (598 images)	94.37 % accuracy	Limited generalizability
Huang et al. (2023) [26]	EfficientNet-Inception CNN + U-Net	Citrus dataset (800 images)	95.6 % classification accuracy; 87.7 % severity segmentation	Only 2 citrus diseases; small, lab-based dataset
Arifin et al. (2024) [27]	ResNet50 features + Logistic Regression	Citrus dataset (1814 images)	99.69 % accuracy	Small, imbalanced dataset; no lesion localization
Sun et al. (2024) [11]	YOLOv5 + shuffle-channel blocks	Natural orchard images (4252 images)	0.93 mAP	Manual hyperparameter tuning and post-processing needed
Aksoy et al. (2025) [28]	ResNet152V2 (transfer learning)	Kaggle apple fruit disease (502 images)	92 % classification accuracy	Small dataset (4 classes)
Faye et al. (2025) [29]	ResNet50 for severity grading	SenMangoFruitDDS (862 images)	97.8 % accuracy	Only on mango; limited background variability
He et al. (2025) [30]	Sparse Attention YOLOv11	Passion fruit dataset (10,000 annotated images)	90 % F1-score	Only passion fruit; stem-focused labels; high computational cost

Table 2
Descriptions of several widely used plant disease datasets. **Note:** # stands for the number of something.

Dataset	Year	Category	# Species	# Classes	# Images
PlantVillage [37]	2015	Classification	14	38	54,305
PlantDoc [36]	2020	Classification	13	27	2598
Citrus diseases [39]	2024	Classification	1	5	759
Pomegranate fruit diseases [38]	2024	Classification	1	5	5099
Fruit disease dataset [40]	2024	Detection	8	6	81,000

demonstrated a significant performance gain on various DETR variants. The integration of Co-DETR into DINO-Deformable-DETR achieved 66.0 % AP on the Common Objects in Context (COCO) test-development set.

3. Materials

Table 2 highlights the evolution of benchmark datasets in plant disease research. Earlier datasets, such as PlantDoc [36] and PlantVillage [37], included diseases affecting both fruits and leaves on multiple species but did not specifically focus on fruit diseases. In contrast, smaller self-collected datasets, such as the Pomegranate Fruit Diseases [38] and Citrus Diseases [39], primarily focus on diseases of single fruit types and contain fewer than 3000 images, which limits their scalability and generalizability.

This research stands out by training the proposed model on a large fruit disease identification dataset containing roughly 81,000 images that cover six different fruit disease types [40]. Provided by the National Information Society Agency of Korea (NIA),[1] this extensive dataset exceeds the scope and size of most existing datasets. The collection of data was made possible through the collaboration of Jeju Special Self-Governing Province,[2] with additional support from Flexink[3] and Bgrinfo[4] for data acquisition, and GDS Consulting[5] for data refinement and processing. The scale and diversity of this dataset significantly contribute to the strength and practical relevance of this study.

For details on the data collection process, including camera settings and acquisition methods, please refer to [40]. Fig. 1 presents representative images from each class of the fruit disease dataset on eight different plant species, including banana, fig, lemon, mango, mandarin, olive, passion fruit, and pitaya.

Fruits displaying signs of disease, such as spots, lesions, or other visible deformities, are visually inspected in both natural environments like orchards and controlled settings such as research greenhouses. Annotations are made at the lesion or affected region level. Each symptom is evaluated using specific attributes, including texture, spread, and severity, to ensure accurate labeling. Annotation guidelines follow established diagnostic criteria specific to each disease, as outlined below.

- Anthracnose (*Colletotrichum spp.*): Anthracnose affects a wide variety of plants, including pitaya, passion fruit, and olive [41]. Anthracnose typically presents small, sunken, dark brown to black lesions on the fruit's skin. These lesions may extend and eventually lead to significant areas of rot. The disease can cause premature fruit drop, leaf loss, and a significant reduction in overall fruit yield.
- Bacterial fruit blotch (*Acidovorax citrulli*): A serious disease caused by the bacterium *Acidovorax citrulli* [42]. The disease typically manifests as dark, water-soaked lesions on the fruit's surface. These lesions often start small but can rapidly expand to cover large portions of the fruit. As the disease progresses, the affected areas may crack and release a sticky, amber-colored bacterial exudate. The lesions can combine and lead to large, irregular blotches that severely affect the fruit's appearance and marketability. In severe cases, the entire fruit may become soft and rot.
- Broad mite (*Polyphagotarsonemus latus*): Broad mite is a tiny pest that can cause significant damage to various plants. The mites can infest young lemon fruits [43] and cause russeting or scars on the fruit surface. The affected fruits may become deformed and dropped prematurely in extreme cases.
- Weevil (*Curculionoidea*): Weevils [44] are small beetles that can cause significant damage to a variety of plants, including fig. Some weevil species burrow into the fruit and cause internal damage that may not be immediately visible from the outside. As a consequence, weevil infestation can lead to premature fruit drop, and the affected

[1] https://www.nia.or.kr/site/nia_kor/main.do
[2] https://www.jeju.go.kr/index.htm
[3] https://flexink.com/en/home/home-en/
[4] http://www.bgrinfo.co.kr/
[5] http://gdsconsulting.co.kr/

Fig. 1. Depiction of the six classes of fruit diseases from the dataset used in this study, with the affected regions highlighted by red BB.

fruits may become contracted. The entry points created by weevils can also serve as gateways for secondary infections by fungi or bacteria, which can further degrade the fruit's quality.

- Thrips (*Thysanoptera*): Thrips feed by piercing the surface of plant tissues and sucking out the contents of the cells [45], which leads to a range of symptoms that can seriously affect the health and yield of the plants. The most common symptom is surface scars, which affect the quality of the fruits.
- Fungal infection: Fungal infections can significantly impact the quality, marketability, and production of fruits such as bananas, lemon, mango, and fig [46]. Each type of fruit can be affected by specific fungal pathogens, which lead to distinct symptoms and potential economic losses. For example, black mildew forms a thin, black layer that can cover significant portions of the fruit's surface, such as lemon and mango. Although the fungus does not penetrate the fruit, it can lead to an unsightly appearance on the affected fruits. Powdery mildew can appear as a white to greyish powdery growth on the skin of figs. This fungal layer can lead to a rough fruit's surface and cause the fruit to crack in severe cases.

The annotation process focused on capturing both the visual characteristics of lesions and any related symptoms or traits that could improve the disease detection performance of the models. A dedicated team of 15 experts from MKG Engineering and Construction (MKGENC) was tasked with a five-month image annotation assignment. Each person annotated approximately 55 images per day to ensure that various disease symptoms were labeled precisely. An open-source annotation tool developed in Python was used to facilitate the entire annotation process [47]. Fig. 2 provides an overview of the dataset by showing the number of images for each disease class. It includes a total of 81,000 labeled images, which were split into 80 % for training, 10 % for validation, and 15 % for testing. Therefore, 64,800 images were used for training, while 8100 images were designated for both validation and testing.

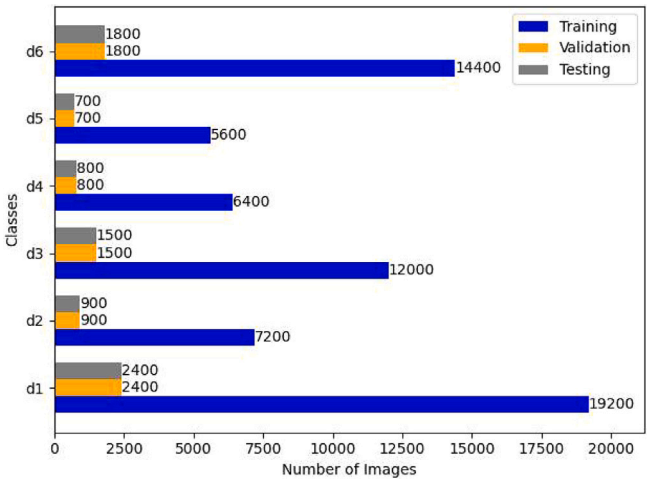


Fig. 2. A horizontal bar chart revealing the distribution of images per each disease class from d1–d6.

