

Article

A Metrologically Guided YOLOv12 Framework with Augmentation-Free Training and Two-Phase Optimization for Multiclass Tomato Leaf Disease Detection

Ihtisham Ul Haq ^{1,*} , Francesco Felicetti ¹ , Domenico Luca Carnì ¹  and Francesco Lamonaca ^{1,2} 

¹ Department of Computer Engineering, Modelling, Electronics and Systems Engineering, University of Calabria, 87036 Arcavacata di Rende, Italy

² Consiglio Nazionale delle Ricerche—Istituto di Nanotecnologia (CNR–NANOTEC), Institute of Nanotechnology, 87036 Arcavacata di Rende, Italy

* Correspondence: ihtishamul.haq@dimes.unical.it

Abstract

Tomatoes are highly vulnerable to a wide range of leaf diseases, which significantly reduce agricultural yield and quality. Timely and precise detection of these diseases is essential for sustainable crop management and food security. This study analyzes configuration-level bidirectional multi-scale feature propagation within the native YOLOv12-s architecture, with emphasis on architectural behavior under controlled experimental conditions. The computational topology and parameterization of YOLOv12 are preserved, while bidirectional feature aggregation is activated at configuration level to examine its influence on cross-scale semantic consistency and localization reliability. The framework was trained and evaluated on a curated dataset of 4030 annotated RGB images spanning ten tomato leaf disease categories. All models were trained under an augmentation-free protocol and unified evaluation settings to isolate architectural effects from data-driven performance inflation. Under these controlled conditions, configuration-level bidirectional activation yields measurable improvements in detection consistency and spatial agreement while maintaining identical model complexity. Performance is evaluated using mAP, precision, recall, F1-score, and error-type decomposition within a measurement-consistency framework. The proposed configuration achieves 95.9% mAP@50 and 87.1% mAP@50–95 under identical experimental conditions, providing empirical evidence that topology-preserving feature routing influences multi-scale semantic stability in lesion detection.

Keywords: precision agriculture; metrological validation; plant pathology; real-time disease identification; image-based diagnosis; model calibration and traceability; leaf lesion segmentation



Academic Editors: Abdussalam Elhanashi and Pierpaolo Dini

Received: 20 January 2026

Revised: 17 February 2026

Accepted: 24 February 2026

Published: 26 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Tomato (*Solanum lycopersicum*) is among the most widely cultivated and consumed horticultural crops globally, owing to its considerable economic value and nutritional significance. Native to South America, species within the *Solanum* genus have undergone extensive diversification, supporting a current global tomato output exceeding 180 million metric tons per year. Major contributors to this production include China, India, and the United States, which collectively rank among the world's leading producers [1,2]. Tomatoes are valued for their rich content of vitamin C, potassium, folate, and lycopene, an antioxidant linked to reduced risks of heart disease and cancer [3,4]. Moreover, tomatoes

display substantial genetic variability, which establishes them as an important model organism for investigating fruit ripening processes, hormonal regulation, and nutrient biosynthetic pathways [4,5].

Despite their agronomic and economic value, tomato crops are particularly prone to a variety of pathogens, including bacterial spot, blights, and viruses such as TYLCV, which collectively account for up to 80% yield losses in severely affected regions [6–8]. Recent estimates suggest global economic losses from tomato diseases reach billions of dollars annually, often requiring extensive pesticide application, which increases costs and raises environmental concerns [9–11]. A concise and accurate disease diagnosis system is thus essential for enabling sustainable and cost-efficient tomato farming. Traditional methods relying on visual inspection or lab-based assays are not scalable for commercial agriculture due to being time-consuming and labor-intensive [12,13].

In recent years, deep learning (DL) has revolutionized plant disease diagnostics by offering scalable, fast, and highly accurate image-based classification systems [13,14]. Among DL-based object detectors, the YOLO (You Only Look Once) family has garnered considerable attention due to its real-time processing capabilities, compact design, and strong performance across diverse visual recognition tasks [15]. While YOLOv5 and YOLOv8 have demonstrated success in plant disease detection, a key limitation persists: ensuring traceability, reproducibility, and standardization in real-world agricultural settings, where measurement reliability is paramount. In this study, we adopt the YOLOv12 scale as the primary baseline to align with embedded/edge deployment constraints, and we perform scale-matched comparisons against YOLOv8s and YOLOv5s under an identical training and inference protocol (input 640×640 , batch = 1, Tesla T4). This ensures a fair and consistent evaluation of both accuracy and latency across models.

To address this gap, this study introduces a YOLOv12-based pipeline for multiclass detection and classification of tomato leaf diseases, utilizing a high-quality annotated dataset from Roboflow. YOLOv12 was originally introduced by Tian et al. [16] as an attention-centric real-time object detection framework, emphasizing efficient feature aggregation and improved spatial reasoning. YOLOv12 builds upon the architecture of YOLOv11 by incorporating CSPDarkNet as a backbone, SPPF modules, anchor-free detection heads, and lightweight attention mechanisms to boost detection accuracy across scales.

In this work, YOLOv12 is adopted as a foundational architecture rather than being re-defined or structurally modified. The selection of YOLOv12 is motivated by its architectural alignment with the challenges inherent to plant disease detection, particularly the presence of small, heterogeneous lesions and significant scale variability across leaf surfaces. In contrast to earlier YOLO versions such as YOLOv5 and YOLOv8, which rely on anchor-based detection and coarser bounding-box regression, YOLOv12 adopts an anchor-free formulation coupled with Distribution Focal Loss (DFL), enabling more precise localization through probabilistic modeling of bounding-box coordinates. Moreover, its efficient backbone–neck design provides a favorable balance between representational capacity and computational cost, making it well-suited for real-time inference and potential deployment on resource-constrained edge platforms commonly used in precision agriculture.

To enhance scientific rigor and deployment relevance, the study is designed around a controlled and measurement-oriented evaluation protocol, in which detection outputs are treated as repeatable and traceable measurands under fixed operating conditions. This allows architectural and optimization effects to be isolated and quantitatively compared in a reproducible manner.

The key contributions are outlined as follows:

- Development of a deployment-aware YOLOv12-s-based framework for detecting ten tomato leaf diseases under a controlled and reproducible evaluation protocol.

- Systematic investigation of configuration-level bidirectional feature activation within the native YOLOv12 neck, preserving the original computational graph and parameterization while examining its influence on multi-scale semantic consistency.
- Implementation of an augmentation-free two-stage optimization strategy designed to isolate architectural behavior from data augmentation effects and training-session artifacts.
- Measurement-oriented comparative analysis against scale-matched YOLOv5s and YOLOv8s baselines under identical operating conditions, emphasizing inter-model consistency, error-type decomposition, and localization reliability rather than absolute performance ranking.

The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of current research related to tomato leaf disease detection using deep learning approaches, while also highlighting the emerging role of metrological principles in AI-driven diagnostic systems. Section 3 details the proposed methodology, encompassing the dataset characteristics, the architectural design of the YOLOv12 model, the two-stage training pipeline, and the conceptual integration of metrological considerations. Section 4 defines the performance metrics employed to evaluate the detection system, including accuracy-based and consistency-based indicators, and subsequently presents the experimental evaluation, analyzes the model's classification and localization performance, and discusses the implications of the findings. Finally, Section 5 concludes the study by summarizing the key contributions and outlining future directions aimed at advancing metrologically grounded and reproducible AI solutions for agricultural applications.

2. Literature Review

Tomato leaf disease detection using deep learning has advanced significantly, particularly with the development of object detection models such as the YOLO family. To structure the discussion and highlight comparative insights, this section is divided into three subsections: convolutional neural network (CNN)-based approaches, YOLO-based advances, and hybrid/optimized models.

2.1. CNN-Based Approaches and Hybrid Detectors

CNNs have been widely adopted for image-based disease classification owing to their capacity to learn and represent high-level visual features. However, these models typically lack object localization capability and are less effective in real-time detection scenarios compared to YOLO-based detectors. Ramose et al. [17] combined CNN with YOLOv3 to leverage CNN feature extraction while benefiting from YOLO's fast object localization. While this hybrid model improved accuracy, it introduced additional computational complexity and slower inference.

2.2. YOLO-Based Advances for Tomato Disease Detection

YOLO (You Only Look Once) models [18] have become a dominant architecture for real-time agricultural applications due to their unified detection pipeline and inference speed. Lee et al. [19] studied tomato leaf disease recognition using the YOLOv11 architecture with systematic hyperparameter optimization. Among the YOLOv11 configurations, YOLOv11m was identified as providing the most favorable balance between predictive performance and computational efficiency, attaining a fitness score of 0.98885, with corresponding recall, precision and mAP@0.5 values of 0.98597, 0.99104 and 0.99197, respectively. Subsequent refinement using a one-factor-at-a-time optimization scheme in conjunction with random search further enhanced the fitness score to 0.99268, while precision, recall, and mAP@0.5 increased to 0.99190, 0.99348, and 0.99262, respectively.

Abulizi et al. [20] proposed DM-YOLO (a YOLOv9 derivative) with DySample modules for improved lesion localization under occlusion, reporting 92.5% precision and 86.4% mAP. Similarly, Wang and Liu [21] proposed TomatoGuard-YOLO, a lightweight YOLOv10-based detector integrating a multi-path inverted residual unit and a dynamic focusing attention mechanism for improved multi-scale feature fusion and lesion localization. The framework additionally incorporates a Focal-EIoU loss to mitigate class imbalance and enhance bounding-box regression performance, attaining an mAP₅₀ of 94.23% at an inference speed of 129.64 FPS with a compact model size of 2.65 MB on a real-world tomato disease dataset.

Robustness under challenging visual conditions has also been investigated. Cardellicchio et al. [22] and Sun et al. [23] confirmed YOLO's resilience under inconsistent lighting and occlusion, where YOLOv5/YOLOv8 variants maintained stable performance and YOLO-FMDI reduced parameters via Feature Diffusion Interaction and UIB modules. For real-world deployment, Obu et al. [24] demonstrated that YOLOv5 can run on a Raspberry Pi with minimal resource requirements, motivating lightweight deployments for newer YOLO generations.

Beyond tomato-specific studies, modular customization trends are consistent across YOLO versions, including improved losses and attention/fusion modules (e.g., improved YOLOv3 pipelines [25], channel-attention pyramids [26], and fusion-aware variants [27]). Related work has also extended YOLO to attention-driven multi-scale fusion for tomato leaves [28] and to edge/smart-city contexts [29], indicating architectural portability. YOLOv7-based variants for large-scale and UAV sensing [30,31] and attention-heavy YOLOv8 derivatives in other domains [32,33] further reinforce the value of cross-scale feature integration, which is a core design target in recent YOLO families.

2.3. Hyperparameter Optimization and Practical Deployment Considerations

While architectural enhancements dominate recent literature, hyperparameter tuning has emerged as a crucial factor. Ramos et al. [34] and Solimani et al. [35] used systematic and Bayesian optimization strategies to boost mAP and precision across datasets, showing that even lightweight models benefit significantly from proper tuning. Automated ML and hyperparameter optimization foundations [5,36–39] provide a methodological basis for improving robustness across changing domains (smart agriculture, phenotyping, and environmental monitoring).

