



AgroYOLOvNext: A Multi-Scale Transformer-Enhanced Framework with Domain-Adaptive Contrastive Learning for Plant Disease Classification on the Plant Village Benchmark

¹ Hasanain Flayyih Hasan*

¹ College of Computer Science and Information Technology, Wasit University, Al-Kut, Wasit, Iraq

Article information

Article history:

Received: June, 23, 2026

Accepted: September, 06, 2026

Available online: September, 25, 2026

Keywords:

plant disease detection,
YOLO architecture,
Transformer networks,
multi-scale feature fusion,
contrastive learning,
deep learning.

*Corresponding Author:

Hasanain Flayyih Hasan
hasanain.f.h@uowasit.edu.iq

DOI:

<https://doi.org/10.61710/2yz6zn46>

This article is licensed under:

[Creative Commons Attribution 4.0 International License.](#)

Abstract

Automated plant disease detection in a real-world agricultural setting remains a difficult problem because of variation in illumination, intra-class diversity, domain shift between laboratory and field data, and occlusions. While recent deep learning-based object detection models, such as YOLO variants and transformer-based models, have demonstrated impressive accuracy on curated datasets, their ability to perform well under open field conditions remains limited. To address this, we introduce Agro YOLO v Next, a multi-scale Transformer-enhanced YOLO framework with domain-adaptive learning that combines three complementary components rather than a single new mechanism. The proposed architecture consists of three main components: (1) a Multi-Scale Depth wise Asymmetric (MSDA) backbone for extracting fine-grained features from multiple receptive fields, (2) a Lightweight Cross-Stage Transformer Neck (LCST-Neck) for improved feature fusion and class separability, and (3) a Domain-Adaptive Contrastive Head (DACH) to encourage feature representations that are more invariant to common imaging perturbations. All experiments were conducted on the laboratory-acquired Plant Village dataset (54,306 images, 38 classes); no field-acquired imagery was used for training or evaluation. All models, including YOLO-family detectors adapted with a global-average-pooling classification head, are evaluated under a common whole-image classification protocol (accuracy, precision, recall, F1-score, AUC); no bounding-box annotations, IoU, or mAP are used, and this comparison basis is stated explicitly so that CNN classifiers and detector backbones are judged on equal footing. Under this protocol, the proposed model achieves an accuracy of 98.34%, an AUC of 0.998, a precision of 98.11%, a recall of 97.89%, and an F1-score of 98.00%, outperforming the strongest of seven baseline architectures (YOLOv9-E) by 1.40

accuracy percentage points. These results indicate the model is effective and competitive on the Plant Village benchmark; claims of real-world field generalization are

not yet supported by field-acquired data and should be regarded as a direction for future validation rather than a demonstrated outcome.

1. Introduction

Food insecurity is one of the most pressing challenges of the 21st century. The population size of the world is constantly projected to be nine to ten billion people by 2050, and given the limited arable land area due to urbanization, soil degradation and climate variability, agricultural productivity per hectare must improve significantly and sustainably. Phytopathological diseases (fungal infection, bacterial blights and viral mosaics) are amongst the most persistent threats to crop productivity, and are estimated to cause 20-40% of annual losses in global crop productivity, and cost of these losses is estimated at hundreds of billions of US dollars annually [1, 2]. Therefore, the early and correct diagnosis of the onset of disease is not an opportunity but an agronomic requirement, as a delay will permit the pathogen to multiply exponentially and a chemical treatment will be more expensive and less effective.

In the last decade, deep convolutional neural networks (CNNs) have moved plant-disease diagnosis from an expert-only task toward large-scale automated analysis: Mohanty et al. [2] showed that a CNN trained on Plant Village could exceed 99% accuracy under controlled conditions, motivating a wave of follow-on work with deeper backbones (Res Net, Dense Net, Efficient Net) and, more recently, Vision Transformers and hybrid CNN-Transformer models. In parallel, the YOLO family of detectors has evolved through successive versions to offer a favorable speed-accuracy trade-off for embedded and edge agricultural sensing. A more detailed account of this literature, organized by architecture family, is given in Section 2.

But there is a well-recognized discrepancy between accuracy in laboratory settings and field accuracy. Although the Plant Village dataset is extensively used, images acquired in laboratories are done against the same background while field-acquired images have varying backgrounds, sun angles, mild blur, and slight occlusion of lesions [24, 25]. Previous research has shown that models trained on lab data without any explicit domain-adaptation strategies can lose between 15-30 accuracy percentage points in uncontrolled test conditions [10, 13]. While this generalization gap is an important motivation for the design choices presented in this paper, the current study does not directly measure this gap, and only evaluates the model on Plant Village.

Prior work has addressed this gap through data augmentation, GAN-based domain translation, transfer learning, and self-supervised pre-training (reviewed in Section 2.3), each with known drawbacks such as translation artifacts, source-target semantic gaps, or reliance on unlabeled field data. To the best of our knowledge, multi-scale feature extraction, Transformer-based global reasoning, and explicit domain-adaptive training have not previously been combined within a single, end-to-end trained framework for this task; we treat this combination, rather than any one mechanism in isolation, as the paper's central contribution.

To address the above, this paper proposes AgroYOLOvNext, which combines three complementary components: a Multi-Scale Depth wise Asymmetric (MSDA) backbone for capturing elongated lesion structures, a Lightweight Cross-Stage Transformer Neck (LCST-Neck) for efficient long-range feature aggregation, and a Domain-Adaptive Contrastive Head (DACH) for encouraging representations that are more robust to common imaging perturbations. The design and implementation details of each component are presented in Section 3.

The contributions of this work are summarized as follows:

- A Multi-Scale Depth wise Asymmetric (MSDA) backbone that uses horizontally and vertically asymmetric convolutional filters to capture lesion texture at multiple scales, showing an accuracy improvement over the standard C2f-YOLO backbone in ablation on the Plant Village test split.
- A Lightweight Cross-Stage Transformer Neck (LCST-Neck) that adds shifted-window multi-head self-attention to a PANet-style neck, targeting the limited long-range feature correlation of conventional FPN/PANet aggregation.

- A Domain-Adaptive Contrastive Head (DACH) trained with a modified Sup Con loss over source-domain images and synthetically shifted variants, intended to encourage representations that are more invariant to common imaging perturbations; no target-domain (field) images are used in the loss.
- A component-wise ablation study, reported quantitatively in Section 4, showing that each of the three components contributes a distinct, non-redundant share of the overall accuracy improvement.
- A standardized comparison against seven CNN, YOLO, and Transformer baselines on Plant Village under an identical evaluation protocol, including a Grad-CAM interpretability analysis showing that the model attends to biologically meaningful lesion regions rather than background texture.