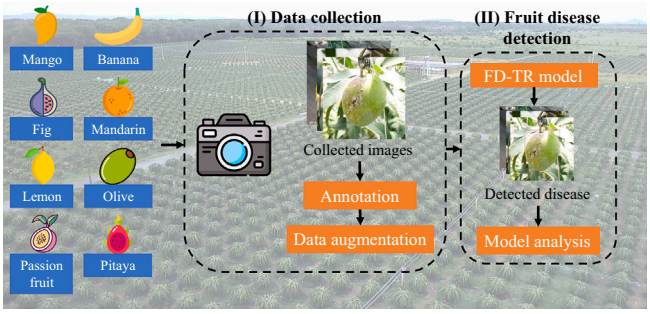


Fig. 3. Description of the primary processes of the proposed fruit disease detection framework (FD-TR).

4. Methods

4.1. System overview

Fig. 3 illustrates the primary steps of the fruit disease detection framework, referred to as FD-TR. In this framework, "FD" represents fruit disease detection, while "TR" refers to the transformer-based model. The two core components of the framework are outlined as follows.

- Data pre-processing: Real-world data often presents significant variability due to factors such as inconsistent lighting (e.g., shade, overexposure, underexposure), blurriness (caused by camera motion or low-quality optics), diverse angles (e.g., oblique views, close-ups), and noise (introduced by sensor imperfections or compression

Fig. 4. Output images of applying predefined data augmentation techniques on the original dataset.

artifacts). Therefore, data augmentation is essential to improve the model's robustness against these real-world challenges and its ability to generalize to unseen data [48]. The data augmentation technique involves artificially replicating these conditions within the dataset to effectively increase its size and diversity.

- Fruit disease detection: While existing object detection models like Mask-RCNN [49], YOLO [50], and SSD [51] achieve strong performance on benchmarks such as COCO [52] and Pascal VOC [53], they rely on manual hyperparameter tuning and multi-stage training. To address these limitations, we propose FD-TR, a transformer-based architecture with efficient end-to-end training. FD-TR focuses on specific parts of the input image most relevant for identifying diseases. Moreover, feature discriminability score analysis provides insights into the model's decision-making process for practical applications [13].

4.2. Data augmentation

This section outlines the image augmentation process applied to the fruit disease training dataset to improve the model's robustness and generalization by simulating various real-world conditions. These augmentation methods were performed on the original training set to better represent the variability encountered in real-world agricultural settings. The augmentation techniques expanded the original training set of 64,800 images fivefold to 324,000 images.

This process involved a series of transformations applied to the original images, including random horizontal and vertical flips to replicate different orientations of fruits on trees, and rotations at angles of 90°, 180°, and 270° to enhance the model's invariance to fruit positioning. In addition, color jittering, where the brightness, contrast, saturation, and hue of input images were randomly adjusted within predefined ranges was applied to mimic varying lighting conditions and potential color distortions caused by natural environments. To increase the model's robustness against the effects of camera noise and environmental factors, Gaussian noise was introduced to the images. Furthermore, random cropping and resizing were performed to expose the model to fruits at different scales and viewpoints. Fig. 4 provides a visual representation of the sampled augmented images obtained through different augmentation techniques.

4.3. Co-DETR framework

Co-DETR introduces a novel collaborative hybrid assignment training scheme designed to enhance the efficiency and effectiveness of DETR-based detectors. This scheme relies on versatile label assignment strategies to significantly boost the encoder's learning capabilities in end-to-end detection frameworks [22]. Co-DETR also optimizes the encoder's learning process by training multiple parallel auxiliary heads with one-to-many label assignments. In addition, Co-DETR improves the overall detection performance by optimizing the attention learning of the decoder through customized positive queries derived from the positive coordinates identified by the auxiliary heads. Fig. 5 illustrates the Co-DETR model, which includes three primary modules: a backbone, a transformer encoder, and a decoder.

According to the standard DETR protocol, the input image is fed into the backbone and encoder to extract latent features. Several predefined object queries subsequently interact with the decoder through cross-attention mechanisms. Co-DETR improves this process by integrating a collaborative hybrid assignment learning and a custom positive query generation module, which optimizes feature learning in the encoder and attention learning in the decoder.

4.3.1. Collaborative hybrid assignments training

To address the insufficient supervision of encoder outputs caused by the limited positive queries in the decoder of standard DETR architectures, Co-DETR integrates multiple label assignment strategies (e.g., Adaptive Training Sample Selection (ATSS), Faster R-CNN) with auxiliary supervision heads. These auxiliary heads strengthen encoder supervision by refining discriminative learning. Specifically, after processing the latent features $\mathcal{F}$, the encoder transforms them into a feature pyramid $\mathcal{F}_1, \ldots, \mathcal{F}_J$ via a multi-scale adapter, where J denotes the number of feature maps with downsampling stride of 2^{2+J}. Following the ViTDet framework, Co-DETR constructs its feature pyramid using a single-scale encoder feature map, which is upsampled using bilinear interpolation.

For example, the feature pyramid is built by sequentially applying upsampling (stride 2 with 3×3 convolution) or downsampling to the encoder's single-scale feature. In multi-scale encoders, only the coarsest resolution features are downsampled to generate the feature pyramid. For each K collaborative heads, the predicted output $\hat{P}_i$ is sequentially propagated through the feature pyramid $\mathcal{F}_1, \ldots, \mathcal{F}_J$. Within the i-th head, module A_i computes supervised targets for positive and negative samples, P_i^{pos}, B_i^{pos}, P_i^{neg}, using the supervised target set G, as follows:

$$P_i^{\{\text{pos}\}}, B_i^{\{\text{pos}\}}, P_i^{\{\text{neg}\}} = A_i(\hat{P}_i, G) \tag{1}$$

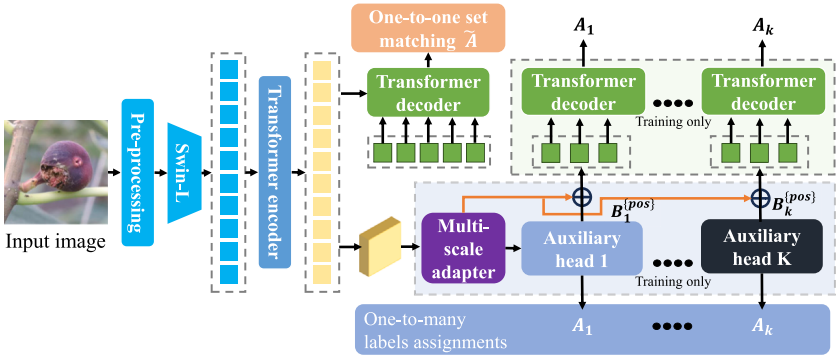


Fig. 5. Illustration of the architecture of the Co-DETR approach.

Table 3
Comprehensive explanation of the model fine-tuning process.

Model	Auxiliary	Loss	Optimizer
Deformable DETR	N/a	Hybrid (L1 + GIoU)	AdamW
DINO	N/a	Hybrid (L1 + GIoU)	AdamW
FD-TR (This study)	BatchFormerV2 [56]	Hybrid (L1 + CIoU)	LAMB

where pos and neg represent the spatial coordinates classified as positive and negative by A_i. The index j corresponds to the feature index within the feature pyramid $\mathcal{F}_j$. B_i^{pos} denotes the spatial coordinates of the positive samples, while P_i^{pos} and P_i^{neg} refer to the supervised targets associated with these coordinates, including both category labels and BB regression offsets.

The encoder loss function can be defined as follows:

$$\mathcal{L}_i^{\text{enc}} = \mathcal{L}_i(\hat{P}_i^{\{\text{pos}\}}, P_i^{\{\text{pos}\}}) + \mathcal{L}_i(\hat{P}_i^{\{\text{neg}\}}, P_i^{\{\text{neg}\}}) \tag{2}$$

For negative samples, the regression loss is excluded from consideration. The objective of optimization for the K auxiliary heads is therefore defined as follows:

$$\mathcal{L}^{\text{enc}} = \sum_{i=1}^{K} \mathcal{L}_i^{\text{enc}} \tag{3}$$

4.3.2. Customized positive queries generation

In the one-to-one matching paradigm, each ground-truth box is paired with a single specific query as its supervised target. However, when the number of positive queries is insufficient, this can lead to inefficient cross-attention learning within the transformer decoder. To address this issue, Co-DETR generates a diverse set of customized positive queries. Specifically, in the i-th auxiliary head, the customized positive query $Q_i \in \mathbb{R}^{M_i \times C}$ (where M_i represents the number of positive samples) is generated through the following process:

$$Q_i = \text{Linear}(\text{PE}(B_i^{\{\text{pos}\}})) + \text{Linear}(\mathbb{E}(\{\mathcal{F}_*\}, \{\text{pos}\})) \tag{4}$$

Here, $\text{PE}(\cdot)$ represents positional encoding, which extracts the relevant feature from $\mathbb{E}(\cdot)$ based on the spatial positive and negative coordinates $(j, \mathcal{F}_j)$.