Table 1 summarizes representative studies, datasets, and key trade-offs in accuracy, complexity, and field robustness. Overall, the literature indicates that next-generation YOLO pipelines should be evaluated not only on accuracy and speed, but also on reproducibility and deployment constraints, particularly for edge and in-field diagnostics.

Recent research has explored various YOLO models for tomato disease recognition and related defect/disease tasks. Alhwaiti et al. [40] used YOLO-based models for plant disease identification and highlighted challenges under complex backgrounds. Kumar and Vani [41] investigated image-based tomato leaf disease detection with YOLO-style pipelines. Oni and Prama [42] introduced a computationally efficient custom CNN tailored for real-time tomato leaf disease detection. Their framework demonstrated strong classification performance while maintaining low inference latency, highlighting its suitability for practical agricultural monitoring applications. Chen et al. [43] improved YOLOv8n for tomato leaf disease detection, and Shen et al. [44] optimized YOLOv8 for precision detection. Ramos and Sappa [17] conducted a comprehensive evaluation of YOLO-based architectures for tomato leaf disease identification, systematically comparing their detection performance and architectural characteristics. Xu et al. [45] demonstrated lightweight YOLOv5 for leaf disease detection in a related crop, emphasizing efficiency. Weng et al. [46]

reported improved feature map fusion for YOLOv6-style detection. Bellout et al. [47] introduced a lightweight YOLO-based framework integrated with IoT infrastructure for real-time tomato leaf disease detection and management. Liang et al. [48] presented an online YOLO-based method for tomato surface defect detection in Foods, demonstrating strong potential for practical deployment.

Table 1. Summary of existing literature on tomato disease detection.

Ref.	Dataset Used	AI Model	Key Contribution	Evaluation Metrics	Limitation
[40]	PlantVillage	YOLOv3	Real-time detection with transfer learning	Accuracy, Precision, Recall, F1	Poor background separation
[41]	Custom Tomato	YOLOv4	CSPDarknet53 + data augmentation	mAP, IoU, FPS	Needs large data
[42]	PlantVillage + Field	YOLOv5	Lightweight model for mobile use	mAP@0.5, Inference Time	Low accuracy in natural light
[43]	Kaggle	YOLOv7	Attention for improved localization	Precision, Recall, mAP@0.5	High computation cost
[44]	Self + PlantVillage	YOLOv8	Early stage classification	Accuracy, mAP@0.5, FPS	Limited regional testing
[17]	Tomato Disease	CNN + YOLOv3	Hybrid model for improved feature extraction	Accuracy, Precision	Slower inference
[45]	PlantVillage	Efficient YOLOv5	Reduced model size and inference time	mAP, FPS, Size	Slightly less accurate
[46]	Kaggle Tomato	YOLOv6	Improved feature fusion under lighting noise	mAP, Time	Poor generalization
[47]	Farm Dataset	YOLOv4-Tiny	Edge-device optimized deployment	Speed, Accuracy	Precision trade-off
[48]	PlantVillage	YOLO-based (tomato defects)	Improved online detection with task-specific design	Accuracy, mAP@0.5, IoU	Task-/setup-dependent benchmarking

3. Proposed Methodology

The proposed methodology for detecting and classifying tomato leaf diseases using the YOLOv12 object detection framework comprises four interlinked components: dataset acquisition and preprocessing, architectural design of the YOLOv12 model, training strategy, and the integration of conceptual metrological principles aimed at enhancing reliability and reproducibility. Additionally, a standardized evaluation protocol (input 640×640 , batch = 1 for latency, fixed random seed) is specified to enable fair, reproducible comparisons and transparent reporting.

3.1. Dataset Description

The tomato leaf disease dataset used in this study was obtained from Roboflow Universe [1]. It contains 4030 high-resolution RGB images, each annotated with one of ten class labels representing various disease categories or healthy leaf conditions. The dataset annotations adhere to the YOLO labeling convention, with bounding box coordinates expressed in normalized form. The dataset includes:

- **Bacterial Spot:** dark, water-soaked lesions that may enlarge and defoliate leaves.
- **Early Blight:** concentric ring lesions forming a “bullseye” pattern.
- **Late Blight:** greenish-black, irregular lesions under moist conditions.
- **Leaf Mold:** yellow spots with olive mold underneath.
- **Septoria Leaf Spot:** small circular lesions with dark borders and gray centers.
- **Target Spot:** irregular lesions with rings and chlorotic halos.
- **Mosaic Virus:** mottled yellow-green patterns and leaf curling.
- **Yellow Leaf Curl Virus (YLCV):** upward curling, yellowing, thickened margins.
- **Spider Mites:** stippling and webbing, leading to chlorosis and drop.
- **Healthy:** non-diseased leaves used for control classification.

To ensure consistency during training, all images were resized to 640×640 pixels, matching the default input resolution for YOLO-style detectors. Auto-orientation corrections were applied based on embedded EXIF metadata to standardize camera alignment.

Unlike many prior studies that employ extensive geometric or photometric augmentations to artificially increase sample diversity [49,50], this work deliberately adopts an augmentation-free baseline. This design choice is not intended to claim that data augmentation is ineffective; rather, it is introduced as a controlled experimental condition to isolate architecture-driven performance gains from data-driven performance inflation.

Data augmentation is widely recognized as a powerful mechanism to improve generalization, particularly under limited data regimes. However, in applied plant pathology, aggressive or uncalibrated synthetic perturbations may alter lesion morphology, spatial texture, or color cues in ways that deviate from natural disease progression, thereby introducing augmentation-driven bias. Such bias can obscure whether observed performance improvements originate from model architecture or from distributional artifacts induced by augmentation.

By preserving original image conditions, the present study aims to (i) maintain lesion morphology fidelity for interpretability, (ii) ensure traceable and repeatable evaluation conditions, and (iii) establish a reproducible architectural baseline for YOLOv12 under realistic visual statistics. This strategy aligns with recent observations that excessive augmentation may mask dataset biases or reduce ecological validity in plant disease detection pipelines [51].

Dataset integrity was further verified through structured label diagnostics and bounding-box parameter analysis, enabling consistent interpretation of detection outputs and supporting controlled comparisons across model variants.

The dataset was partitioned into 70% training, 20% validation, and 10% testing, ensuring uniform class distribution across all subsets for fair evaluation. No geometric or photometric augmentation is applied during training (augmentation-free baseline); images are normalized to $[0, 1]$, and the split is class-stratified with a fixed seed (seed = 42) to ensure exact reproducibility. Inference is reported at 640×640 , and training uses the same resolution for consistency.

Representative samples from the Roboflow tomato leaf disease dataset, covering all ten classes and illustrating intra-class visual variability, are shown in Figure 1. The class-wise distribution of annotations and associated label diagnostic statistics used to assess dataset balance are presented in Figure 2. In addition, the statistical relationships among normalized bounding-box parameters (x , y , width, and height), employed to verify annotation consistency and spatial coverage, are visualized in Figure 3.

3.2. YOLOv12 Model Architecture

To ensure architectural clarity and systematic exposition, the YOLOv12-s configuration adopted in this study is described within a hierarchical backbone–neck–head framework, explicitly distinguishing between the original architectural formulation and the configuration-level enhancements introduced for the present application.

The YOLOv12 architecture used in this study follows the original attention-centric, anchor-free formulation introduced by Tian et al. [16]. The original design integrates CSPDarkNet as backbone, ELAN-style feature aggregation in the neck, and anchor-free detection heads optimized for real-time object detection.

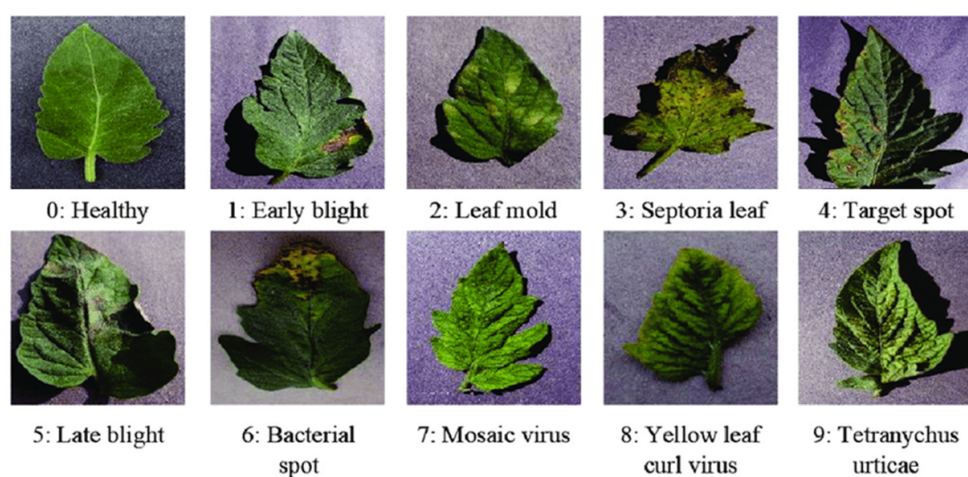


Figure 1. Sample images from the Roboflow dataset showing different tomato leaf diseases.