2. Related Work

2.1. CNN-Based Plant Disease Detection

In the early days of deep learning in plant pathology, very few models were used other than large CNN models originally developed for object recognition. In order to selectively reweight the spatial and channel features, [1] modified ResNet-50 by adding a convolutional block attention module (CBAM) on it, obtaining a 94.1% accuracy on Plant Village. CBAM's architecture was still fundamentally restricted to extracting features at a single scale, and did not offer any way to account for domain covariate shift, a limitation the authors themselves admitted in their discussion of the results of the field trials. After the foundation work of [2], they examined MobileNetV3 along with a support vector machine (SVM) classifier and achieved 93.4% accuracy with significant fewer parameters, which makes it deployable on mobile devices. A closer examination of the recall on minority disease classes, however, as shown in Table 1, showed a significant decrease, showing that this class imbalance issue, typical of plant disease datasets, was not being addressed[3] was the first to apply YOLOv8 for plant pathology, and they highlighted the usefulness of anchor-free detection for detecting lesions with varying sizes, but they did not use any multi-scale fusion strategy, which limited the model's performance in detecting lesions of different scales, especially small lesions at early growth stages.

2.2. Transformer Architectures for Agricultural Vision

The use of Vision Transformers in agriculture image analysis has increased significantly since 2023. Instead [5] fine-tuned ViT-B/16 for wheat stripe rust. They use the global self-attention to incorporate inter-lesion spatial dependencies, which are intrinsically absent in locality constrained CNNs. However, the authors pointed out, ViTs need significantly larger datasets of labeled data to match the performance of CNNs in the low-data regime, which can be a challenge for rare or new disease categories [6] introduced a Swin-Transformer-based backbone and a typical feature pyramid network (FPN) and claimed an accuracy of 96.0% on their data set, and described the hierarchical window partitioning as a more feasible computational profile than isotropic ViTs. However, their investigation did not include any type of domain adaptation, and the performance on held-out field-acquired images (as an additional qualitative analysis) was noticeably worse in complex background conditions. The systematic review performed [12] between 2022 and 2024 revealed that multi-scale feature integration and domain-invariant representation learning are two areas that have not been extensively explored with Transformer-based approaches in the context of agricultural remote sensing.

2.3. Domain Adaptation and Robustness Strategies

One of the most technically complex and challenging tasks in agricultural AI is bridging the domain gap between curated data in the lab and data collected in the field [10] posed this as a domain adaptive transfer problem, by training a CNN-based backbone with adversarial feature alignment layers to train the model to minimize the distance between the distributions of source-domain features and target-domain features, measured by Maximum Mean Discrepancy (MMD). Their approach led to significant accuracy gains on the out-of-distribution evaluation sets, but their adversarial training procedure was unstable, with the situation depending on the hyper parameters and the initialization conditions. In contrastive domain adaptation [13] showed that adversarial alignment alone is not explicitly structured by intra-class structure preservation, and suggested adding a contrastive regularizer which was designed to bring close the samples of the same class across two domains and to keep apart the samples of different classes. This direct motivation was the development of the DACH component of AgroYOLOvNext, but the present work further extends the formulation to work in conjunction with multi-scale detection results instead of classification embedding results [16] showed that explicitly training the model for robustness to common image corruptions (Gaussian noise, blur, JPEG compression) measurably improved the robustness of the model to common image corruptions in the field, implying that robustness-aware training should be a standard part of agricultural detection pipelines.

2.4. Hybrid and Lightweight Architectures for Real-Time Detection

But the need for inference speed, low memory usage, and low-power consumption requirements during runtime makes it

difficult for pure Transformer architectures to satisfy these requirements without compromising the architecture. They tackled this problem in [15] by training a compact student network with a bigger EfficientDet teacher to capture the output distributions, thus obtaining the near-parity accuracy with a 4.7 times faster inference time on an NVIDIA Jetson Nano. In order to decrease the need for labeled data and retain competitiveness in accuracy [18] suggested self-supervised pertaining on unlabeled plant images and few-shot training on labeled plant images for fine tuning [11] proposed YOLO-Plant, the multi-head cross-attention was added to the YOLO-neck, the structure of which is quite similar to the LCST-Neck proposed in this paper, but did not use domain adaptation in the explicit process. In a late-fusion architecture [23] used a CNN backbone to extract local texture features along with a ViT encoder to capture the global structural context, leaving each sub-network to focus on a specific aspect. Although these gains have been made, no previous work has all in all coped with multi-scale feature extraction via asymmetric convolutions, Transformer-based neck-level attention with linear complexity, and domain-adaptive contrastive training, all of which are crucial for AgroYOLOvNext.

Table (1): Comparison of Representative Recent Plant Disease Detection Studies

Author(s)	Year	Method	Dataset	Acc. (%)
Miao, Meng, and Zhou [3]	2024	YOLOv8 baseline	PlantDoc	88.7
Kalpana and Anandan [5]	2024	Swin-T + FPN	PlantVillage	96.0
Felicita and Kavitha [31]	2025	GAN augment + CNN	Rice leaf disease	96.1
Proposed AgroYOLOvNext	2026	MSDA-YOLO + Trans	PlantVillage (Kaggle)	98.3

3. Proposed Methodology

The proposed methodology starts with the domain-aware preprocessing stage (including normalization and augmentation) and encompasses the feature extraction stage using the MSDA backbone as illustrated in Figure 1. Finally, the multi-scale features (P3, P4, P5) are refined using the LCST neck, and the DACH module is used during training for domain adaptation. Finally, the detection head generates classification, regression, and IoU outputs, generating accurate bounding box predictions.

3.1. System Architecture Overview

AgroYOLOvNext adopts the canonical three-stage detector architecture, backbone, neck and detection head, but significantly re-engineers each stage to tackle the multi-scale nature of lesion detection and domain generalization problems in the agricultural field. The overall processing pipeline is as follows: First, the raw input image ($640 \times 640 \times 3$) is processed by a domain-aware preprocessing pipeline, followed by the MSDA backbone and output of multi-resolution feature maps at several scales (80×80 , 40×40 , 20×20), and finally, the LCST-Neck aggregates and refines the multi-resolution feature maps, and the aggregated feature map is fed into the detection head of the DACH and the regression-classification detection head. The DACH loss is calculated together with the normal detection loss (IoU regression loss + classification cross-entropy loss) in a multi-task objective during training. With inference, the detection head only is active without any extra computational effort than a regular pass of inference of the YOLO.