Therefore, there are $K+1$ query groups involved in the one-to-one matching process, including those with label assignments. The auxiliary label assignment shares weights with the standard L decoder layers. In the auxiliary branches, all queries are conditioned on the positive query, eliminating the need for redundant matching. The loss for the l-th decoder layer in the i-th auxiliary head is formalized as follows:

$$\mathcal{L}_{i,l}^{\text{dec}} = \tilde{\mathcal{L}}(\tilde{P}_{i,l}, P_i^{\{\text{pos}\}}) \tag{5}$$

where $\tilde{\mathcal{L}}^{\text{dec}}$ denotes the loss from the original one-to-one matching branch. Finally, the global objective function of Co-DETR is defined as:

$$\mathcal{L}^{\text{global}} = \sum_{l=1}^{L} (\tilde{\mathcal{L}}_l^{\text{dec}} + \lambda_1 \sum_{i=1}^{K} \mathcal{L}_{i,l}^{\text{dec}} + \lambda_2 \mathcal{L}^{\text{enc}}) \tag{6}$$

Here, λ_1 and λ_2 are the coefficients that balance the different losses.

4.4. Model customization

Although Co-DETR can be applied to state-of-the-art transformer architectures such as DETR with Improved deNoising Anchor Box (DINO) [54] and Deformable DETR [33] for fruit disease detection, the performance of these base models remains sensitive to critical factors like label assignment strategies, robustness to complex backgrounds, and adaptability under varying environmental conditions. To optimize transformer-based detection for fruit disease detection, several targeted adjustments were introduced to the original transformer models' architecture and optimization process. These modifications were implemented before applying the Co-DETR approach, as outlined in Table 4.

Table 4
Ablation study comparing the dynamic α and fixed candidates $\{0.25, 0.5, 1.0\}$.

α Values	Mean mAP
Dynamic	0.77
0.25	0.68
0.5	0.75
1	0.7

The modifications include integrating BatchFormerV2 to enhance feature representation through batch-based learning, adopting the LAMB optimizer, known for its efficiency in training large-scale models [55], and utilizing the Complete Intersection over Union (CIoU) loss function instead of the GIoU to improve localization accuracy. These modifications are expected to improve the baseline models' performance and generalization capabilities in the fruit disease domain (Table 3).

- BatchFormerV2 (BF): Proposed by Hou et al. [56], BF enhances transformers' capacity to model inter-sample relationships within mini batches. Unlike conventional transformer blocks that operate on pixel- or patch-level feature maps, BF processes feature structured by batch size. In FD-TR framework, BF implements a two-stream architecture where both branches share weights and merge into a unified transformer decoder. This design ensures efficiency and coherence during the training process as all shared blocks are consistently trained with the same weights. Moreover, the original transformer blocks retain their full functionality without BF, which minimizes any additional computing during inference. The application of BatchFormerV2 into various transformer models, such as DETR [32] and Deformable-DETR [33], consistently demonstrated a performance improvement of over 1.3 mAP on the benchmark MS COCO dataset.
- Complete Intersection over Union (CIoU): The Generalized IoU (GIoU) extends the standard IoU metric by measuring the overlap between the predicted and ground truth BB while considering areas outside their intersection [57]. CIoU improves GIoU by introducing additional terms that account for localization precision and aspect ratio alignment. This refinement enables better convergence and improved detection accuracy compared to GIoU loss. Therefore, CIoU and L1 loss are utilized to calculate the box regression reconstruction loss for FD-TR model in this study.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{d^2\left(p, p^{gt}\right)}{c^2} + \alpha V \tag{7}$$

The variable c denotes the diagonal length of the smallest enclosing box that covers both the predicted and ground truth BB, while d represents the Euclidean distance between their center points. p and p^{gt} refer to the central points of the predicted and ground truth BB, respectively. The variable V measures the consistency of the aspect ratios, and α serves as a trade-off parameter that assigns less weight when the overlap is low and more weight when the overlap is high. The value of α is computed dynamically as:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \qquad \alpha = \frac{v}{(1-\text{IoU}) + v}, \tag{8}$$

We compared a dynamically computed α with fixed values $\alpha \in \{0.25, 0.5, 1.0\}$ on the validation set (Table 4). The dynamic α showed the highest peak validation mAP (0.77), but $\alpha = 0.5$ achieved a comparable validation mAP (0.75). To improve reproducibility and make cross-experiment comparisons more straightforward, we therefore use $\alpha = 0.5$ in all subsequent experiments. Moreover, fixed α also reduces hyperparameter tuning. If the aim is to maximize single-run peak mAP, dynamic α remains an appropriate choice.

- LAMB optimizer: While AdamW is commonly considered the default optimizer for a variety of vision transformer-based models [12,58] have identified potential training instability, particularly when there is an increased ratio between the L2-norm of weights and gradients. To mitigate this issue, this study adopts the Layer-wise Adaptive Large Batch Optimization (LAMB) optimizer as an alternative. LAMB combines the strengths of both the Adam and Layer-wise Adaptive Rate Scaling (LARS) optimizers [55]. In particular, the layer-wise adaptive technique from LAMB normalizes each dimension based on the square root of the second moment, while also applying layer-wise normalization. This method has been proved to be effective for distributed training and has demonstrated effectiveness in transformer models on large-scale datasets.

$$
\begin{aligned}
m_t &= \beta_1 m_t^{(\text{prev})} + (1-\beta_1)\, g_t \\
v_t &= \beta_2 v_t^{(\text{prev})} + (1-\beta_2)\, g_t^2 \\
m_t &= \frac{m_t}{1-(\beta_1)^t} \\
v_t &= \frac{v_t}{1-(\beta_2)^t} \\
r_t &= \frac{m_t}{\sqrt{v_t}+\epsilon} \\
x_{t+1}^{(i)} &= x_t^{(i)} - \eta_t \frac{\phi\left(\left\|x_t^{(i)}\right\|\right)}{\left\|r_t^{(i)}+\lambda x_t^{(i)}\right\|}\left(r_t^{(i)}+\lambda x_t^{(i)}\right)
\end{aligned}
\tag{9}
$$

where the hyperparameters β_1 and β_2 regulate momentum and weight decay, respectively. m_t refers to the first moment estimate at time step t, and v_t indicates the second moment estimate. The parameter λ manages the degree of layer-wise adaptiveness, while η_t represents the learning rate vector at time t, and ϕ denotes the parameter vector at the same instance. A small constant ϵ is introduced to prevent division by zero. In addition, r_t represents the update ratio used in the LAMB optimizer.

4.5. Feature discriminability scores analysis

After training, feature discriminability maps are generated by analyzing the multi-scale feature outputs from the DETR-based model [32]. They offer valuable insights into how the model distributes its focus on different regions of the input image. The feature discriminability scores are obtained by extracting multi-scale features from the model's final layers. For each feature map, the L2-norm is computed across the channel dimension to quantify the activation strength at each spatial location, consistent with established visualization practices for CNN activations [59]. The resulting feature discriminability scores are then normalized by their maximum values to ensure consistent intensity of different scales.

To visualize the feature discriminability scores, each normalized feature map is resized to match the dimensions of the input image using linear interpolation. The resized maps from each scale are then aggregated by combining them together, followed by averaging to produce a final feature map that integrates information from all scales. This final map highlights the regions that the model considers most relevant during the prediction process, with higher values indicating areas of greater focus. The output feature discriminability map is a valuable tool for evaluating the model's interpretability and its ability to correctly identify disease-affected regions in the image.