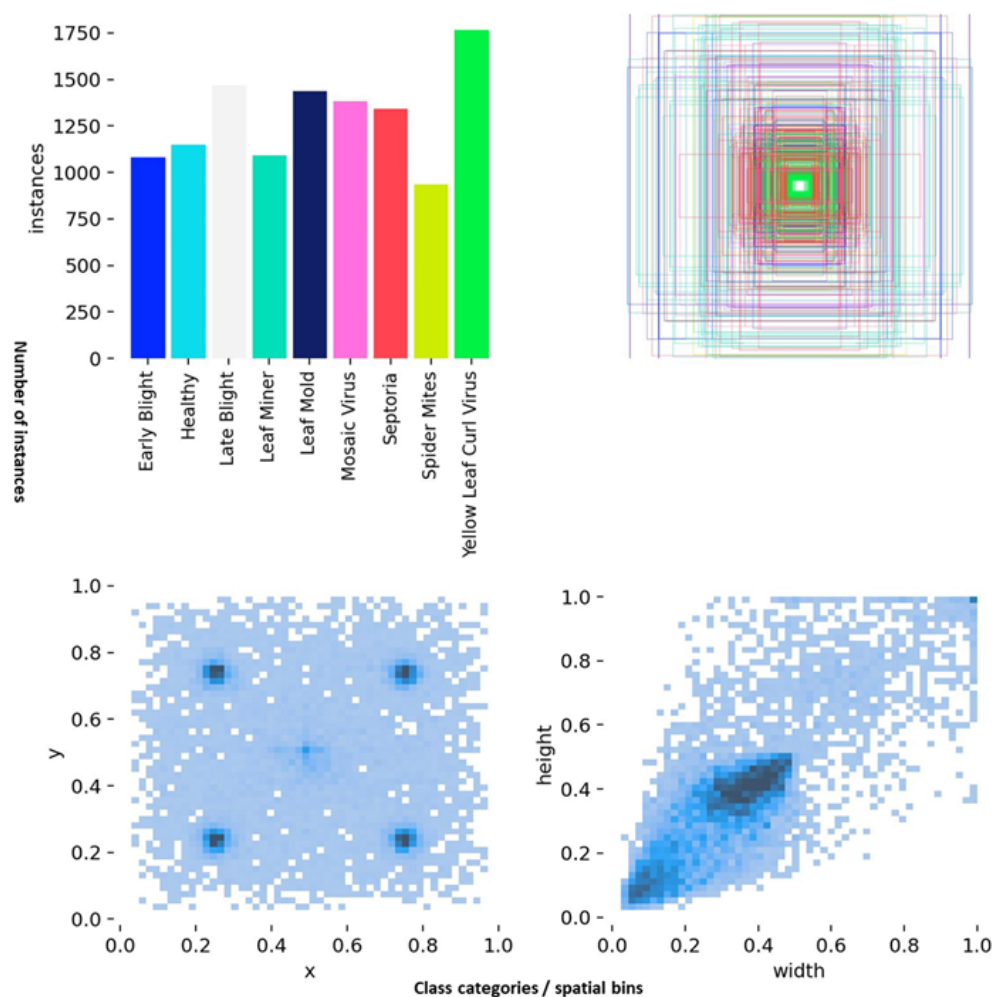


Figure 2. Comparative detection performance of YOLO variants retrained on the same tomato leaf disease dataset under a shared experimental protocol. Distribution analysis of the tomato leaf disease dataset: class-wise instance counts (top-left), bounding box overlay (top-right), spatial distribution of bounding box centers in normalized coordinates (x, y) (bottom-left), and width–height distribution of annotations (bottom-right). In the heatmaps, blue intensity indicates normalized instance frequency per spatial bin, with darker shades representing higher density.

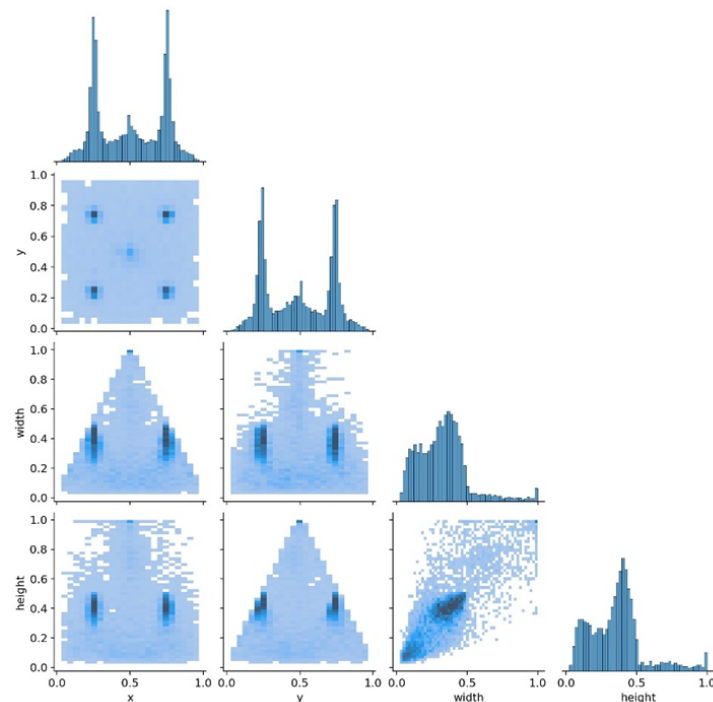


Figure 3. Correlation matrix of bounding box parameters (x , y , width, height) across annotated objects.

Figure 4 provides a structured overview of the complete end-to-end pipeline adopted in this work, including data preparation, model configuration, training strategy, evaluation protocol, and reliability analysis. Within this global framework, the YOLOv12-s model constitutes the central detection component. The figure is organized to visually separate (i) data processing, (ii) architectural configuration, (iii) optimization strategy, and (iv) evaluation and interpretation stages, thereby improving interpretability of the overall methodological flow.

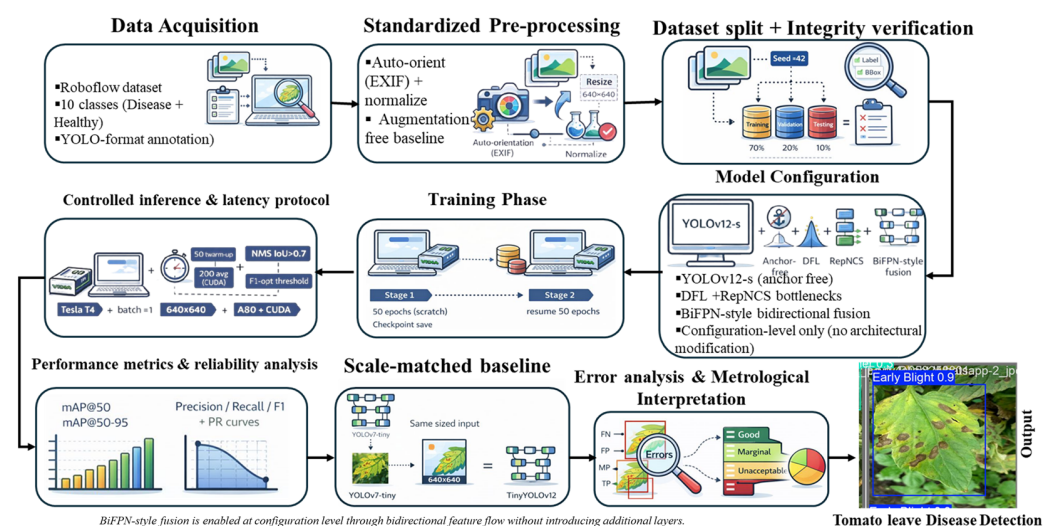


Figure 4. Systematic overview of the proposed YOLOv12-s-based end-to-end framework for multi-class tomato leaf disease detection. The pipeline integrates data acquisition, standardized preprocessing, stratified dataset partitioning with integrity verification, configuration-level YOLOv12-s setup, two-stage optimization, controlled inference and latency evaluation, performance and reliability assessment, scale-matched baseline comparison, and error-oriented metrological interpretation, culminating in disease localization and classification outputs. Bidirectional feature fusion is enabled at configuration level without architectural modification.

For architectural specificity, Figure 5 presents the YOLOv12-s macro-architecture in explicit backbone–neck–head form, illustrating the flow of multi-scale feature maps from CSPDarkNet through the feature aggregation neck to the anchor-free detection head. Complementarily, Figure 6 isolates the neck component and contrasts unidirectional (top-down) feature propagation with the bidirectional fusion behavior activated at configuration level in the present study. Together, these figures provide both structural and functional clarity of the detection architecture.

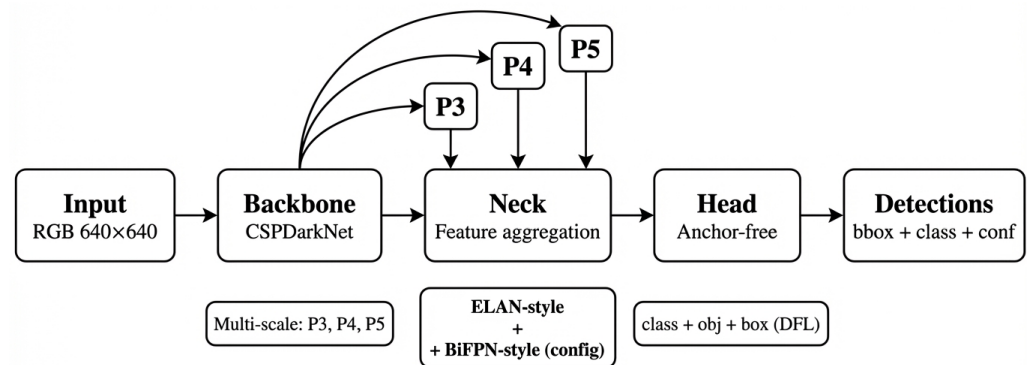


Figure 5. YOLOv12-s macro-architecture (Backbone–Neck–Head) used in this work. Bidirectional feature fusion is enabled at configuration level in the neck without architectural layer additions.

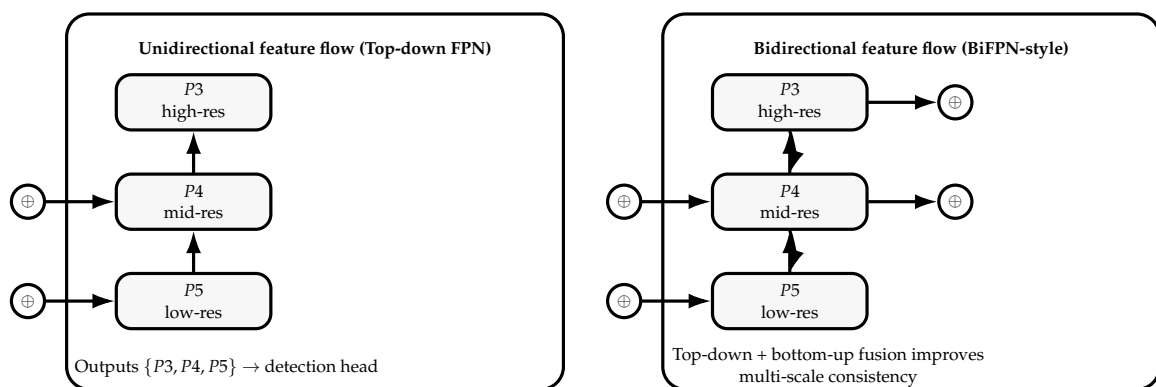


Figure 6. Neck detail comparison: unidirectional (top-down FPN) versus bidirectional (BiFPN-style) feature flow enabling multi-scale fusion at configuration level. $\oplus$ = element-wise addition used for feature aggregation across pyramid levels.

The detection head jointly predicts class confidence, objectness, and a discretized bounding-box distribution for precise localization. The global detection loss comprises (i) an IoU-based regression term for box geometry, (ii) a Distribution Focal Loss (DFL) term for fine-grained box discretization, and (iii) binary cross-entropy terms for class and objectness, consistent with the original YOLOv12 formulation.

In contrast to earlier YOLO variants such as YOLOv5 and YOLOv8, which rely on anchor-based detection and direct regression of bounding-box offsets, YOLOv12 adopts an anchor-free prediction strategy combined with DFL-based probabilistic localization. This design reduces sensitivity to anchor configuration and improves localization stability for small and irregular objects, which are characteristic of leaf disease symptoms.

Backbone: The CSPDarkNet backbone employs Cross Stage Partial (CSP) connections to enhance gradient propagation and computational efficiency. Each convolution block includes batch normalization and SiLU activation:

$$\text{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}, \quad (1)$$

where $\sigma(\cdot)$ denotes the sigmoid function.

The backbone progressively extracts hierarchical spatial–semantic representations from input images resized to 640×640 , generating multi-scale feature maps (P_3 , P_4 , P_5) that are subsequently forwarded to the neck for cross-scale aggregation, as depicted in Figure 5.