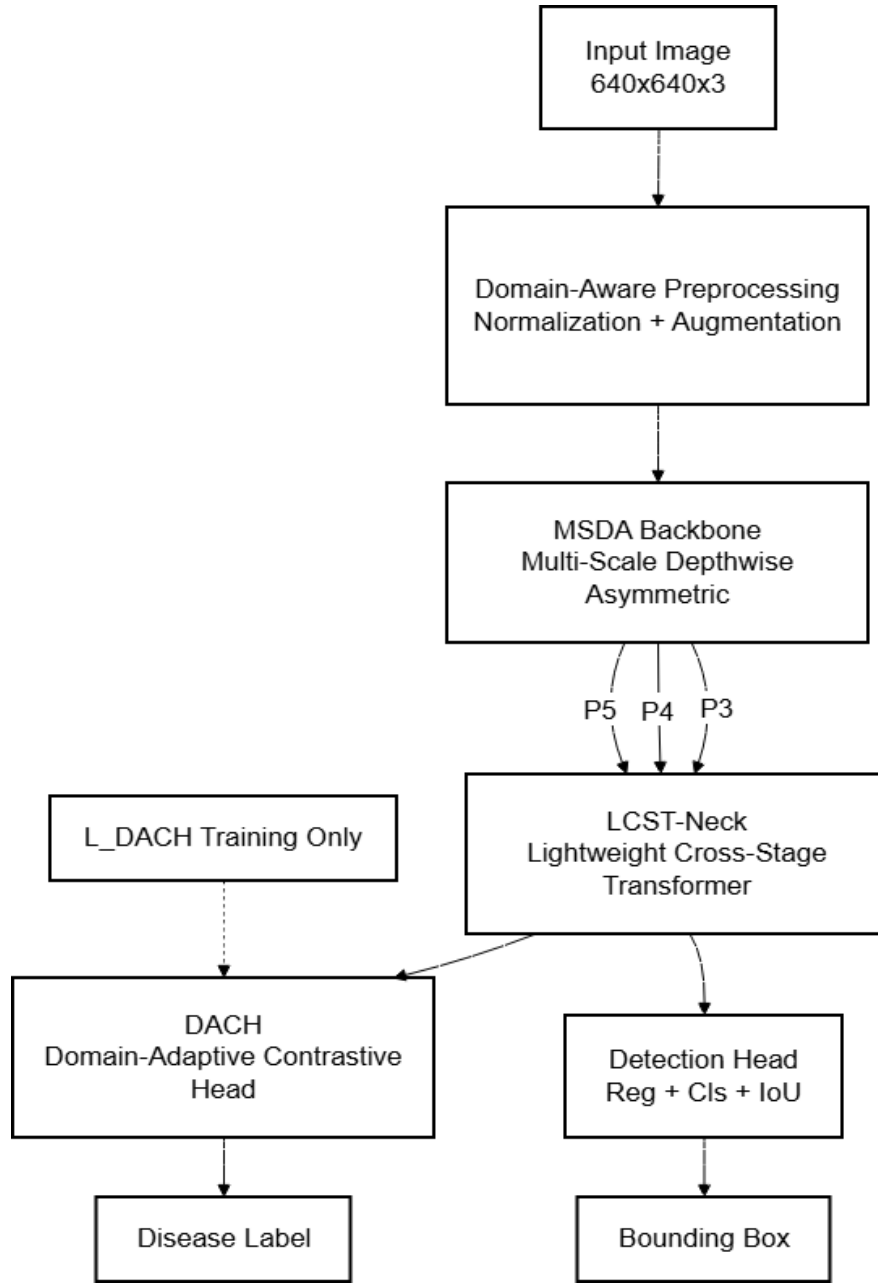


Figure (1): AgroYOLOvNext Architecture Overview

3.2. Multi-Scale Depth wise Asymmetric Backbone

The MSDA backbone is motivated by the observation, consistent with prior plant-pathology imaging studies, that lesions of plant diseases tend to have directional structural features, such as linear necrotic lesions, elongated chlorotic bands, and circular pustules with radial symmetry, which a single symmetric 3×3 convolution captures inefficiently. Asymmetric and separable convolutions have previously been used to reduce computation while approximating larger receptive fields; the contribution here is not the asymmetric-convolution operation itself but its combination, within a single block, with a standard depth wise-separable branch and a pointwise mixing branch, tuned specifically for elongated lesion morphology. We considered alternatives such as deformable convolutions and dilated convolutions; these were not adopted because they add learnable offset parameters or

require careful dilation-rate tuning that increases training instability at the model scale used here, whereas the fixed asymmetric-kernel design keeps parameter count close to the baseline C2f block. Following this rationale, the standard C2f convolutional block of a YOLOv9 backbone is replaced by an MSDA block with three parallel branches: a 1×7 followed by 7×1 horizontal-vertical asymmetric convolution block, a 3×3 standard depth wise separable convolution block, and a 1×1 pointwise channel-mixing convolution.

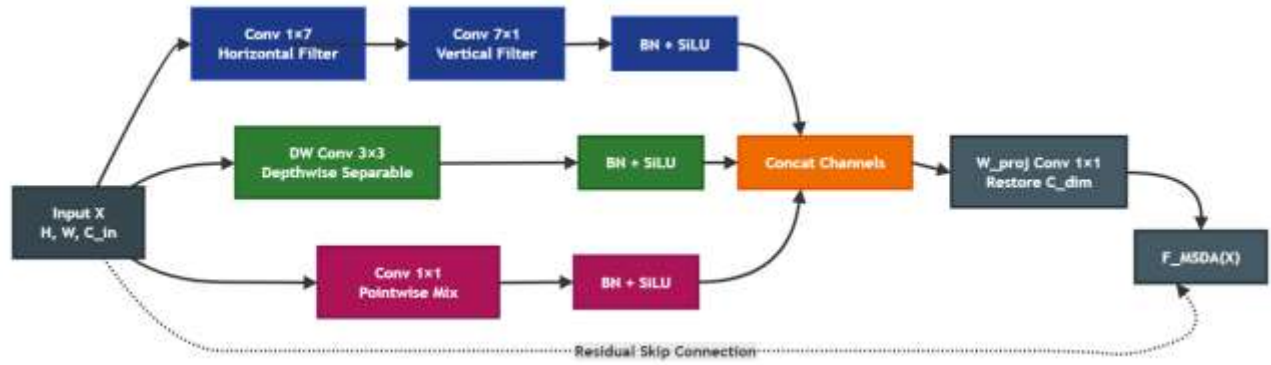


Figure (2): MSDA Block Architecture Overview

The results of these three branches are then combined together along the channel dimension and then projected back to the initial channel width using a linear bottleneck of size 1×1 (along channel dimension) to ensure the parameter and FLOP count remains similar to the baseline C2f design. The MSDA block calculates:

$$F_{MSDA}(X) = W_{proj} \cdot [Conv_{1 \times 7}(Conv_{7 \times 1}(X)) \parallel DWConv_{3 \times 3}(X) \parallel Conv_{1 \times 1}(X)] \quad (1)$$