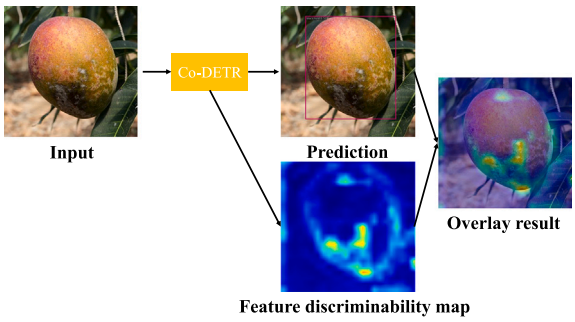


Fig. 6. Visualization of the feature discriminability map prediction process.

Let feats be a list of multi-scale feature maps, each with dimensions $L \times B \times C \times H \times W$. L is the number of layers or scales, $B = 1$ is the batch size, C is the number of channels, and $H \times W$ are the spatial dimensions. Based on the multi-scale feature maps, the feature discriminability map attn_map can be mathematically represented as follows:

$$
\text{attn\_map} = \frac{1}{L}\sum_{i=1}^{L} \text{resize}\left(\frac{\|\text{feat}[i]\|_2}{\max\left(\|\text{feat}[i]\|_2 + \varepsilon\right)}, H_{\text{img}}, W_{\text{img}}\right) \tag{10}
$$

where $\|\text{feat}[i]\|_2$ represents the L2-norm of the feature map at the i-th scale, calculated along the channel dimension for generating a feature map of size $H \times W$. The term $\max\left(\|\text{feat}[i]\|_2\right)$ denotes the maximum value in the normed feature map, which is used to normalize the map. The function $\text{resize}(\cdot, H_{\text{img}}, W_{\text{img}})$ interpolates the normalized feature map to match the dimensions $H_{\text{img}} \times W_{\text{img}}$ of the input image. The summation aggregates the resized feature discriminability maps from all scales, and the division by L averages the aggregated map.

In Fig. 6, the feature discriminability map extraction of an input image highlights how FD-TR effectively focuses on disease-affected regions using multi-scale features from the encoder. The map illustrates the DETR-based model's ability to precisely target the main regions showing disease symptoms. This demonstrates the model's robustness and accuracy in detecting various fruit diseases.

4.6. Implementation details

The fruit disease detection framework was developed using the MMDetection library v2.25.3, built on PyTorch 1.11.0. To ensure consistent and fair experimentation, all detection models in the study utilized ResNet-50 and Swin backbone pre-trained on the ImageNet dataset. The training process was conducted on an Nvidia A100 GPU with 40 GB of memory.

We integrate our Co-DETR into existing DETR-like pipelines while maintaining similar training settings to the baseline models. For $K = 2$, we implement both ATSS and Faster-RCNN as auxiliary heads, whereas for $K = 1$, we use only the ATSS head. In addition, the number of learnable object queries is set to 300, and the weight coefficients $\{\lambda_1, \lambda_2\}$ are set to their default values of {1.0, 2.0}.

For all transformer-based experiments (FD-TR and DETR variants), each model is trained for up to 15 epochs with validation process performed at the end of each epoch. Early stopping is applied to the validation bounding-box loss with a patience of three epochs and a minimum improvement threshold $\Delta = 10^{-3}$. If the bounding-box loss fails to decrease by at least Δ for three consecutive epochs, training halts and the model reverts to the weights from the epoch with the lowest validation loss.

4.7. Evaluation protocols

In this section, we comprehensively evaluate the fruit disease recognition framework using several standard metrics, including mAP, precision, and recall. These metrics are computed based on the three elements of the confusion matrix: true positive (TP), false positive (FP), and false negative (FN). Precision reveals the ratio of correctly predicted positive instances out of all predicted positives, while recall captures the proportion of true positives among all actual positives in the dataset. The formulations of these metrics are as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \end{aligned} \tag{11}$$

To evaluate the overall detection accuracy of multiple disease classes, a standard average precision metric was calculated. In particular, we adopt $AP@[IoU = 0.50 : 0.95]$, which measures the detection performance at IoU thresholds from 0.50 to 0.95. This threshold is used to evaluate the model's ability to localize fruit diseases by calculating the area under the precision-recall curve at the specified IoU threshold. The AP for each class is determined from this curve, and the mAP is then computed as the average of the AP values on all disease types. The mAP is expressed as follows:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{12}$$

where N represents the number of disease types, and AP_i denotes the average precision for the i-th disease class. AP_i is calculated based on the precision-recall curve for that disease type.

5. Results

5.1. Comparison of transformer models

In this experiment, a comprehensive comparison of fruit disease detection performance is conducted by applying Co-DETR on various DETR-based models, including Deformable DETR [33] and DINO [54]. Moreover, two different backbones, Swin Transformer and ResNet-50, are employed and compared, resulting in a total of four model variants. The models include Co-DETR on the Deformable DETR with the ResNet-50 backbone (co_deformable_detr_r50), Co-DETR on the Deformable DETR with the Swin backbone (co_deformable_detr_swin), Co-DETR on the DINO model with the ResNet-50 backbone using 5-scale feature processing (co_dino_5scale_r50), and Co-DETR on the DINO model with the Swin backbone using 5-scale feature processing (co_dino_5scale_swin). The performance comparison is shown in Fig. 7.

Overall, the co_dino_5scale_swin model demonstrates the highest performance with a detection mAP starting at around 0.6 and steadily improving to around 0.81 by the 12th epoch. This indicates that the Swin backbone combined with 5-scale feature extraction is particularly effective in detecting fruit diseases. The co_dino_5scale_detr model also performs well, closely following co_dino_5scale_r50 while maintaining a high performance at around 0.79 at the 12th epoch. The co_deformable_detr_r50 model shows relatively stable performance but with lower performance compared to the DINO models. In contrast, the co_deformable_detr_swin model exhibits significant fluctuations in its performance, particularly between epochs 5 and 7, where it experiences a sharp drop to around 0.25 mAP. However, the model recovers rapidly from epoch 8th and reaches a comparable mAP of approximately 0.72 by the 12th epoch. These fluctuations suggest that while the Deformable DETR architecture may be more sensitive to certain training conditions, it is capable of eventually reaching a competitive performance.

Given that Co-DETR on the DINO model with the Swin backbone demonstrated the highest fruit disease detection performance, we selected this configuration as the default model for subsequent experiments (referred to as FD-TR). This extension was chosen because it delivered robust and stable detection accuracy during training and validation. FD-TR was then used to evaluate the effects of additional enhancements, such as data augmentation techniques, hyperparameter tuning, and its deployment in real-world environments.

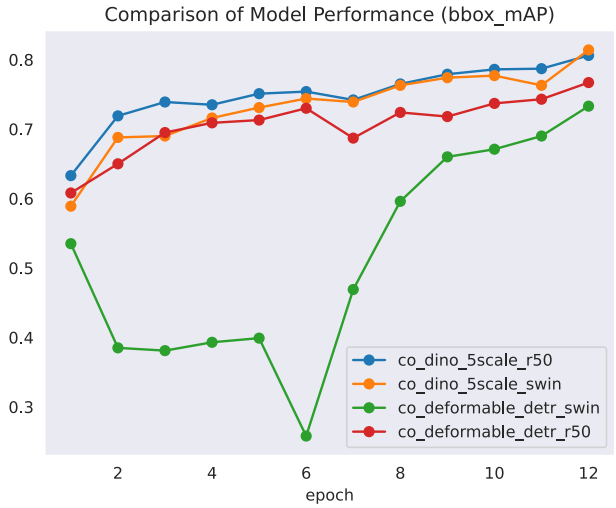


Fig. 7. Comparison of fruit disease detection performance using Co-DETR applied to two baseline DETR models: Deformable DETR and DINO.