Neck: The original YOLOv12 neck employs ELAN-style feature aggregation as described in [16]. In the present work, the computational topology of the original neck is preserved; however, bidirectional information flow across pyramid levels is explicitly activated at configuration level, yielding BiFPN-style feature fusion behavior (Figure 6). Importantly, no additional layers, blocks, or parameters are introduced. The modification operates through controlled feature routing and scale interaction within the native architecture rather than structural redesign. This distinction resolves potential ambiguity between architectural modification and configuration-level activation: the graph structure remains identical to the baseline YOLOv12 formulation, while the effective feature propagation becomes bidirectional, enhancing cross-scale semantic consistency for small, medium, and large lesion regions.

Head: The anchor-free detection head operates on multi-resolution feature maps (e.g., 80×80 , 40×40 , and 20×20), predicting object centers and bounding-box distributions. DFL models each coordinate as a discrete probability distribution, enabling sub-pixel localization via soft-argmax while maintaining stable gradients.

The anchor-free formulation is particularly advantageous for leaf disease detection, where lesion morphology varies significantly and predefined anchor priors may introduce localization bias. The probabilistic regression mechanism improves spatial tightness and reduces bounding-box instability under irregular lesion boundaries. It is explicitly emphasized that no architectural modifications, additional modules, or task-specific layers are introduced beyond the native YOLOv12 design. Performance differences observed in this study arise from (i) scale selection (YOLOv12-s), (ii) configuration-level bidirectional feature activation, (iii) augmentation-free controlled training, and (iv) deployment-oriented evaluation protocol, rather than structural redesign of the original architecture.

3.3. Training Strategy and Configuration

Training was conducted on Google Colab using NVIDIA Tesla T4 GPUs. A two-stage plan was adopted:

- **Stage 1:** Training from scratch for 50 epochs.
- **Stage 2:** Resuming training for an additional 50 epochs using saved weights.

The augmentation-free training protocol adopted in this study is intentionally coupled with the two-stage optimization strategy to establish a stable and interpretable convergence baseline. In contrast to augmentation-intensive pipelines, which may entangle architectural learning with synthetic data effects, this configuration enables the intrinsic learning dynamics of the YOLOv12 architecture to be examined under minimally altered data distributions. The original YOLOv12 study [16] employed a conventional single-session training paradigm for benchmark evaluation. In the present work, the two-stage (checkpoint-resumed) strategy is introduced as an application-level optimization to accommodate cloud-based computational constraints and, more importantly, to examine convergence stability under resumed training conditions, without altering model capacity, optimization objectives, or architectural design.

Crucially, the total number of training epochs remains fixed at 100, ensuring equivalence with a single uninterrupted 100-epoch run. The two-stage protocol therefore does not constitute additional training, but rather a controlled continuation of the same optimization trajectory. The two-stage training protocol was applied exclusively to the proposed YOLOv12 model. Comparative baselines (YOLOv5s and YOLOv8s) were trained using

a single uninterrupted 100-epoch run under otherwise identical experimental conditions, including augmentation-free data, optimizer configuration, input resolution, and fixed random seed. This design ensures that observed performance differences are attributable to architectural and optimization behavior rather than session-dependent or augmentation-induced variance.

Each training phase was initialized with a learning rate of 0.01 and a batch size of 16, while a cosine annealing schedule was employed for learning rate adjustment:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos \left(\frac{T_{\text{cur}}}{T_{\text{max}}} \pi \right) \right). \quad (2)$$

Each phase took approximately 4200 s. Validation metrics per epoch included classification loss, objectness score, mAP, precision, and recall. Mixed-precision training (FP16) with gradient scaling and an exponential moving average (EMA) of model weights is employed to stabilize convergence. Unless otherwise specified, optimization uses AdamW (betas = (0.9, 0.999)) with weight decay 5×10^{-4} , a linear warm-up for the first 3 epochs, and gradient clipping at 10.0. Non-maximum suppression (NMS) applies an IoU threshold of 0.7; confidence thresholds are selected post hoc via the F1-optimal operating point on the validation set. All experiments adopt a fixed random seed (seed = 42) and deterministic CuDNN settings where applicable to maximize run-to-run repeatability. Notably, the exclusion of augmentation does not imply that higher absolute accuracy cannot be achieved. Instead, it establishes a conservative and reproducible baseline against which the incremental benefits and potential biases introduced by augmentation strategies can be systematically evaluated in future work.

Latency measurement protocol: To ensure fair timing, inference latency is measured on a single NVIDIA Tesla T4 with batch = 1 and input 640×640 . The first 50 iterations are treated as warm-ups; wall-clock time is then averaged over the subsequent 200 iterations using CUDA events (forward pass only). Pre-/postprocessing (image decoding, resizing) is excluded, and NMS is included. Performance results are expressed as the mean $\pm$ standard deviation, reported in milliseconds per image. Importantly, the two-stage optimization does not alter model capacity or inference cost. Instead, it facilitates late-stage weight refinement and reduces sensitivity to transient training instabilities, allowing accuracy improvements to emerge through stabilized convergence rather than through extended training or architectural scaling.

Epoch budget as a controlled optimization variable: In this study, the number of epochs is treated as a controlled *optimization budget* rather than a routine hyperparameter. Under the augmentation-free protocol, early epochs primarily support coarse representation learning, whereas later epochs refine localization and confidence calibration under the same data distribution. To make the role of the epoch budget explicit, we report epoch-wise validation trends and quantify the marginal performance gained in the refinement phase (epochs 51–100) relative to the initial phase (epochs 1–50). This enables the reader to interpret performance improvements as convergence stabilization and spatial-consistency refinement, rather than as an artifact of extended or unconstrained training.

3.4. Metrological Considerations

Although physical calibration was not performed, the proposed framework embeds metrological principles to support traceability, consistency, and scientific rigor in AI-based visual detection. In the context of this study, the object detection system is interpreted as a measurement system, in which disease presence and spatial extent are inferred from image-based observations under controlled and repeatable conditions.

Within this interpretation, the detection outputs are treated as measurement quantities derived from a fixed acquisition and preprocessing pipeline. Specifically, the disease class label and the geometric descriptors of the detected region—namely the bounding-box center coordinates (c_x, c_y) and dimensions (w, h) in image space—constitute the primary measurands. These quantities represent spatial and categorical measurements of disease manifestations on tomato leaves, inferred from RGB imagery resized to a standardized resolution (640×640) .

The measurement process comprises a deterministic sequence of operations, including image normalization, multi-scale feature extraction, bidirectional feature aggregation, and anchor-free inference of bounding-box geometry and class confidence. The resulting bounding boxes correspond to spatial measurements of disease regions, while confidence scores provide a probabilistic indication of detection reliability. Traceability is promoted through standardized reporting based on widely adopted object-detection metrics, including mAP@0.5 and mAP@0.5:0.95 [22], which enable benchmarkable and reproducible comparisons across models and studies operating on the same dataset.

From a metrological standpoint, repeatability refers to the ability to obtain consistent measurement outputs under identical experimental conditions. In this work, repeatability is strengthened through the use of fixed dataset partitions, a fixed random seed, identical optimizer configurations, and a controlled two-stage (checkpoint-resumed) training protocol. This strategy ensures that observed variations in detection outputs arise from model learning behavior rather than from stochastic initialization effects, cloud-session interruptions, or transient numerical instabilities.

In particular, checkpoint-resumed training enables the same optimization trajectory to be deterministically continued across computational sessions, reducing sensitivity to GPU initialization variance and session-dependent fluctuations. This consideration is especially relevant in long training regimes executed under constrained or time-limited cloud environments, where uninterrupted execution cannot always be guaranteed. Additional validation tools, including confusion matrices, precision–recall (PR) curves, and class-wise performance analysis, are employed as consistency indicators to assess the stability and reliability of the detection measurements across disease categories.

While the present study does not implement formal uncertainty quantification, the framework is structured to support future extensions such as bounding-box uncertainty analysis, confidence-threshold calibration, and error-bound estimation on localization accuracy. These extensions would enable explicit uncertainty modeling and closer alignment with classical measurement theory; nevertheless, the current formulation establishes a metrologically interpretable evaluation path in which localization fidelity and error behavior can be compared under controlled operating conditions.

4. Results and Discussion

The performance results of the YOLOv12 model over 100 training epochs are split into two distinct phases: initial training (epochs 1–50) and extended training (epochs 51–100). Each subsection analyzes loss trends, precision–recall curves, confidence thresholds, class-wise performance, and comparative evaluations. All metrics and latencies follow the standardized protocol (input 640×640 , batch = 1 for timing on Tesla T4, confidence threshold chosen at the F1-optimal operating point).

In addition to reporting quantitative results, this section interprets convergence behavior and error characteristics to assess detection reliability and deployment relevance under controlled conditions.

4.1. Training Performance (Epochs 1–50)

During the first 50 epochs, YOLOv12 demonstrated stable learning with rapid loss reduction and consistent validation improvements. As shown in Figure 7, the box loss decreased from approximately 3.0 to 0.58, while classification loss declined from 4.43 to 0.67.

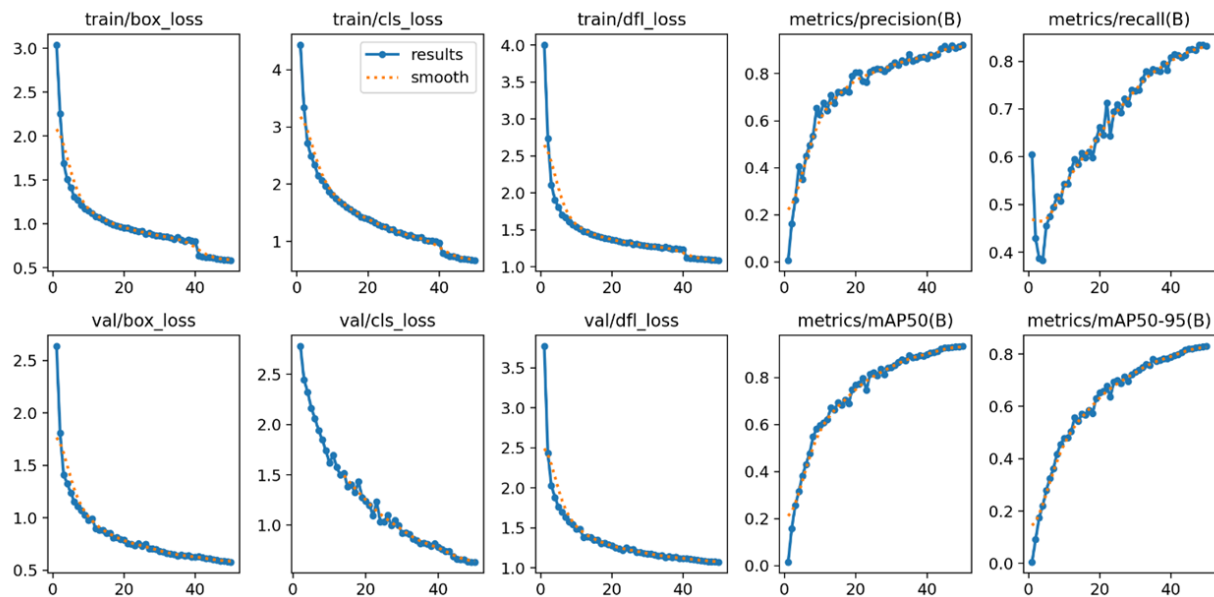


Figure 7. Training and validation loss curves (epochs 1–50), including box loss, classification loss, and Distribution Focal Loss (DFL).