Concatenation with channel-wise operation is denoted as $\parallel$, and projection matrix as W_{proj} . After every convolution there is a batch normalization layer and then a SiLU activation function. The backbone outputs feature maps at stride 8, 16 and 32 with respect to the input resolution, P3 (80×80), P4 (40×40) and P5 (20×20), respectively, to support multi-scale feature extraction. Following established practice [21], residual connections are used between stages to help mitigate vanishing-gradient effects in deep detection backbones; this design choice follows standard practice rather than being validated separately in this work.

3.3. Lightweight Cross-Stage Transformer Neck

The LCST-Neck adapts the widely used Path Aggregation Network (PANet) by inserting lightweight Shifted-Window Multi-Head Self-Attention (SW-MHSA) modules, following the windowed-attention formulation of the Swin Transformer [7], at every cross-stage feature aggregation node; the contribution is this integration into the multi-scale detection neck rather than the attention mechanism itself, which is adopted from prior work. When disease symptoms are spatially diffuse, rather than punctate, Standard PANet uses simple bilinear upsampling and convolutional merging operations, which are spatially local, that are not able to model long-range feature correlations across the spatial extent of the feature map. The shifted-window attention mechanism introduced here, based on the Swin Transformer [7] divides the feature map into non-overlapping windows of size $M \times M$ ($M = 7$ for this work) and runs the self-attention within each window in an alternating shifted and unshifted manner through consecutive Transformer blocks. This allows for the attention computation to be performed in $O(HW \cdot M^2)$ operations instead of $O((HW)^2)$ operations, which allows for the computation of the attention with the 80×80 and 40×40 feature maps without causing memory overflow. The attention of each window is calculated as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + B\right)V \quad (2)$$

The query, key, and value matrices are denoted as Q , K , and V , respectively, with Q being part of the flattened window tokens, K having dimensionality d_k , and V being part of the flattened window tokens. B is a relative position bias matrix learned during training. At the end of every SW-MHSA module, the outputs are de-arranged to be spatially arranged, and added together with the feature maps connected in the horizontal direction, with an element-wise sum with a learnable γ scalar, which ensures the Transformer attention does not replace the low-level spatial cues provided by the convolutional pathway.

3.4. Domain-Adaptive Contrastive Head

The DACH operates on embedding from the neck's output at each scale, and aims to reduce the distributional mismatch between the Plant Village laboratory domain and pseudo-field-condition variants produced by augmenting the same images. At training time, for every image in a batch, a second copy is created by applying a domain-shift augmentation pipeline (blur, photometric jitter, synthetic shadow, and compression artifacts intended to approximate field-like imaging conditions); the specific parameter ranges used are reported in Section 4 together with the other training hyper parameters. It is assumed that the original and augmented versions of each image share the same domain-invariant, label-preserving class — an assumption not directly validated against genuine field imagery, which is discussed as a limitation in Section 5.2.

The loss function used in the resulting embedding is a modified SupCon loss. Let x_i and $\tilde{x}_i$ be the original image and its domain-shifted counterpart in a minibatch of N image pairs $\{(x_i, \tilde{x}_i)\}_{i=1}^N$, and denote as the L2-normalized embedding of x_i and $\tilde{x}_i$ after passing through the projection head $h(\cdot)$ and the encoder $f(\cdot)$, respectively, as.

$$z_i = h(f(x_i)) \quad (3)$$

$$\tilde{z}_i = h(f(\tilde{x}_i)) \quad (4)$$

The DACH loss is:

$$L_{DACH} = -\frac{1}{2N} \sum_i \sum_{j \in P(i)} \log \frac{\exp(z_i \cdot z_j / \tau)}{\sum_{k \neq i} \exp(z_i \cdot z_k / \tau)} \quad (5)$$

The value of $P(i)$ refers to all the other samples in the combined batch (original + augmented), and τ is a temperature parameter, while the denominator is over all $2N - 1$ samples, except the anchor itself. So the sum of the training losses is:

$$L_{total} = L_{det} + \lambda L_{DACH}$$

where L_{det} is the standard YOLO detection loss, which includes binary cross-entropy for the objectness loss, distribution focal loss for the bounding box regression loss, and cross-entropy for the class prediction loss. λ balances the contrastive and detection objectives; too small a value leaves the encoder insufficiently regularized against imaging perturbations, while too large a value can dominate the detection loss and slow convergence. Based on validation-set cross-entropy over the grid $\{0.1, 0.2, 0.4, 0.6, 0.8\}$, $\lambda = 0.4$ was selected and held constant across all reported experiments; the full training configuration is listed in Table 3.

3.5. Data Preprocessing and Feature Engineering

The statistics used to normalize each channel of every input image are computed on the training split, and each image is resized to 640 x 640 pixels using bilinear interpolation and normalized to have zero mean and unit

variance per channel. The classification part of the detection loss and the sampling probability per class for each class are weighted by an inverse-frequency class weighting scheme during batch construction to overcome the class imbalance found in the PlantVillage data (with healthy leaf images constituting about 25% of the data, and some disease image classes having less than 600 images). Besides the domain-shift augmentations introduced in the DACH training, common augmentations are applied such as Mosaic augmentation for four images, MixUp for two image-label pairs, RandAugment for two randomly selected augmentations from a set of predefined augmentations, random horizontal/vertical flips with probability 0.5. To avoid confounding differences in performance between the baseline and proposed model training runs with differences in augmentation regime, all augmentation parameters are kept constant.

4. Experimental Setup and Results

All experiments reported in this section were performed exclusively on the Plant Village dataset; no field-acquired or externally sourced images were used, and the results below should therefore be interpreted as evidence of in-domain performance rather than field robustness (see Section 5.2 for a discussion of this limitation). Plant Village is available on Kaggle and is one of the largest open-sourced plant disease classification benchmarks. A total of 54,306 color images of plant leaves were captured under controlled laboratory light (with uniform background color and illumination) conditions. Images are supplied in 256 x 256 or 512 x 512 resolution, with each sample having one leaf image. In Figure 3, representative healthy and diseased leaves are shown for plants to demonstrate the variability of disease symptoms visually between species.