Table 5
Comparison of FD-TR model performance on original and augmented data.

	mAP	Precision	Recall
Original data	0.76	0.75	0.78
Data augmentation	0.81	0.79	0.82

5.2. Preprocessing module analysis

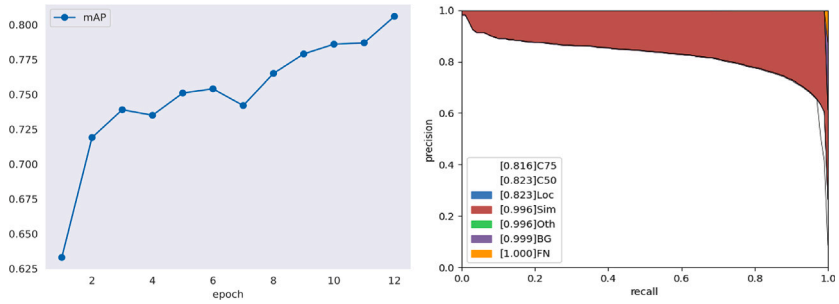
This section examines the impact of data augmentation on the proposed FD-TR model by comparing its results with the one trained on raw data. As shown in Table 5, FD-TR trained with augmented images outperformed the one trained on raw data. For example, the mAP increased by 0.05 from 0.76 to 0.81, indicating better overall accuracy in detection. The data augmentation approach also reduced the false positive detection (higher precision) and increased the rate of correctly identifying true positives (higher recall).

The observed performance improvement suggests that data augmentation plays a crucial role in boosting FD-TR model's ability to detect fruit diseases with higher detection accuracy. By introducing variations in the training data, augmentation not only boosts detection precision but also significantly improves the model's robustness.

5.3. PD-TR performance evaluation

Fig. 8 provides a detailed performance evaluation of FD-TR model, which consists of two charts.

- The evaluation mAP performance (a) plots FD-TR's mAP over 12 epochs of training. The mAP started at approximately 0.625 and steadily increased. It peaked at around 0.8 by the 12th epoch. This consistent improvement in mAP indicated that the model was learning effectively and becoming increasingly better at detecting diseases as training progressed. The gradual increase suggested that the model generalized well and converged to high performance, especially in the later epochs.



(a) Evaluation mAP performance **(b) Precision-Recall curve**

Fig. 8. Detailed performance evaluation of FD-TR model using different evaluation metrics.

Table 6
Evaluation results of the proposed model on different fruit disease classes.

	d1	d2	d3	d4	d5	d6	Average
mAP	0.78	0.74	0.88	0.83	0.77	0.86	0.81
Precision	0.76	0.73	0.85	0.8	0.77	0.84	0.79
Recall	0.79	0.77	0.89	0.82	0.79	0.88	0.82

- The precision-recall curve (b) represents the trade-off between precision and recall for different thresholds. This curve can be used to evaluate how well FD-TR performs at different confidence levels. Overall, the model accurately detected diseases with minimal false positives because the curve showed high precision for most recall values. Key metrics like C75, C50, and Loc revealed precise localization and detection capabilities, with precision values around 0.816 to 0.823, suggesting that the model performed well even under challenging IoU thresholds. The model also excelled in distinguishing between similar diseases (Sim) and avoiding background errors (BG), with a precision near 1.0 in both cases. The curve's slight decline at very high recall indicates that while the model maintained accuracy under most conditions, it introduced minor false positives when recall was pushed to its limit. Finally, a good false negative (FN) rate showed that the model had a very low rate of missing diseased fruits.

Table 6 describes the experimental results of FD-TR framework in detecting six different fruit diseases, including anthracnose (d1), bacterial fruit blotch (d2), broad mite (d3), weevil (d4), thrips (d5), and fungal infection (d6). In general, FD-TR framework showed consistent performance in detecting all disease classes with an average mAP of 0.81, precision of 0.79, and recall of 0.82. The model achieved the highest performance for detecting d3 and d6 with the mAP scores of 0.88 and 0.86, respectively. These classes also obtained strong precision (0.85 and 0.84) and recall (0.89 and 0.88). On the other hand, the detection performance for d2 and d5 was slightly lower, with mAP values of 0.74 and 0.77. The low detection performance of d2 and d5 could be due to several factors: 1) fewer labeled instances in the training data, which limited the framework's ability to extract distinct features for these diseases, and 2) visual similarities between d2 and d5 made it challenging for the model to effectively differentiate between these diseases and others.

5.4. Analysis of the feature discriminability analysis

Table 7 reports the mean and standard deviation of the normalized L_2-norm discriminability scores for each disease class over the test set. The scores confirm that the model focuses more strongly on classes with more distinct lesion features, such as anthracnose, broad mite, and fungal infection.

Table 7
Mean (± std) of feature discriminability scores per disease class.

Disease class	Mean (± std) score
(d1) Anthracnose	0.86 ± 0.05
(d2) BFB	0.65 ± 0.10
(d3) Thrips	0.72 ± 0.03
(d4) Weevil	0.68 ± 0.08
(d5) Broad mite	0.80 ± 0.01
(d6) Fungal infection	0.83 ± 0.03

Fig. 9 provides a detailed description of the proposed framework for effectively detecting six distinct fruit diseases. Each row in the figure serves a distinct purpose. Row (a) displays the original images of fruits affected by diseases such as anthracnose, BFB, thrips, weevil, broad mite, and fungal infection. The second row (b) demonstrates the model's detection results by highlighting the areas where the model has identified disease presence with BB and predicted labels.

Overall, the model correctly predicted and localized the fruit diseases precisely. In order to explain the model's prediction process, the third row (c) further shows attention-weight visualizations from FD-TR model. The extracted attention map reveals where the model is focusing its attention on the images. Warmer color areas indicate higher focus, which is typically around spots showing visible symptoms of the disease. It can be concluded by observing the attention maps that the model focused on disease regions but also provided visual explanations for its predictions. Moreover, the attention analysis also enhanced trust and understanding in its diagnostic capabilities.

Fig. 10 demonstrates FD-TR model's performance on some challenging fruit disease detection cases, such as lighting variations, image blurring, and low contrast. The top row (a) displays the input images, while the second row (b) shows the detection results, including the predicted BB, disease name and confidence score. The attention map visualization in the bottom row (c) indicates how FD-TR model focuses on specific regions of the image for its predictions.

FD-TR model demonstrates strong disease prediction performance in real-world conditions. This is important for practical deployment in agricultural environments where image quality may vary. For instance, the model demonstrates its robustness by accurately detecting anthracnose (first column) and black mold (fourth column) with high confidence scores of 0.94 and 0.77, respectively. In these cases, the model focuses effectively on the infected areas with well-defined and concentrated regions in the attention maps.

In contrast, for more challenging cases such as weevil (second column) and broad mite (third column), the attention maps appear more diffuse, with less sharply defined focus areas. Factors such as image blurring and uneven lighting seem to affect the model's ability to identify the diseased regions accurately. This results in a lower confidence score for weevil detection (0.48), indicating the model's difficulty in isolating

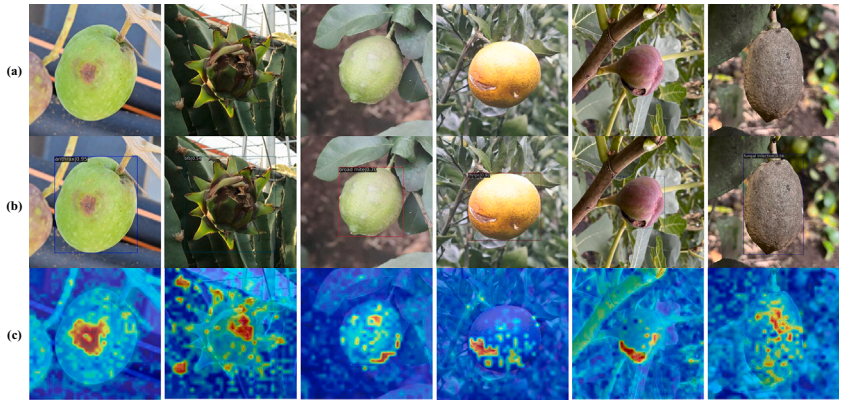


Fig. 9. The proposed model's outputs for each fruit disease, including (a) input images, (b) detection results, and (c) feature discriminability visualizations.