Distribution Focal Loss decreased from 4.0 to 1.1, confirming the model's capability to learn bounding-box regression effectively. Precision rose from 0.006 to 0.922, recall improved from 0.604 to 0.832, and mAP@50 reached 93.35%, while mAP@50–95 reached 82.99%. An early anomaly at epoch 1 caused an infinite classification loss, which was corrected by epoch 3. At this stage, the F1-optimal confidence threshold was already stabilizing near 0.55, indicating an emergent balance between precision and recall.

Early stabilization of the confidence threshold suggests that the detector rapidly achieves balanced decision boundaries, which is desirable for deployment scenarios with limited threshold tuning flexibility.

The confusion matrix (Figure 8) indicated strong early accuracy for Spider Mites ($\approx 97\%$) and Mosaic Virus ($\approx 91\%$), whereas the Healthy class showed occasional misclassification under textured backgrounds. The most frequent cross-confusions occurred between Yellow Leaf Curl Virus and Septoria.

Qualitative inspection of detection outputs after 50 epochs (Figure 9) confirms accurate bounding-box alignment and high confidence scores across multiple disease classes, demonstrating early-stage localization reliability under diverse leaf textures. The results obtained at the end of the first 50 epochs indicate that the model has largely completed its coarse feature learning phase, with major gains in precision, recall, and localization already realized. However, minor metric fluctuations and class-specific confusions suggest that further refinement is required to stabilize late-stage convergence.

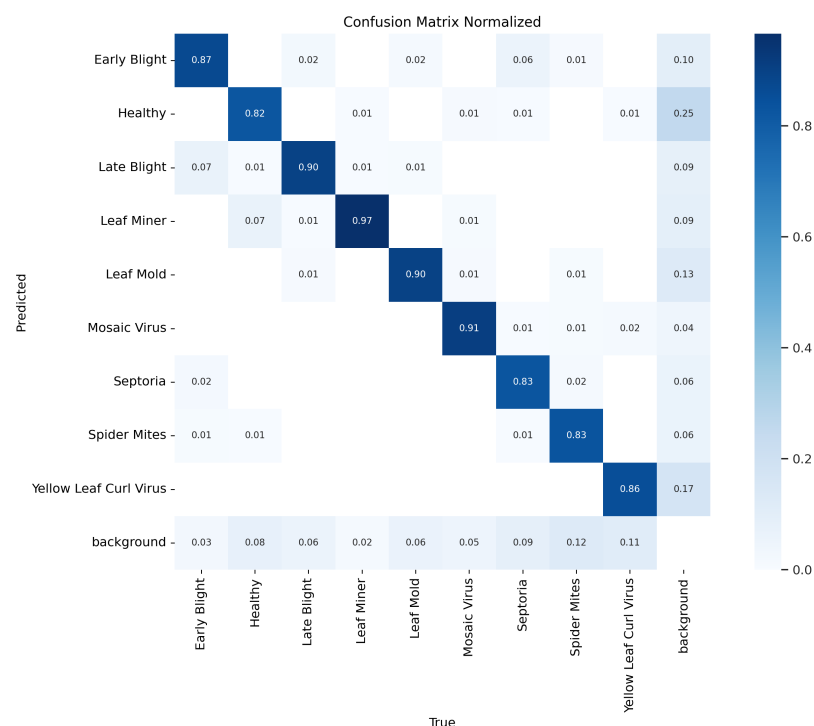


Figure 8. Confusion matrix (epochs 1–50) for the ten disease categories.



Figure 9. Example validation predictions after 50 epochs showing bounding-box alignment and detection confidence.

4.2. Extended Training (Epochs 51–100)

The resumed training phase (epochs 51–100) primarily acts as a refinement stage rather than a re-learning phase. Loss curves exhibit reduced oscillations, and validation metrics

demonstrate smoother convergence, indicating that checkpoint-resumed optimization contributes to stabilizing the final parameter configuration. Importantly, the observed performance gains in the second phase do not arise from additional training beyond 100 epochs, but from improved convergence consistency during the late optimization stage. This behavior supports the use of two-stage training as a repeatability-enhancing strategy rather than as a mechanism for artificial performance inflation.

Further training (epochs 51–100) enhanced model generalization and stabilized metrics. Figure 10 shows continued loss reduction: box loss reached 0.50, classification loss 0.47, and DFL ≈ 1.0 . mAP@50 peaked at 95.91% and mAP@50–95 rose to 87.07%. Class-specific precision increased, as evident in the updated confusion matrix (Figure 11). Representative validation samples from the final training stage (Figure 12) illustrate further improvements in spatial localization and confidence calibration, particularly for visually overlapping disease categories. At convergence, macro-F1 exceeded 0.90 at a confidence threshold of 0.54–0.55, consistent with the curve-based analysis in Figure 13.

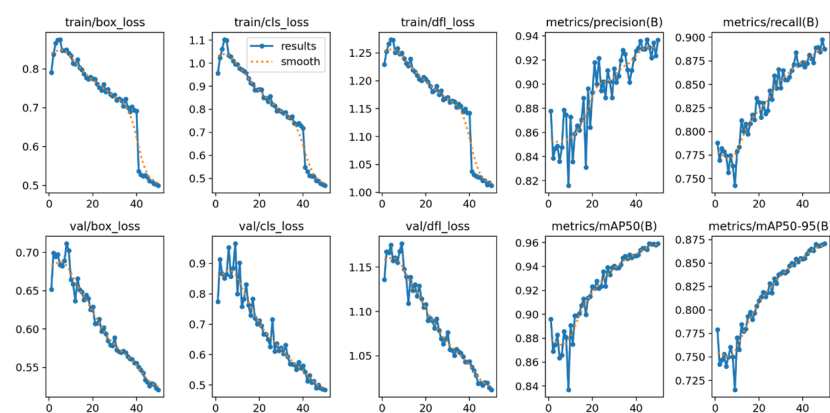


Figure 10. Extended training loss curves (epochs 51–100) illustrating continued model refinement.

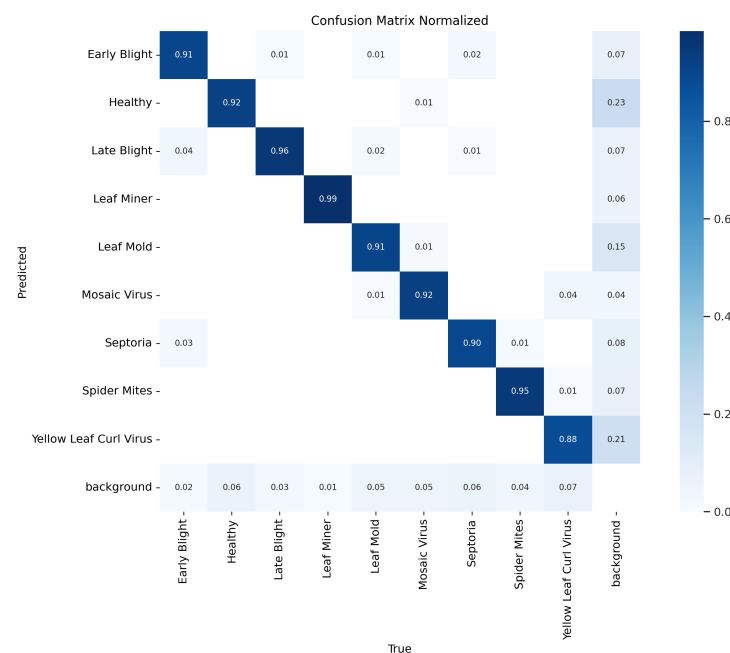


Figure 11. Final confusion matrix after 100 epochs revealing enhanced inter-class discrimination.

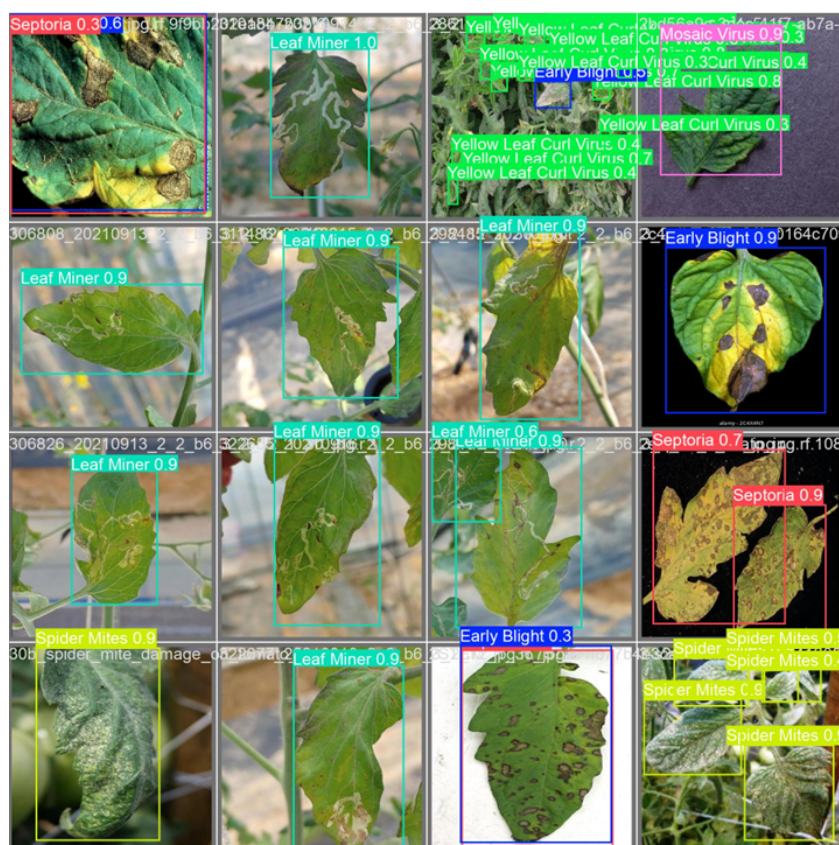


Figure 12. Final validation predictions (epochs 51–100) showing improved localization and detection confidence.