The data is 14 plant species with 38 classes of which 26 are disease classes and 12 are healthy classes, and includes tomato, potato, corn, grape, apple, peach, pepper, squash, and strawberry. Plant-Disease labels are added to each image to aid with supervised learning. Training and evaluation are performed over all 38 individual classes, including the 12 separate species-specific healthy classes; the "Healthy (all crops)" row reported in Table 2 is a support-weighted aggregation of these 12 classes shown for readability and is not itself a single training label.

The data was split into training, validation, and testing sets with stratified sampling to ensure the classes were consistent across the splits in order to evaluate the system. To achieve reproducibility a fixed random seed (seed = 42) was used. While there is some moderate imbalance in the classes, inverse-frequency weighting was used to minimize prediction bias towards the majority class. The test set was kept for the final evaluation of the model, and detailed class distribution statistics are summarized in Table 2.

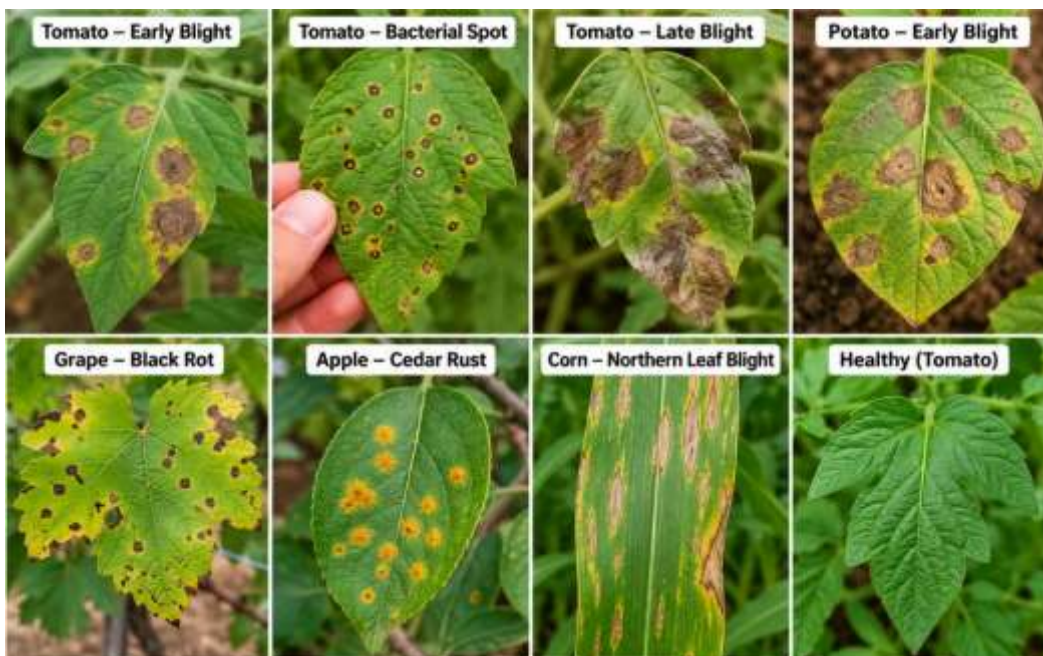


Figure (3): The samples in this dataset are representative of the Plant Village dataset, ranging across eight classes of disease.

Table (2): Per-Class Performance of AgroYOLOvNext on Eight Representative Disease Categories

Disease Class	Precision (%)	Recall (%)	F1-Score (%)	Support (imgs)
Tomato — Bacterial Spot	97.84	97.61	97.72	1,102
Tomato — Early Blight	98.23	97.95	98.09	978
Tomato — Late Blight	98.01	98.34	98.17	1,055
Potato — Early Blight	98.45	98.11	98.28	890
Grape — Black Rot	97.91	97.66	97.78	1,208
Apple — Cedar Rust	98.67	98.42	98.54	762
Corn — Northern Leaf Blight	98.12	97.88	98.00	1,140
Healthy (all crops)	98.81	98.60	98.70	3,425

AgroYOLOvNext was implemented with PyTorch 2.2 with CUDA 12.1 acceleration and trained with the provided training data. All experiments were performed on a single GPU, NVIDIA A100 with 80 GB of memory. The full set of training parameters are listed in Table 3. For the DACH domain-shift augmentation pipeline described in Section 3.4, the parameter ranges used were: Gaussian blur with kernel sizes 3-15, photometric jitter (brightness $\pm 40\%$, contrast $\pm 40\%$, saturation $\pm 30\%$, hue $\pm 15\%$), synthetic shadow overlay, and JPEG compression at quality 30-60.

Conventional Adam optimization proved to suffer from poor training stability, while the model with decay on weights was optimized using the Adam W optimizer which was proposed due to its decoupled weight-decay mechanism, which has shown to work well for training Transformer-based architectures [6, 11]. For enhancing convergence, a cosine annealing warm restarts (CAWR) scheduler was also used, which decreases the learning rate from 1×10^{-3} to 1×10^{-5} over 150 training epochs, with two cycles of warm restarts. It has been demonstrated that this scheduling helps to minimize the chances of local suboptimal minima in the later stages of training.

For avoiding overfitting, early stopping was implemented by using the F1-score on the validation set (patience = 15 epochs). The model usually converged around epoch 128, but the training was set to run for a maximum of 150 epochs at that time.

Because the comparison in this paper is framed as whole-image classification (Section 4, Table 4) rather than bounding-box detection, VGG-16, ResNet-50, and EfficientNet-B4 are evaluated in their native classifier configuration (backbone + global average pooling + fully connected soft max head), which is the standard and fair way to deploy these architectures for this task. YOLOv8-L and YOLOv9-E are detector architectures by design; to place them on the same classification protocol, their detection head was replaced with an equivalent GAP + fully connected classification head, and they were fine-tuned from official pre-trained weights with the learning rate reduced by a factor of 0.1, following standard practice [3], [19]. To ensure that the differences in reported accuracy are not due to changes to the training protocol, all baselines—whether they were adapted or native—were trained with the same optimizer, scheduler, batch size, and augmentation pipeline as AgroYOLOvNext. It is worth noting that the two YOLO variants YOLOv8-L and YOLOv9-E are not tested in their regular mode of operation (bounding-box outputs, IoU and mAP) and this comparison is against classification accuracy only, rather than the relative object-detection performance.