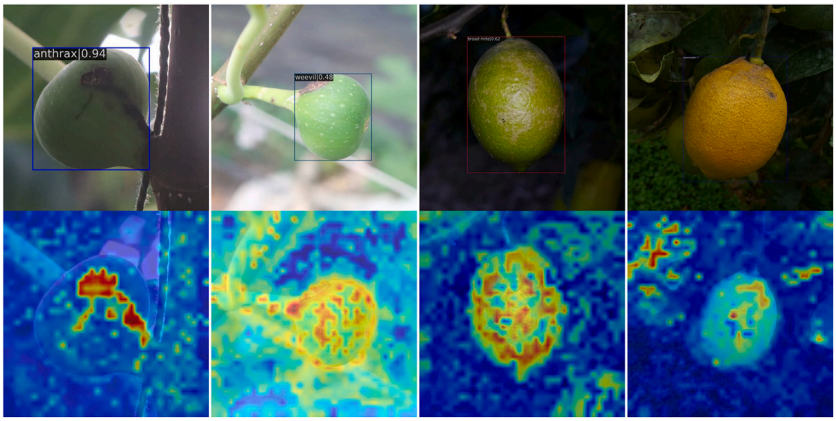


Fig. 10. The proposed model's outputs for challenging cases, including (a) input images, (b) detection results, and (c) feature discriminability visualizations.

Table 8
Ablation analysis for evaluating the effects of different components on the performance of FD-TR model.

Configuration	CIoU + L1 loss	LAMB optimizer	BatchFormerV2	mAP
Baseline	–	–	–	0.812
+ CIoU only	✓	–	–	0.847
+ LAMB only	–	✓	–	0.818
+ BatchFormerV2 only	–	–	✓	0.853
+ CIoU & LAMB	✓	✓	–	0.834
+ CIoU & BatchFormerV2	✓	–	✓	0.882
+ LAMB & BatchFormerV2	–	✓	✓	0.838
Full integration	✓	✓	✓	0.894

the specific features of the disease. Nevertheless, FD-TR model manages to generate reasonable predictions.

5.5. Analysis of the effectiveness of customized components to the performance of FD-TR model

This section reports the effectiveness of important components of the proposed fruit disease detection model's performance. Table 8 summarizes the ablation study's results of each component of FD-TR model.

The baseline configuration, without any of the proposed components, achieved an mAP of 0.812. When added individually, CIoU + L1 loss improved the mAP to 0.847, which demonstrated its significant contribution to the model performance. The LAMB optimizer showed a marginal improvement to 0.818, while BatchFormerV2 alone boosted the mAP to 0.853. Further analysis of pairwise combinations revealed additional insights. The combination of LAMB optimizer with CIoU + L1 loss or BatchFormerV2 yielded lower mAP compared to using CIoU + L1 loss or BatchFormerV2 alone. However, these configurations achieved an average of 13 % faster convergence and reduced training time. Meanwhile, the integration of both CIoU + L1 loss with BatchFormer V2 led to a substantial increase to 0.882, which suggested a stronger interaction between these two components. Finally, the full integration of all three components achieved the highest performance at 0.894, which highlighted their effectiveness in enhancing the model's capabilities.

Table 9
Model performance evaluation between the proposed model and five state-of-the-art DL models on the validation dataset.

Model name	mAP	Precision	Recall
SSD [51]	0.69	0.66	0.70
YOLOv8 [60]	0.8	0.81	0.83
DETR [32]	0.72	0.71	0.73
Deformable DETR [33]	0.74	0.74	0.77
DINO [54]	0.75	0.74	0.76
FD-TR (Ours)	0.89	0.86	0.87

5.6. Comparison with other models

Table 9 presents a performance comparison between the proposed FD-TR model and five other state-of-the-art detection models (YOLOv8 [60], SSD [51], DETR [32], Deformable DETR [33], DINO [54]). When evaluated on the validation dataset, FD-TR consistently outperformed the others on all metrics. Specifically, FD-TR significantly outperformed the next best model by 9 % with an mAP of 0.89. In addition, with high precision and recall values, FD-TR demonstrated its ability to accurately identify and localize objects. In contrast, SSD exhibited the lowest performance, with an mAP of 0.69, and a precision and recall values of 0.66 and 0.70, respectively.

Moreover, while other transformer-based models like DETR, Deformable DETR, and DINO demonstrated higher performance over SSD, they were consistently outperformed by FD-TR. For example, Deformable DETR showed an mAP of 0.74, precision of 0.74, and recall of 0.77, while DINO achieved slightly better precision and recall but a comparable mAP. YOLOv8, well-known for its performance, performed well with an mAP of 0.80 but was outperformed by FD-TR in all metrics. The results highlight that FD-TR model provides the most accurate and reliable predictions for fruit disease detection due to several enhancements such as the Co-DETR scheme and effective integration of other components.

5.7. Comparison on various benchmark datasets

Table 10 describes the performance of FD-TR on four publicly available datasets compared to the baseline model (Co-DETR). This table includes two agricultural datasets (PlantVillage [61] and Pest-D2Det [62]) and widely used general benchmarks (COCO [52] and VOC2012 [63]). The variation in domain complexity, class count, and dataset size provides a comprehensive evaluation of the model's adaptability.

In the agricultural domain, FD-TR demonstrates significant advancements, particularly on PlantVillage, where it achieves an mAP of 0.594, an 18.7 % gain over the YOLOv8 baseline (0.407). This improvement highlights FD-TR's effectiveness in handling high-class diversity (38 classes) and complex disease manifestations. Similarly, on Pest-D2Det, FD-TR obtains an mAP of 0.731, a 2.7 % increase over the D2Det baseline (0.704), which confirms its strength in pest detection tasks with fewer classes (10). These results indicate that FD-TR performs well in agricultural context, where precise feature learning and optimization are critical for real-world applications like crop monitoring.

For general-domain datasets, FD-TR exhibits robust but context-dependent performance. On VOC2012 (20 classes), it achieves an mAP of 0.812, a modest 0.8 % improvement over the baseline CoupleNet (0.804). However, on COCO (80 classes), FD-TR records an mAP of 0.589, approximately 7.0 % below Co-DETR's reported 0.659. This gap does not undermine FD-TR's efficiency but rather reflects key architectural and training differences. Co-DETR leverages a large ViT-Large backbone and extensive pre-training on Objects365 (optimized for large-scale benchmarks like COCO). In contrast, FD-TR prioritizes lightweight efficiency using Swin as backbone, and targets agricultural specialization without targeted pre-training. FD-TR's modifications (BatchFormerV2 for enhanced feature representation, CIoU for improved box learning, and LAMB for training stabilization) emphasize domain-specific adaptability over maximizing COCO accuracy. Despite the lower score, FD-TR remains competitive with many transformer-based detectors and aligns with its goal of balancing performance, efficiency, and specialization. Overall, these results confirm FD-TR's contributions, particularly in agricultural contexts, while maintaining versatility across domains.

5.8. Real-world robustness analysis

To evaluate the model's ability to distinguish healthy fruits, which is a critical requirement for real-world agricultural applications, an independent test dataset comprising 500 images of healthy fruits was collected. These images were curated from a publicly available agricultural image repository and verified by domain experts to confirm the absence of disease symptoms. This dataset was excluded from training and reserved solely for evaluating the model's performance in real-world scenarios. An image was classified as "healthy" if no disease-related BB were predicted. The model correctly identified 431 out of 500 healthy images, leading to a false positive rate of 13.8 %. This demonstrates that FD-TR can effectively differentiate healthy fruits from unhealthy ones in most cases. Fig. 11 highlights three failure modes where natural fruit features were mistakenly classified as disease symptoms. In these cases, the model misinterpreted natural variations in fruit appearance, such as blemishes, color gradients, or developmental traits, as pathological indicators:

- Case (A): A healed scar on a citrus fruit (red arrow) was misclassified as a fungal infection (confidence: 0.47). The model failed to distinguish the scar's shallow, textured appearance from active fungal lesions.
- Case (B): A young dragon fruit exhibiting natural tip browning (red arrow) was incorrectly flagged as infected with BFB, despite lacking characteristic water-soaked lesions.
- Case (C): A faint reddish patch on a young fig (red arrow) was predicted as a fungal spot, even though the coloration was uniform and confined to healthy epidermal tissue.

These examples revealed that the model's false positives occurred not from complex background clutter or extreme lighting artifacts, but from everyday morphological and variations traits of healthy fruits that were not included in the training set. Such improvements would enhance the model's robustness to real-world variability and reduce overfitting to disease-centric features.