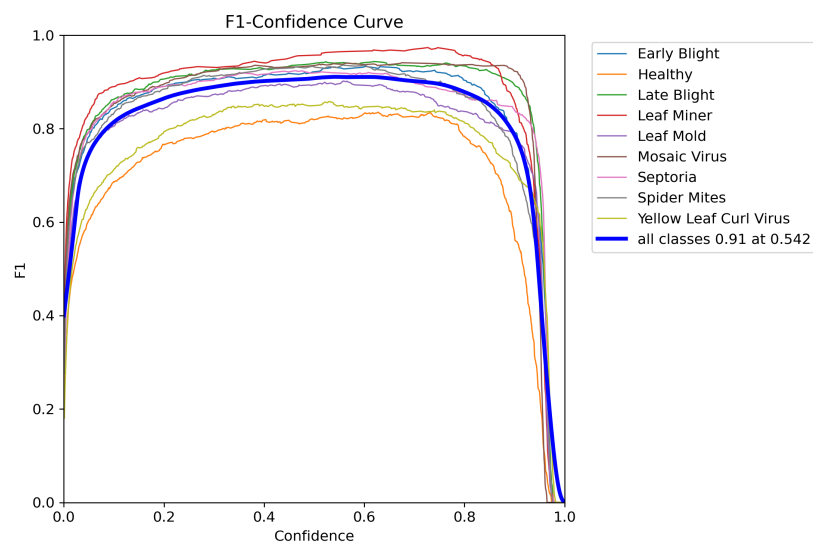


Figure 13. F1-score versus confidence threshold. The optimal threshold shifted from 0.553 to 0.542, with F1 increasing from 0.87 to 0.91.

These results indicate that late-stage optimization primarily improves localization precision and confidence reliability, which are critical for reducing false alarms and missed detections in practical applications.

4.3. Curve-Based Analysis

The class-wise precision–recall curves in Figure 14 shown strong separability for most disease categories, with consistently high precision maintained across a wide recall

range. Yellow Leaf Curl Virus is over-represented, as quantitatively illustrated by the class distribution in Figure 15, which may bias learning toward dominant categories. The model exhibited optimal confidence thresholds around 0.54–0.55, achieving macro-F1 scores above 0.90. Leaf Miner and Early/Late Blight exhibited strong precision–recall trade-offs, reinforcing class separability. Operating at the F1-optimal threshold ensured a stable precision–recall balance across classes with visually similar lesion morphologies.

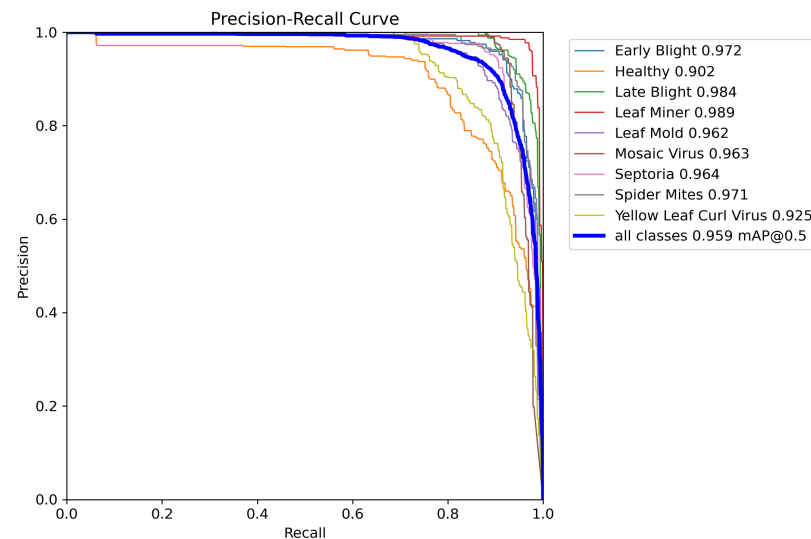


Figure 14. Precision–recall curves per class after training.

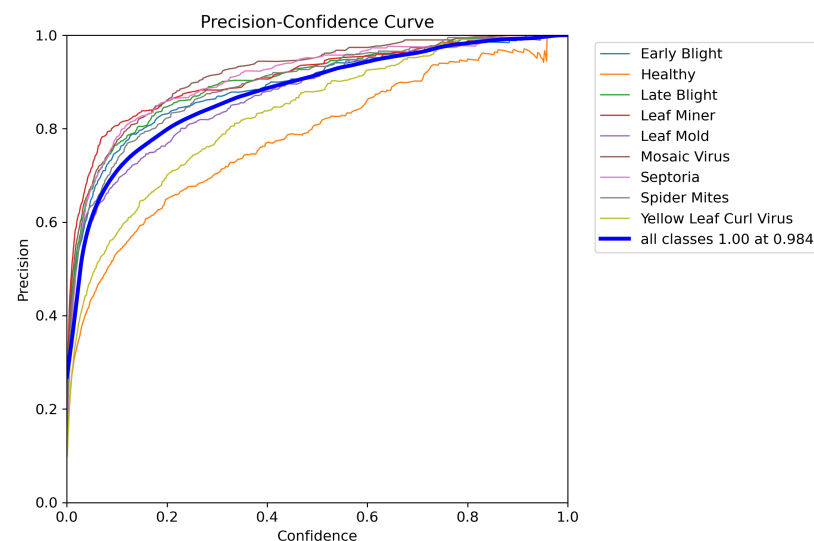


Figure 15. Class distribution highlighting the over-representation of Yellow Leaf Curl Virus instances.

4.4. Comparative Evaluation

A scale-matched comparative study was conducted against common YOLO baselines using a shared tomato leaf disease dataset and a controlled experimental protocol (input 640×640 , augmentation-free training, identical data splits, fixed random seed, and inference latency measured on Tesla T4 with batch = 1). In addition to latency reported in milliseconds per image, inference throughput is also expressed as Frames Per Second (FPS), computed as $\text{FPS} = 1000/\text{ms}$, to facilitate real-world deployment interpretation. The comparison emphasizes relative detection accuracy and computational efficiency under constrained and reproducible conditions.

Importantly, this comparative evaluation does not claim that the selected experimental configuration represents the individually optimal operating point for each model. Instead, all YOLO variants are intentionally evaluated under a common training and inference protocol to enable a controlled, architecture-relative performance comparison on the same dataset. While numerical improvements are observed, the primary objective of this comparison is not to assert architectural superiority but to analyze how configuration-level bidirectional feature propagation influences cross-scale semantic consistency under controlled conditions. All quantitative results reported in this subsection were obtained through in-house re-training and evaluation conducted by the authors. No numerical performance values were adopted from external literature. References cited in the “Source” column identify the original architectural formulations of the respective YOLO variants and serve solely for attribution purposes, not as sources of experimental results.

From a metrological perspective, the comparative evaluation can be interpreted as an inter-model consistency analysis, where each YOLO variant acts as an independent measurement system observing the same physical phenomenon (leaf disease patterns) under identical observational conditions. This approach enables relative assessment of measurement behavior rather than claims of absolute model optimality across arbitrary deployment scenarios.

The augmentation-free training protocol is consistently applied across all models to avoid augmentation-induced distribution shifts that could obscure attribution of performance differences. While certain models may benefit from task-specific augmentation strategies, excluding augmentation establishes a conservative and reproducible baseline in which observed differences primarily reflect architectural design and optimization dynamics rather than data manipulation effects. To ensure repeatability and controlled comparison, all models were trained for an equivalent total of 100 epochs. The proposed YOLOv12 model employed a two-stage (checkpoint-resumed) protocol (50 + 50 epochs), whereas YOLOv5s and YOLOv8s were trained using a single uninterrupted 100-epoch run under otherwise identical optimization settings. This design ensures that performance differences are not attributable to unequal training duration or stochastic initialization effects.

Table 2 summarizes detection performance metrics, while Table 3 reports model complexity indicators to contextualize accuracy–efficiency trade-offs. In metrological terms, precision and recall quantify false-alarm and miss-detection rates (Type-I and Type-II errors), respectively, while mAP@50–95 provides a multi-threshold measure of spatial agreement between predicted and reference bounding boxes. These metrics jointly characterize both detection reliability and localization consistency.

Table 2. Comparative performance of YOLO models under unified settings (input 640×640 ; augmentation-free training; fixed split and seed; inference measured on Tesla T4 with batch = 1; NMS included).

Model	mAP@50	Recall	Precision	mAP@50–95	Speed (ms)	FPS	Key Features	Source
YOLOv5s	89.50%	78.20%	84.70%	63.10%	5.2	192.3	PANet, CIOU	[49]
YOLOv8s	91.30%	81.50%	86.40%	67.30%	4.8	208.3	BiFPN, SimOTA	[50]
YOLOv12 (orig.)	93.40%	85.70%	90.20%	83.50%	5.1	196.1	DFL, RepNCS	[51]
YOLOv12 (proposed)	95.90%	88.90%	93.60%	87.10%	4.6	217.4	DFL, RepNCS, BiFPN	This work

Table 3. Model complexity and parameter comparison of YOLO variants under identical input resolution (640×640).

Model	Parameters (M)	GFLOPs	Model Size (MB)	Detection Paradigm
YOLOv5s	~7.2	~16.5	~14.0	Anchor-based
YOLOv8s	~11.2	~28.6	~22.0	Anchor-based
YOLOv12 (orig.)	~9.5	~24.1	~19.0	Anchor-free (ELAN neck)
YOLOv12 (proposed)	~9.5	~24.1	~19.0	Anchor-free (BiFPN-style)

The entry “YOLOv12 (orig.) ” corresponds to the baseline configuration introduced by Tian et al. [16]. The proposed configuration retains the same computational graph and parameterization while enabling BiFPN-style bidirectional feature aggregation behavior through configuration-level control. Because the computational topology and parameter count remain unchanged, performance differences can be interpreted as resulting from feature routing behavior rather than structural redesign. This distinction is central to the contribution of the present study. The structural similarity between the macro-architecture in Figure 5 and the original YOLOv12 schematic further confirms that the topology is preserved, and that the observed differences arise from configuration-level activation of bidirectional feature propagation.

Inference speed and deployment relevance: Beyond latency expressed in milliseconds per image, inference throughput is reported in FPS to evaluate suitability for real-time agricultural monitoring. Under the unified timing protocol (Tesla T4, batch = 1, 640×640 , NMS included), the proposed YOLOv12 configuration achieves 4.6 ms per image, corresponding to 217.4 FPS. This throughput exceeds that of YOLOv5s (192.3 FPS), YOLOv8s (208.3 FPS), and YOLOv12 (orig.) (196.1 FPS), demonstrating that configuration-level bidirectional feature activation does not introduce computational overhead. The results indicate that the observed accuracy and localization improvements are achieved while maintaining high inference efficiency, supporting practical deployment in real-time disease monitoring systems.

Error-Type and Localization Analysis

To further support a measurement-oriented interpretation, an error-type breakdown is reported in Table 4. False positives (FP) and false negatives (FN) are interpreted as Type-I and Type-II detection errors, respectively, while the mean IoU error acts as a proxy for spatial measurement deviation. The reduction in FP and FN rates can be interpreted as improved decision consistency under identical operating thresholds. From a measurement perspective, the decreased mean IoU error indicates tighter spatial agreement, suggesting enhanced cross-scale semantic reinforcement rather than architectural complexity expansion. Importantly, these improvements emerge without increasing parameter count or modifying the underlying topology, supporting the interpretation that configuration-level bidirectional routing contributes to improved multi-scale coherence.