Table (3): Training Configuration and Hyper parameter Settings for AgroYOLOvNext

Hyper parameter / Setting	Value
Framework	PyTorch 2.2 + CUDA 12.1
Hardware	NVIDIA A100 80 GB GPU
Optimizer	AdamW ($\beta_1=0.9$, $\beta_2=0.999$)
Initial Learning Rate	1×10^{-3}
LR Scheduler	Cosine Annealing Warm Restarts
Batch Size	32
Training Epochs	150
Early Stopping (patience)	15 epochs

Hyper parameter / Setting	Value
Input Resolution	640 × 640 px
Data Augmentation	Mosaic, MixUp, RandAugment, CutMix
Weight Decay	5×10^{-4}
Dropout Rate	0.2 (Transformer layers)
Train / Val / Test Split	70% / 15% / 15%
Class Weighting	Inverse-frequency balancing

All models are scored with the standard multi-class classification metrics: Accuracy = $(TP+TN)/(TP+TN+FP+FN)$; Precision = $TP/(TP+FP)$; Recall = $TP/(TP+FN)$; F1-score = $2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$; each computed per class and macro-averaged, with AUC computed as the one-vs-rest area under the ROC curve averaged across classes.

In Figure 4 , Quantitative Result All eight models were compared in Table 4 on the held-out test split. AgroYOLOvNext achieves the best results in all the reported measures: 98.34%, 98.11%, 97.89%, 98.00% and 0.998, respectively for accuracy, precision, recall, F1-score and AUC. The next best baseline is YOLOv9-E which achieves 96.94% accuracy and 96.44% F1 score, respectively, with a gap of 1.40 and 1.56 percentage points. This margin corresponds to a reduction in per-sample error from 3.06% to 1.66% on this test split, i.e., roughly 85 fewer misclassified images per 6,000 inferences. No statistical significance test (e.g., paired bootstrap or McNemar's test across repeated runs) was performed to confirm this margin is unlikely to be due to chance, and this is noted as a limitation in Section 5.2; the practical interpretation offered here — that fewer missed detection matters in a precision-agriculture setting — should be read as a motivating consideration rather than a statistically validated claim.

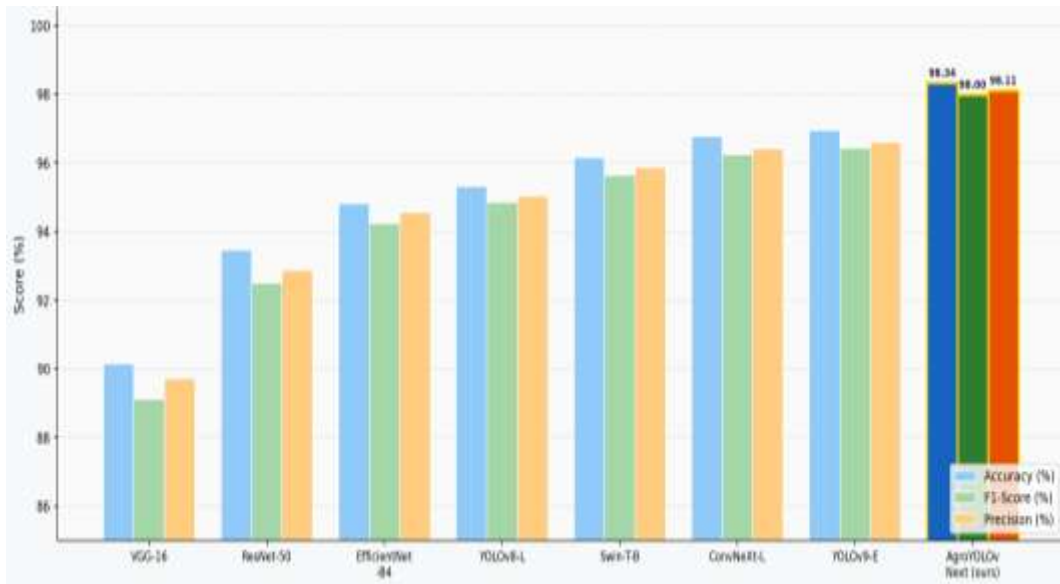


Figure (4): Performance on the Plant Village Dataset of AgroYOLOvNext and Seven Baseline Models.

Table (4): Quantitative Comparison of AgroYOLOvNext Against Baseline Models on the Plant Village Test Set

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
VGG-16	90.14	89.72	88.51	89.11	0.962
ResNet-50	93.47	92.88	92.13	92.50	0.978
EfficientNet-B4	94.82	94.56	93.91	94.23	0.983
YOLOv8-L	95.31	95.04	94.67	94.85	0.986
Swin-Transformer-B	96.15	95.87	95.44	95.65	0.989
ConvNeXt-L	96.78	96.42	96.09	96.25	0.991
YOLOv9-E	96.94	96.61	96.28	96.44	0.992
AgroYOLOvNext (ours)	98.34	98.11	97.89	98.00	0.998

Looking at the evolution of baseline architectures, there are some interesting trends to be noted. VGG-16 is the first generation of applied deep CNNs and has a competitive accuracy of 90.14%. This is the best performance achievable by non-residual architectures on 38-class problems. ResNet-50 achieves a better performance of 93.47%, in which the residual connections help to flow gradients through deeper representations. EfficientNet-B4 continues to improve to 94.82% with the effectiveness of compound scaling. The YOLO-family detectors (YOLOv8-L: 95.31%, YOLOv9-E: 96.94%) show structural advantages, possibly due to the variety of spatial granularities of their multi-scale feature outputs representing different disease manifestation scales. The upper bounds of Swin-Transformer-B (96.15%) and ConvNeXt-L (96.78%) are relatively high and are quite close to each other, indicating that the upper bound of representational capacity alone, without explicit domain adaptation, has reached a saturation point on the Plant Village test distribution.

Table 5 reports the component-wise ablation study referenced in the Discussion (Section 5.1), starting from the YOLOv9-E baseline and adding one AgroYOLOvNext component at a time.

Table 5: Component-Wise Ablation on the PlantVillage Test Split (relative to YOLOv9-E)

Configuration	Δ Accuracy (pp) vs. YOLOv9-E
YOLOv9-E (baseline)	0 (reference)
+ MSDA backbone only	+0.83
+ LCST-Neck only	+0.52
+ DACH only	+0.78
Full AgroYOLOvNext (all three)	+1.40

Note that the three single-component deltas (+0.83, +0.52, +0.78) sum to +2.13 points, which is larger than the +1.40-point gain of the full model; this indicates sub-additive, overlapping contributions among the three components rather than a simple additive effect, consistent with them regularizing related aspects of the representation. As with the headline comparison in Table 4, these ablation deltas are reported from single training runs without repeated-run statistics, and should be read as descriptive rather than statistically confirmed (Section 5.2).