6. Discussion

FD-TR model improves fruit disease detection by combining the Co-DETR training scheme with the DINO transformer model, multi-scale feature extraction, and attention mechanisms. Key model customization,

Table 10
FD-TR performance and gains compared to baseline methods. Note: pp stands for absolute gain in percentage points.

Dataset	Domain	# Classes	# Images	Baseline mAP	FD-TR mAP	pp (%)
PlantVillage	Agriculture	38	54,308	0.407 (YOLOv8 [64])	0.594	18.7
Pest-D2Det	Agriculture	10	9472	0.704 (D2Det [65])	0.731	2.7
COCO	General	80	118,287	0.659 (Co-DETR [22])	0.589	−7
VOC2012	General	20	11,540	0.804 (CoupleNet [66])	0.812	0.8

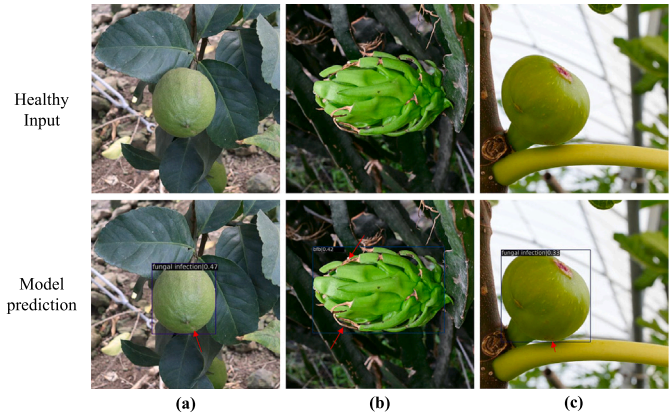


Fig. 11. Samples of false positive prediction by the model for healthy fruit images.

including CIoU loss for precise BB, the LAMB optimizer for faster convergence, and BatchFormerV2 for scalable training, enhanced detection performance and efficiency for six fruit disease classes. FD-TR's end-to-end design and integrated data augmentation improved robustness under diverse real-world scenarios, such as lighting and angles.

The experimental results showed that targeted customization improved detection mAP from 0.81 to 0.89. FD-TR also outperformed YOLOv8 (0.80) and Deformable DETR (0.74). With precision and recall rates of 0.86 and 0.87, respectively, it demonstrated robust generalization across diverse disease symptoms, scales, and environmental conditions. These capabilities are crucial for real-world agricultural settings, where early and accurate detection is crucial for effective intervention and crop protection. Furthermore, its attention-based interpretability via feature discriminability scores and deformable attention weights provided transparent insights into decision-making. The evaluation on healthy fruit images, as introduced in Section 5.8, demonstrated the model's potential to operate effectively in real-world settings where both diseased and healthy fruits are present. Although, a false positive rate of 13 % on healthy samples was promising, the misclassifications highlighted a limitation in the current training data, which lacked explicit healthy examples.

7. Conclusions

This research introduces an enhanced end-to-end transformer-based fruit disease recognition model that can be applied to real-life disease management systems. The dataset used to train the model consists of 81,000 images of six different fruit diseases. The proposed FD-TR model demonstrates high detection performance on the dataset compared to state-of-the-art models such as YOLOv8, DINO, and Deformable DETR. FD-TR is based on the DINO transformer model with an improved Co-DETR training scheme and additional components like CIoU loss, the LAMB optimizer, and BatchFormerV2. These improvements contribute to the model's enhanced detection capabilities and faster convergence during training. Therefore, FD-TR model not only improves the accuracy of predictions but also achieves robust performance in various experiments.

Moreover, FD-TR model's ability to maintain high performances on diverse testing scenarios demonstrates its generalization ability and reliability. Even in challenging cases, such as images affected by poor lighting or blurring, the model provides correct and robust predictions. The attention mechanism of the transformer allows the model to focus on relevant disease features, which reduces false predictions. In addition, the unique multi-scale attention map extracted from the transformer offers experts/farmers valuable insights into how the model detects and highlights disease-related areas. FD-TR model represents a significant advancement in automated disease detection and offers substantial potential to improve agricultural productivity and disease management in modern farming.

While FD-TR model demonstrates strong performance in detecting fruit diseases, several limitations persist. One of the main limitations is the reliance on a dataset with a limited number of disease classes, which fails to capture the full diversity of fruit diseases and environmental conditions. Moreover, the model's performance could be further optimized in challenging environmental conditions, where it occasionally struggles to detect diseases accurately. In the future, the dataset can be expanded to include more diverse conditions and disease types to improve the model's generalizability. In addition, techniques like multimodal data integration, which analyze data from sensors such as infrared cameras or spectroscopy, can be considered for further development and improvement. Finally, the model optimization on edge/mobile devices is a critical future work to enable real-time, on-field disease detection, especially in resource-constrained environments. This would involve exploring lightweight backbones and model compression techniques to reduce computational demands for edge devices.

CRediT authorship contribution statement

Yanfen Li: Writing – original draft. **Muhammad Fayaz:** Data curation. **Sufyan Danish:** Visualization. **Lilia Tightiz:** Investigation. **Hanxiang Wang:** Methodology, Conceptualization. **Tan N. Nguyen:** Validation, Supervision. **L. Minh Dang:** Writing – review & editing, Validation, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China (No. 62502271), the Young Scientists Fund of the Natural Science Foundation of Shandong Province, China (No. ZR2025QC630), the Natural Science Foundation of Rizhao City, China (Nos. RZ2024ZR33 and RZ2024ZR34).

Data availability

Data will be made available on request.

References

[1] FAO, How to feed the world in 2050 (2009). https://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf (2023-Jul-15).
[2] O. Turnbull, M. Homer, H. Ensaff, Food insecurity: its prevalence and relationship to fruit and vegetable consumption, J. Hum. Nutr. Diet. 34 (5) (2021) 849–857.
[3] M. Dang, H. Wang, Y. Li, T.-H. Nguyen, L. Tightiz, N. Xuan-Mung, T.N. Nguyen, Computer vision for plant disease recognition: a comprehensive review, Bot. Rev. 90 (3) (2024) 1–61.
[4] V. Singh, N. Sharma, S. Singh, A review of imaging techniques for plant disease detection, Artif. Intell. Agric. 4 (2020) 229–242.
[5] L.C. Ngugi, M. Abelwahab, M. Abo-Zahhad, Recent advances in image processing techniques for automated leaf pest and disease recognition–a review, Inf. Process. Agric. 8 (1) (2021) 27–51.
[6] S.A. Gaikwad, K.S. Deore, M.K. Waykar, P.R. Dudhane, G. Sorate, Fruit disease detection and classification, Int. Res. J. Eng. Technol. 4 (2017) 1151–1154.
[7] B.J. Samajpati, S.D. Degadwala, Hybrid approach for apple fruit diseases detection and classification using random forest classifier, in: 2016 International conference on communication and signal processing (ICCSP), IEEE, 2016, pp. 1015–1019.
[8] I.M. Nasir, A. Bibi, J.H. Shah, M.A. Khan, M. Sharif, K. Iqbal, Y. Nam, S. Kadry, Deep learning-based classification of fruit diseases: an application for precision agriculture, Comput. Mater. Contin. 66 (2) (2021) 1949–1962.
[9] N. Yao, F. Ni, M. Wu, H. Wang, G. Li, W.-K. Sung, Deep learning-based segmentation of peach diseases using convolutional neural network, Front. Plant Sci. 13 (2022) 876357.
[10] L.M. Dang, S.I. Hassan, I. Suhyeon, A.K. Sangaiah, I. Mehmood, S. Rho, S. Seo, H. Moon, Uav based wilt detection system via convolutional neural networks, Sustain. Comput. Inform. Syst. 28 (2020) 100250.