Table 4. Error-type comparison of YOLO models under identical operating conditions (input 640×640 ; F1-optimal confidence threshold; identical NMS settings). Lower values indicate better reliability.

Model	FP (%)	FN (%)	Mean IoU Error (1 – IoU)
YOLOv5s	15.30	21.80	29.5
YOLOv8s	13.60	18.50	26.3
YOLOv12 (proposed)	6.40	11.10	9.2

The proposed YOLOv12 model exhibits substantially lower FP and FN rates, indicating improved reliability in both detection presence and absence decisions. Moreover, the markedly reduced mean IoU error reflects tighter spatial agreement between predicted and reference disease regions, corresponding to higher localization measurement fidelity. Together, these results demonstrate that the proposed framework not only improves aggregate detection accuracy but also enhances the reliability and consistency of disease localization measurements, which is critical for real-world precision agriculture applications where both false alarms and missed detections carry operational cost.

4.5. Contextual Comparison with Recent State-of-the-Art

To further position the present study within the current research landscape, a contextual comparison with recent tomato leaf disease detection frameworks is provided in Table 5. The selected studies [20,52,53] were chosen based on recency (2023–2025), architectural relevance, and reported quantitative performance metrics.

Table 5. Contextual comparison with recent state-of-the-art tomato leaf disease detection methods.

Reference	Model	Dataset	mAP@50	Architecture Modified	Augmentation
[20]	DM-YOLO (YOLOv9-based)	Roboflow; 4124 images; 9 classes	86.4%	Yes (DySample + MPDIoU loss)	Yes
[52]	Improved YOLOv8-Seg	Custom dataset (1600 → 9600 via augmentation)	92.2% (Seg mAP@0.5)	Yes (Seg head + backbone refinements)	Yes
[53]	ViT-CGA (Transformer-based)	Composite (>18k images; 10 classes)	93.9%	Yes (ViT + context-aware attention)	Yes
This work	YOLOv12-s (config-level)	Roboflow; 4030 images; 10 classes	95.9%	No (Topology preserved)	No

Table 5 summarizes representative recent studies on tomato leaf disease detection. Direct reimplementations of these methods was not feasible due to architecture-specific modules (e.g., DySample layers, transformer-based attention blocks, segmentation heads), dataset heterogeneity (e.g., composite or large-scale augmented datasets), and task differences (detection versus instance segmentation), which substantially diverge from the controlled experimental setting adopted in this work.

Abulizi et al. [20] proposed DM-YOLO, a YOLOv9-based framework incorporating DySample layers and MPDIoU loss, evaluated on a Roboflow-derived dataset with augmentation. Yue et al. [52] extended YOLOv8 toward instance segmentation through backbone refinements and segmentation heads, reporting segmentation mAP@0.5 rather than detection mAP@50–95. Karimanzira [53] introduced a transformer-based context-aware architecture trained on a composite dataset exceeding 18,000 images with extensive augmentation strategies. In contrast, the present study preserves the original YOLOv12 computational topology and parameter count while enabling configuration-level bidirectional feature propagation. All experiments were conducted under an augmentation-free protocol and unified operating conditions to isolate architectural behavior from augmentation-driven performance amplification.

4.6. Observations and Challenges

The epoch-wise checkpoints in Table 6 show that the majority of coarse detection capability emerges within the first 50 epochs, whereas the refinement phase (epochs 51–100) contributes measurable gains in recall and, more notably, in mAP@50–95, which reflects multi-threshold spatial agreement. Concretely, mAP@50 increases from 0.9336 (epoch 50) to 0.9591 (epoch 100), while mAP@50–95 increases from 0.8300 to 0.8707 under unchanged topology and parameter count. This supports the interpretation that late-stage training primarily improves localization tightness and decision consistency rather than altering model capacity.

Table 7 summarizes training dynamics, showing smooth metric maturation and late-stage stabilization. Remaining challenges include confusion between visually similar diseases, class imbalance, and sensitivity to variations in illumination.

Challenges included:

- **Healthy-class confusion:** Visual similarity with mild disease symptoms and textured backgrounds.
- **Class imbalance:** Yellow Leaf Curl Virus is over-represented (Figure 15).
- **Visual overlap:** Septoria and Leaf Mold share lesion traits, increasing cross-confusions.

- **Environmental variations:** Illumination changes affected generalization. Future work will explore robustness under occlusion, illumination shifts, and cultivar variability via controlled augmentations and cross-regional validation.

These challenges reflect inherent limitations of RGB-based diagnosis rather than deficiencies of the proposed framework, emphasizing the importance of controlled evaluation and conservative interpretation of results.

Table 6. Epoch-budget significance: validation metrics at representative checkpoints showing coarse learning (early epochs) versus refinement (late epochs) under identical settings.

Epoch	Precision	Recall	mAP@50	mAP@50–95
10	0.6259	0.5433	0.5973	0.4787
25	0.8162	0.7101	0.8208	0.6993
50	0.9227	0.8325	0.9336	0.8300
75	0.9020	0.8434	0.9357	0.8282
100	0.9364	0.8875	0.9591	0.8707

Table 7. Training-phase summary across epochs.

Epoch Range	Train Loss	Val Loss	Precision	Recall	mAP@50	mAP@50–95	Events
1–10	3.04 → 1.14	∞ → 0.97	0.006 → 0.63	0.60 → 0.54	0.015 → 0.60	0.005 → 0.47	Instability resolved
11–25	1.12 → 0.91	0.99 → 0.73	0.67 → 0.82	0.54 → 0.71	0.60 → 0.82	0.47 → 0.70	Metrics surge
26–50	0.88 → 0.58	0.74 → 0.58	0.82 → 0.92	0.69 → 0.83	0.81 → 0.93	0.68 → 0.83	Maturity phase
51–75	0.79 → 0.74	0.65 → 0.60	0.87 → 0.91	0.78 → 0.84	0.89 → 0.94	0.77 → 0.83	Fluctuation correction
76–100	0.73 → 0.50	0.59 → 0.52	0.90 → 0.94	0.85 → 0.89	0.93 → 0.96	0.83 → 0.87	Peak performance

5. Conclusions and Future Work

This work presented a YOLOv12-based deep learning framework for the detection and classification of ten tomato leaf disease categories using RGB imagery from a publicly available, expertly annotated dataset. Without introducing architectural modifications, the study demonstrated that careful configuration of the original YOLOv12 design, combined with disciplined training and evaluation protocols, can yield strong detection accuracy and reliable localization performance under controlled conditions.

The proposed configuration achieved a peak mAP@50 of 95.9% and mAP@50–95 of 87.1%, outperforming scale-matched YOLOv5 and YOLOv8 baselines while maintaining competitive inference efficiency. Across 100 training epochs, the model exhibited stable loss convergence and consistent improvements in precision, recall, and F1-score. Qualitative analysis of confusion matrices and detection outputs further confirmed improved discrimination among visually similar disease categories, including Septoria, Mosaic Virus, and Leaf Mold.

From a metrological perspective, the framework conceptually embeds principles of traceability, repeatability, and reproducibility, despite the absence of physical calibration. Standardized evaluation metrics and fixed operating conditions enable consistent interpretation of detection outputs across experiments. The use of a controlled two-stage training protocol and reliability-oriented error analysis supports the interpretation of bounding-box predictions as repeatable spatial measurands, reinforcing deployment feasibility under constrained computational settings.

It is important to emphasize that the present validation was conducted under a controlled dataset-bounded experimental setting. Although the Roboflow dataset contains naturally captured images exhibiting variability in background, illumination, and lesion morphology, no dedicated cross-regional field validation or independent real-world deployment testing was performed. Consequently, the reported results characterize architectural behavior and dataset-specific generalization rather than fully quantified operational robustness in unconstrained agricultural environments.

5.1. Limitations

Despite the strong performance, several limitations remain:

- Residual misclassification persists for early-stage or mild symptoms, particularly between visually subtle classes such as Healthy and Mosaic Virus.
- Dataset imbalance, notably the over-representation of Yellow Leaf Curl Virus, may influence class-specific generalization.
- Variations in illumination, occlusion, and leaf orientation typical of field conditions remain challenging for RGB-based detection.
- The deliberate adoption of an augmentation-free baseline prioritizes controlled evaluation and architectural attribution; however, this design choice does not explicitly optimize robustness against severe environmental perturbations such as motion blur, extreme lighting shifts, sensor noise, or partial occlusion.
- No independent field-acquired test set was incorporated to quantify domain shift between curated dataset imagery and operational greenhouse or open-field scenarios. As a result, external validity under geographically or climatically distinct conditions remains to be experimentally established.

5.2. Future Work

Future research will extend the present study along the following directions:

- Controlled real-world validation through acquisition of field-captured tomato leaf images across multiple farms and environmental conditions to quantify domain shift and operational robustness.
- Systematic robustness benchmarking under photometric and geometric stress conditions (e.g., controlled illumination shifts, blur, occlusion simulation, and sensor noise injection) to characterize performance degradation profiles.
- Incorporation of structured data augmentation strategies and domain generalization techniques, followed by comparative analysis against the current augmentation-free baseline to quantify robustness gains without conflating architectural effects.
- Evaluation under cross-dataset transfer settings (e.g., PlantVillage, PlantDoc, or region-specific datasets) to measure generalization consistency across heterogeneous agricultural domains.
- Edge-device deployment trials on embedded hardware platforms to validate real-time inference stability under field-operational constraints.
- Controlled ablation of feature routing patterns to quantify the contribution of individual fusion pathways.
- Investigation of alternative topology-preserving feature propagation strategies.

Author Contributions: Conceptualization, I.U.H. and F.L.; Methodology, I.U.H., F.F. and D.L.C.; Software, I.U.H.; Validation, I.U.H., D.L.C. and F.L.; Formal analysis, I.U.H.; Investigation, I.U.H.; Resources, F.F., D.L.C. and F.L.; Data curation, I.U.H.; Writing—original draft, I.U.H.; Writing—review and editing, D.L.C. and F.L.; Visualization, I.U.H. and F.F.; Supervision, D.L.C. and F.L.; Project administration, F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the project “FAIR–Future Artificial Intelligence Research” (I.D.: PE00000013; CUP: H23C22000860006), funded under the Italian National Recovery and Resilience Plan (PNRR), Mission 4, Component 2, Investment 1.3.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors acknowledge Robinia S.r.l.s. for supporting the doctoral research activities of Ihtisham Ul Haq. The author, Francesco Lamonaca, gratefully acknowledges the support

of the Brussels Institute for Advanced Studies (BrIAS), where he serves as a Senior Fellow. BrIAS promotes high-level interdisciplinary and transdisciplinary research by providing an intellectually independent environment that fosters methodological rigor, international collaboration, and cross-fertilization of ideas, thereby supporting scientific excellence and research with societal relevance.