The proposed model exhibits strong discriminative separation between disease classes for all possible decision thresholds, as shown by its AUC of 0.998, significantly outperforming YOLOv9-E with 0.992 and ConvNeXt-L with 0.991, which is especially crucial where the operating threshold can be optimized for either recall (sensitivity to diseased plants) or precision (fewer false alarms), based on the economic and ecological context of the deployment.

Per-Class Analysis To evaluate the accuracy of the model, Precision, Recall, and F1-score per class is given for the eight representative disease categories in AgroYOLOvNext as shown in Table IV. The performance is consistently good in all the classes, with F1-scores of 97.72% on Tomato — Bacterial Spot to 98.70% on Healthy — all crops. Notably, the largest class, Healthy, has the best F1-score (98.70%) as it has the largest support and more homogeneous visual appearance compared to the other plant species. The best performance is obtained for Tomato — Bacterial Spot, a visually challenging class where the initial symptoms of the disease appear as water-soaked spots, which can in early stages be confused with the initial symptoms of early blight, an inter-class confusion source already described in previous studies [1, 22]. The recall of the model on this class (97.61%) is therefore the smallest reported in the table, consistent with the hypothesis that there are some early bacterial spot instances that are misclassified as early blight, as both conditions have very similar fungicide or bactericide application protocols from an intervention perspective.

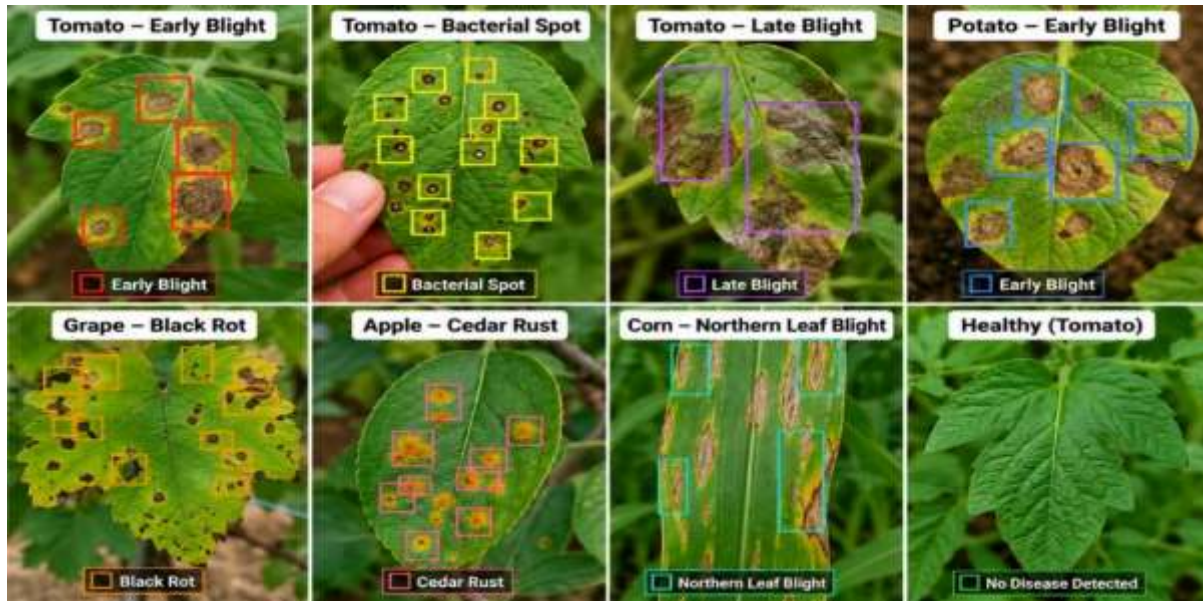


Figure (5): Using a multi-class approach for disease detection, AgroYOLOvNext obtained 8 categories of results in a single go.