[11] H. Sun, J. Xue, Y. Song, P. Wang, Y. Wen, T. Zhang, Detection of fruit tree diseases in natural environments: a novel approach based on stereo camera and deep learning, Eng. Appl. Artif. Intell. 137 (2024) 109148.
[12] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan, M. Shah, Transformers in vision: a survey, ACM Comput. Surv. 54 (10s) (2022) 1–41.
[13] A. Vaswani, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017).
[14] C. Zhu, W. Ping, C. Xiao, M. Shoeybi, T. Goldstein, A. Anandkumar, B. Catanzaro, Long-short transformer: efficient transformers for language and vision, Adv. Neural Inf. Process. Syst. 34 (2021) 17723–17736.
[15] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: the efficient transformer, in: International Conference on Learning Representations, 2020.
[16] C. Wang, Y. Chen, S. Zhang, Q. Zhang, Stock market index prediction using deep transformer model, Expert Syst. Appl. 208 (2022) 118128.
[17] T. Kehinde, O.J. Adedokun, A. Joseph, K.M. Kabirat, H.A. Akano, O.A. Olanrewaju, Helformer: an attention-based deep learning model for cryptocurrency price forecasting, J. Big Data 12 (1) (2025) 81.
[18] H. Chen, Z. Qi, Z. Shi, Remote sensing image change detection with transformers, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–14.
[19] M. Orabi, K.P. Tran, P. Egger, S. Thomassey, Anomaly detection in smart manufacturing: an adaptive adversarial transformer-based model, J. Manuf. Syst. 77 (2024) 591–611.
[20] K.M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J.Q. Davis, A. Mohiuddin, L. Kaiser, et al., Rethinking attention with performers, in: International Conference on Learning Representations, 2020.
[21] P.S. Thakur, P. Khanna, T. Sheorey, A. Ojha, Vision transformer for plant disease detection: Plantvit, in: International Conference on Computer Vision and Image Processing, Springer, 2021, pp. 501–511.
[22] Z. Zong, G. Song, Y. Liu, Detrs with collaborative hybrid assignments training, in: Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 6748–6758.
[23] X. Xie, Y. Ma, B. Liu, J. He, S. Li, H. Wang, A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks, Front. Plant Sci. 11 (2020) 751.
[24] J. You, K. Jiang, J. Lee, Deep metric learning-based strawberry disease detection with unknowns, Front. Plant Sci. 13 (2022) 891785.
[25] S.F. Syed-Ab-Rahman, M.H. Hesamian, M. Prasad, Citrus disease detection and classification using end-to-end anchor-based deep learning model, Applied Intell. 52 (1) (2022) 927–938.
[26] Z. Huang, X. Jiang, S. Huang, S. Qin, S. Yang, An efficient convolutional neural network-based diagnosis system for citrus fruit diseases, Front. Genet. 14 (2023) 1253934.
[27] K.N. Arifin, S.A. Rupa, M.M. Anwar, I. Jahan, Lemon and orange disease classification using cnn-extracted features and machine learning classifier, in: Proceedings of the 3rd International Conference on Computing Advancements, 2024, pp. 154–161.
[28] S. Aksoy, P. Demircioglu, I. Bogrekci, Web-based ai system for detecting apple leaf and fruit diseases, AgriEngineering 7 (3) (2025) 51.
[29] D. Faye, I. Diop, N. Mbaye, D. Dione, M.M. Diedhiou, Mango fruit diseases severity estimation based on image segmentation and deep learning, Discov. Appl. Sci. 7 (2) (2025) 1–12.
[30] Y. He, N. Zhang, X. Ge, S. Li, L. Yang, M. Kong, Y. Guo, C. Lv, Passion fruit disease detection using sparse parallel attention mechanism and optical sensing, Agriculture 15 (7) (2025) 733.
[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.
[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European conference on computer vision, Springer, 2020, pp. 213–229.
[33] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: deformable transformers for end-to-end object detection, in: International Conference on Learning Representations, 2020.
[34] F. Li, H. Zhang, S. Liu, J. Guo, L.M. Ni, L. Zhang, Dn-detr: accelerate detr training by introducing query denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13619–13627.
[35] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, L. Zhang, Dab-detr: dynamic anchor boxes are better queries for detr, in: International Conference on Learning Representations, 2022.
[36] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, N. Batra, Plantdoc: a dataset for visual plant disease detection, in: Proceedings of the 7th ACM IKDD cods and 25th COMAD, Association for Computing Machinery, 2020, pp. 249–253.
[37] D. Hughes, M. Salathé, et al., An open access repository of images on plant health to enable the development of mobile disease diagnostics, arXiv preprint arXiv:1511.08060 2015.
[38] B. Pakruddin, R. Hemavathy, A comprehensive standardized dataset of numerous pomegranate fruit diseases for deep learning, Data In Brief 54 (2024) 110284.
[39] H.T. Rauf, B.A. Saleem, M.I.U. Lali, M.A. Khan, M. Sharif, S.A.C. Bukhari, A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning, Data Brief 26 (2019) 104340.
[40] NIA, Ai hub dataset (2023). https://www.aihub.or.kr (2023-May-11).
[41] A. Ciofini, F. Negrini, R. Baroncelli, E. Baraldi, Management of post-harvest anthracnose: current approaches and future perspectives, Plants 11 (14) (2022) 1856.
[42] J. Daley, T.C. Wehner, Screening for bacterial fruit blotch resistance in watermelon fruit, Crop Sci. 61 (2) (2021) 1228–1240.
[43] M. Cabedo-López, J. Cruz-Miralles, D. Peris, M.V. Ibáñez-Gual, V. Flors, J.A. Jaques, The response of citrus plants to the broad mite polyphagotarsonemus latus (banks)(acari: tarsonemidae), Agric. For. Entomol. 23 (4) (2021) 411–419.
[44] J. Haran, G.J. Kergoat, B.A. de Medeiros, Most diverse, most neglected: weevils (coleoptera: curculionoidea) are ubiquitous specialized brood-site pollinators of tropical flora, Peer Community J. 3 (2023).
[45] A. Gallardo-Ferrand, L.A. Escudero-Colomar, J. Avilla, D. Bosch-Serra, Thrips (thysanoptera: terebrantia) in nectarine orchards in north-east Spain: species diversity and fruit damage, Insects 15 (9) (2024) 699.
[46] A. Goudarzi, S. Samavi, M.A. Mazraie, Z. Majidi, Fungal pathogens associated with pre-and postharvest fruit rots of mango in southern Iran, J. Phytopathol. 169 (9) (2021) 545–555.
[47] K. Wada, Labelme: image polygonal annotation with Python. https://doi.org/10.5281/zenodo.5711226, 2019. https://github.com/wkentaro/labelme.
[48] L. Zhang, G. Zhou, C. Lu, A. Chen, Y. Wang, L. Li, W. Cai, MMDGAN: a fusion data augmentation method for tomato-leaf disease identification, Appl. Soft Comput. 123 (2022) 108969.
[49] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[50] M. Zhang, S. Xu, W. Song, Q. He, Q. Wei, Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion, Remote Sens. 13 (22) (2021) 4706.
[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.
[52] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
[53] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[54] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection, in: The Eleventh International Conference on Learning Representations, 2022.
[55] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large batch optimization for deep learning: training bert in 76 minutes, arXiv preprint arXiv:1904.00962 2019.
[56] Z. Hou, B. Yu, D. Tao, Batchformer: learning to explore sample relationships for robust representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 7256–7266.
[57] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658–666.
[58] K.-A. Tessera, S. Hooker, B. Rosman, Keep the gradients flowing: using gradient flow to study sparse network optimization, arXiv preprint arXiv:2102.01670 2021.
[59] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, arXiv preprint arXiv:1506.06579 2015.
[60] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics yolo, 2023. https://github.com/ultralytics/ultralytics.
[61] Kaggle, PlantVillage for object detection yolo (2024). https://www.kaggle.com/datasets/sebastianpalaciob/plantvillage-for-object-detection-yolo (2025-Jun-25).
[62] H. Wang, Y. Li, L.M. Dang, H. Moon, An efficient attention module for instance segmentation network in pest monitoring, Comput. Electron. Agric. 195 (2022) 106853.
[63] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4203–4212.
[64] Kaggle, PlantVillage for object detection (2023). https://www.kaggle.com/datasets/sebastianpalaciob/plantvillage-for-object-detection-yolo/ (2023-May-11).
[65] J. Cao, H. Cholakkal, R.M. Anwer, F.S. Khan, Y. Pang, L. Shao, D2det: towards high quality object detection and instance segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11485–11494.
[66] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: coupling global structure with local parts for object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 4126–4134.