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

References

1. Bryan, B. Tomato Leaf Disease Dataset. Roboflow Universe. 2023. Available online: <https://universe.roboflow.com/bryan-b56jm/tomato-leaf-disease-ssoha/dataset/63> (accessed on 24 June 2025).
2. Citaristi, I. Specialized agencies and related organizations within the UN system: Food and Agriculture Organization of the United Nations—FAO. In *The Europa Directory of International Organizations 2022*; Routledge: Oxfordshire, UK, 2022; pp. 307–315.
3. Tilahun, S.; Park, D.; Seo, M.; Jeong, C. Review on factors affecting the quality and antioxidant properties of tomatoes. *Afr. J. Biotechnol.* **2017**, *16*, 1678–1687. [\[CrossRef\]](#)
4. Sato, S.; Tabata, S. Tomato genome sequence. In *Functional Genomics and Biotechnology in Solanaceae and Cucurbitaceae Crops*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–13.
5. Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.L.; et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2023**, *13*, e1484. [\[CrossRef\]](#)
6. Oliver, R.P. The history and development of plant pathology. In *Agrios' Plant Pathology*; Academic Press: Cambridge, MA, USA, 2024; pp. 3–61.
7. Kommey, B.; Tamakloe, E.; Opoku, D.; Crispin, T.; Danquah, J. Disease Detection in Tropical Tomato Leaves via Machine Learning Models. *J. Eltikom J. Tek. Elektro, Teknol. Inf. Dan Komput.* **2024**, *8*, 179–191.
8. Gai, Y.; Wang, H. Plant disease: A growing threat to global food security. *Agronomy* **2024**, *14*, 1615. [\[CrossRef\]](#)
9. Junaid, M.D.; Gokce, A.F. Global agricultural losses and their causes. *Bull. Biol. Allied Sci. Res.* **2024**, *2024*, 66. [\[CrossRef\]](#)
10. Song, R.; Zhang, Y.; Lu, P.; Wu, J.; Li, Q.X.; Song, B. Status and perspective on green pesticide utilizations and food security. *Annu. Rev. Food Sci. Technol.* **2024**, *15*, 473–493. [\[CrossRef\]](#)
11. Asefa, E.M.; Damtew, Y.T.; Ober, J. Pesticide water pollution, human health risks, and regulatory evaluation: A nationwide analysis in Ethiopia. *J. Hazard. Mater.* **2024**, *478*, 135326. [\[CrossRef\]](#)
12. Routis, G.; Michailidis, M.; Roussaki, I. Plant disease identification using machine learning algorithms on single-board computers in IoT environments. *Electronics* **2024**, *13*, 1010. [\[CrossRef\]](#)
13. Tartoussi, I.; Dornaika, F.; Alirezazadeh, P.; Jaam, J.E. Deep Learning Image-Based Plant Disease Classification—A Review. In Proceedings of the 2024 International Conference on Computer and Applications (ICCA), Cairo, Egypt, 17–19 December 2024; pp. 1–8.
14. Gkountakos, K.; Ioannidis, K.; Demestichas, K.; Vrochidis, S.; Kompatsiaris, I. A Comprehensive Review of Deep Learning-based Anomaly Detection Methods for Precision Agriculture. *IEEE Access* **2024**, *12*, 197715–197733. [\[CrossRef\]](#)
15. Archana, T.S.; Kumar, D.; Ghosh, P.; Kumar, V. Plant pathogen detection: Exploring modern techniques and future directions. *Indian Phytopathol.* **2024**, *77*, 573–585. [\[CrossRef\]](#)
16. Tian, Y.; Ye, Q.; Doermann, D. Yolov12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
17. Ramos, L.T.; Sappa, A.D. A comprehensive analysis of YOLO architectures for tomato leaf disease identification. *Sci. Rep.* **2025**, *15*, 26890. [\[PubMed\]](#)
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
19. Lee, Y.S.; Patil, M.P.; Kim, J.G.; Seo, Y.B.; Ahn, D.H.; Kim, G.D. Hyperparameter Optimization for Tomato Leaf Disease Recognition Based on YOLOv11m. *Plants* **2025**, *14*, 653. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Abulizi, A.; Ye, J.; Abulizi, H.; Guo, W. DM-YOLO: Improved YOLOv9 model for tomato leaf disease detection. *Front. Plant Sci.* **2025**, *14*, 1473928. [\[CrossRef\]](#)
21. Wang, X.; Liu, J. TomatoGuard-YOLO: A novel efficient tomato disease detection method. *Front. Plant Sci.* **2025**, *15*, 1499278.
22. Cardellicchio, A.; Renò, V.; Devanna, R.; Marani, R.; Milella, A. Tomato detection in challenging scenarios using YOLO-based single stage detectors. In Proceedings of the 2023 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Pisa, Italy, 6–8 November 2023. [\[CrossRef\]](#)
23. Sun, H.; Nicholas, I.T.; Fu, R.; Kang, D.K. YOLO-FMDI: A Lightweight YOLOv8 Focusing on a Multi-Scale Feature Diffusion Interaction Neck for Tomato Pest and Disease Detection. *Electronics* **2024**, *13*, 2974. [\[CrossRef\]](#)
24. Obu, U.; Ambekar, Y.; Dhote, H.; Wadbudhe, S.; Khandelwal, S.; Dongre, S. Crop Disease Detection using Yolo V5 on Raspberry Pi. In Proceedings of the 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 19–20 June 2023. [\[CrossRef\]](#)

25. Liu, J.; Wang, X. Tomato diseases and pests detection based on improved YOLO V3 convolutional neural network. *Front. Plant Sci.* **2020**, *11*, 898. [CrossRef]
26. Kim, M.; Jeong, J.; Kim, S. ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image. *Remote Sens.* **2021**, *13*, 4851. [CrossRef]
27. Chen, S.; Zou, X.; Zhou, X.; Xiang, Y.; Wu, M. Study on fusion clustering and improved YOLOv5 algorithm based on multiple occlusion of Camellia oleifera fruit. *Comput. Electron. Agric.* **2023**, *206*, 107706. [CrossRef]
28. Wang, Y.; Zhang, P.; Tian, S. Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion. *Front. Plant Sci.* **2024**, *15*, 1382802. [CrossRef]
29. Saydirasulovich, N.; Abdusalomov, A.; Jamil, M.K.; Nasimov, R.; Kozhamzharova, D.; Cho, Y.I. A YOLOv6-based improved fire detection approach for smart city environments. *Sensors* **2023**, *23*, 3161. [CrossRef] [PubMed]
30. Yang, S.; Xing, Z.; Wang, H.; Dong, X.; Gao, X.; Liu, Z.; Zhang, X.; Li, S.; Zhao, Y. Maize-YOLO: A new high-precision and real-time method for maize pest detection. *Insects* **2023**, *14*, 278. [CrossRef] [PubMed]
31. Liu, Y.; Zhao, Q.; Wang, X.; Sheng, Y.; Tian, W.; Ren, Y. A tree species classification model based on improved YOLOv7 for shelterbelts. *Front. Plant Sci.* **2024**, *14*, 1265025. [CrossRef] [PubMed]
32. Wang, S.; Li, Y.; Qiao, S. ALF-YOLO: Enhanced YOLOv8 based on multiscale attention feature fusion for ship detection. *Ocean. Eng.* **2024**, *308*, 118233. [CrossRef]
33. Fang, C.; Li, C.; Yang, P.; Kong, S.; Han, Y.; Huang, X.; Niu, J. Enhancing livestock detection: An efficient model based on YOLOv8. *Appl. Sci.* **2024**, *14*, 4809. [CrossRef]
34. Ramos, L.; Casas, E.; Bendek, E.; Romero, C.; Rivas-Echeverría, F. Hyperparameter optimization of YOLOv8 for smoke and wildfire detection: Implications for agricultural and environmental safety. *Artif. Intell. Agric.* **2024**, *12*, 109–126. [CrossRef]
35. Solimani, F.; Cardellicchio, A.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [CrossRef]
36. Elshaw, R.; Maher, M.; Sakr, S. Automated machine learning: State-of-the-art and open challenges. *arXiv* **2019**, arXiv:1906.02287.
37. Bergstra, J.; Yamins, D.; Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Proceedings of the 30th International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 115–123.
38. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
39. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
40. Alhwaiti, Y.; Khan, M.; Asim, M.; Siddiqi, M.H.; Ishaq, M.; Alruwaili, M. Leveraging YOLO deep learning models to enhance plant disease identification. *Sci. Rep.* **2025**, *15*, 7969. [CrossRef]
41. Kumar, A.; Vani, M. Image based tomato leaf disease detection. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
42. Oni, M.K.; Prama, T.T. Optimized Custom CNN for Real-Time Tomato Leaf Disease Detection. *arXiv* **2025**, arXiv:2502.18521. [CrossRef]
43. Chen, M.; Wang, C.; Liu, C.; Yu, Y.; Yuan, Y.; Ma, J.; Zhang, K. Tomato leaf disease detection method based on improved YOLOv8n. *Sci. Rep.* **2025**, *15*, 25837. [CrossRef] [PubMed]
44. Shen, Y.; Yang, Z.; Khan, Z.; Liu, H.; Chen, W.; Duan, S. Optimization of improved YOLOv8 for precision tomato leaf disease detection in sustainable agriculture. *Sensors* **2025**, *25*, 1398. [CrossRef] [PubMed]
45. Xu, X.; Zhou, B.; Li, W.; Wang, F. A method for detecting persimmon leaf diseases using the lightweight YOLOv5 model. *Expert Syst. Appl.* **2025**, *284*, 127567. [CrossRef]
46. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
47. Bellout, A.; Zarboubi, M.; Elhoseny, M.; Dliou, A.; Latif, R.; Saddik, A. LT-YOLOv10n: A lightweight IoT-integrated deep learning model for real-time tomato leaf disease detection and management. *Internet Things* **2025**, *33*, 101663. [CrossRef]
48. Liang, Z.; Zhu, T.; Teng, G.; Zhang, Y.; Gu, Z. YOLO-RGDD: A Novel Method for the Online Detection of Tomato Surface Defects. *Foods* **2025**, *14*, 2513. [CrossRef]
49. Sun, H.R.; Shi, B.J.; Hu, Y.L. A Lightweight YOLO-Based Model in Small Object Detection for UAV Optical Sensors. *IEEE Sensors J.* **2025**, *25*, 17585–17599.
50. Singh, A.; Dass, S. Advanced Traffic Conflict Detection and Risk Assessment Using Multi-Scale Video Analysis: A YOLOv8 Modified and Attention-Enhanced Safety Metrics Evaluation. *Int. J. Intell. Transp. Syst. Res.* **2025**, *23*, 1197–1217.
51. Sapkota, R.; Karkee, M. Improved YOLOv12 with LLM-Generated Synthetic Data for Apple Detection. *arXiv* **2025**, arXiv:2503.00057. Available online: <https://arxiv.org/abs/2503.00057> (accessed on 22 November 2025).

52. Yue, X.; Qi, K.; Na, X.; Zhang, Y.; Liu, Y.; Liu, C. Improved YOLOv8-Seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. *Agriculture* **2023**, *13*, 1643. [[CrossRef](#)]
53. Karimanzira, D. Context-aware tomato leaf disease detection using deep learning in an operational framework. *Electronics* **2025**, *14*, 661. [[CrossRef](#)]

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