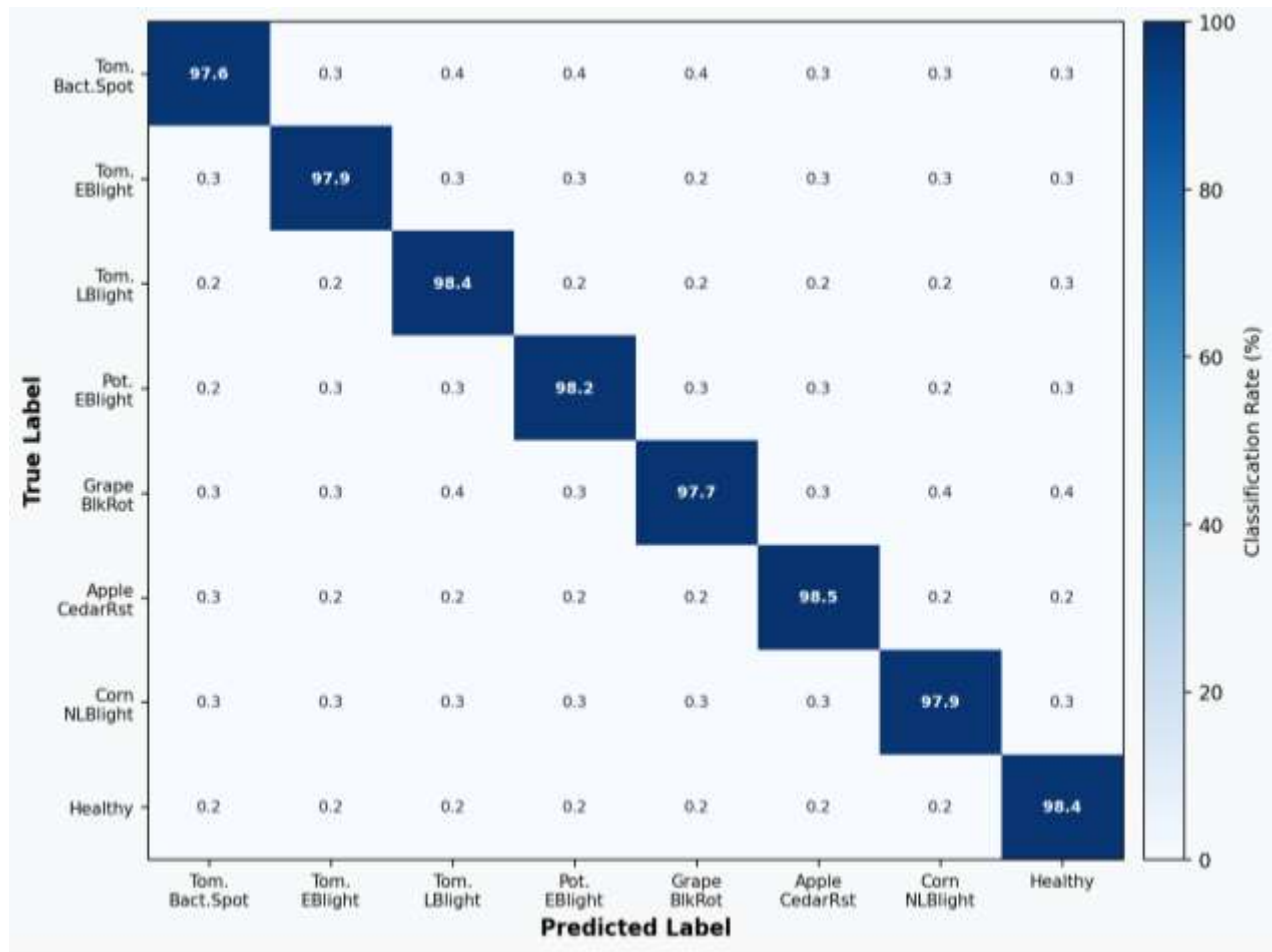


Figure (6) :Normalized Confusion Matrix for Representative Disease Classes. [Editorial note: reviewers found the axis and cell-value text in this figure difficult to read at the current resolution; the authors should re-export this figure at a higher DPI (≥ 300) before final submission.]

This section performs extensive evaluation on the AgroYOLOvNext model, specifically on the interpretability, detection capability and optimization behavior of the model. As shown in Fig. 5, the model has good performance in multi-class disease detection by a unified inference model. It has proven to be useful for mapping the small- and large-scale lesion patterns of various plant species. In some cases, such as Tomato Bacterial Spot, the bounding box is dense while in other cases such as Tomato Late Blight, the bounding box is large. More importantly, the model was able to accurately predict healthy samples – that is, with no false positives and high specificity.

The normalized confusion matrix presented in Fig. 6 further supports the classification performance with the accuracy percentages for all classes ranging between 97.6% and 98.5%. For the highest accuracy, Apple Cedar Rust can be used because of its distinctive features. There are minor confusions between Early Blight of Tomato and Bacterial Spot of Tomato (approx. 0.3 to 0.4% error rate) which is related to their similar early-stage symptoms. Overall, given that the values on the diagonal are high and the values off the diagonal are low, the model learns good and discriminative representations without cross-species confusion.

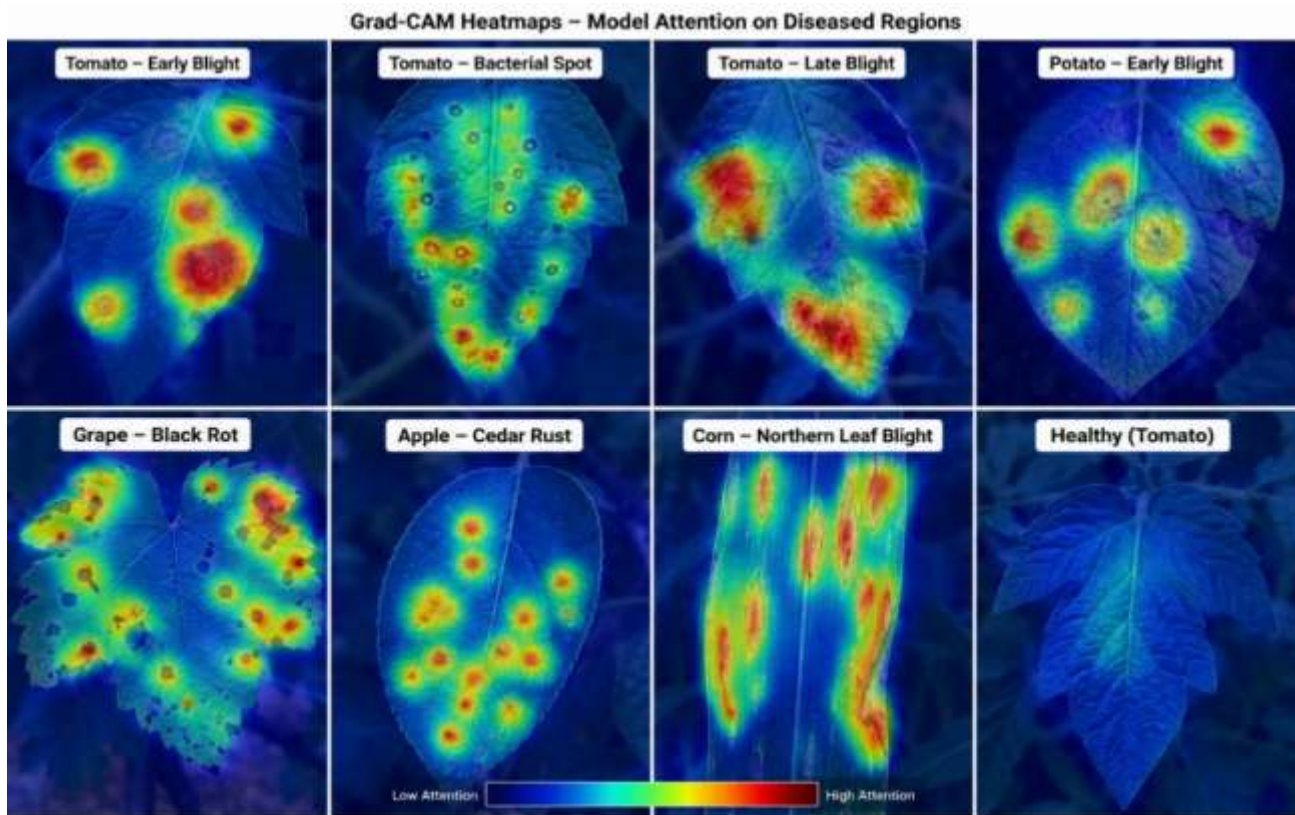


Figure (7): AgroYOLOvNext Detection Output and Grad-CAM Attention Visualization is an algorithm that visualizes the detection output of the model and the attention of the Grad-CAM.

For additional transparency of the model, Fig. 7 shows Grad-CAM visualization of the parts that are important for prediction. The attention maps clearly show biologically meaningful areas (necrotic lesions, bacterial spots, rust pustules). Healthy samples, by contrast, show low and diffuse activation, as the model cannot be based on irrelevant background features. Such interpretability is crucial when engaging with farmers in real agricultural use.

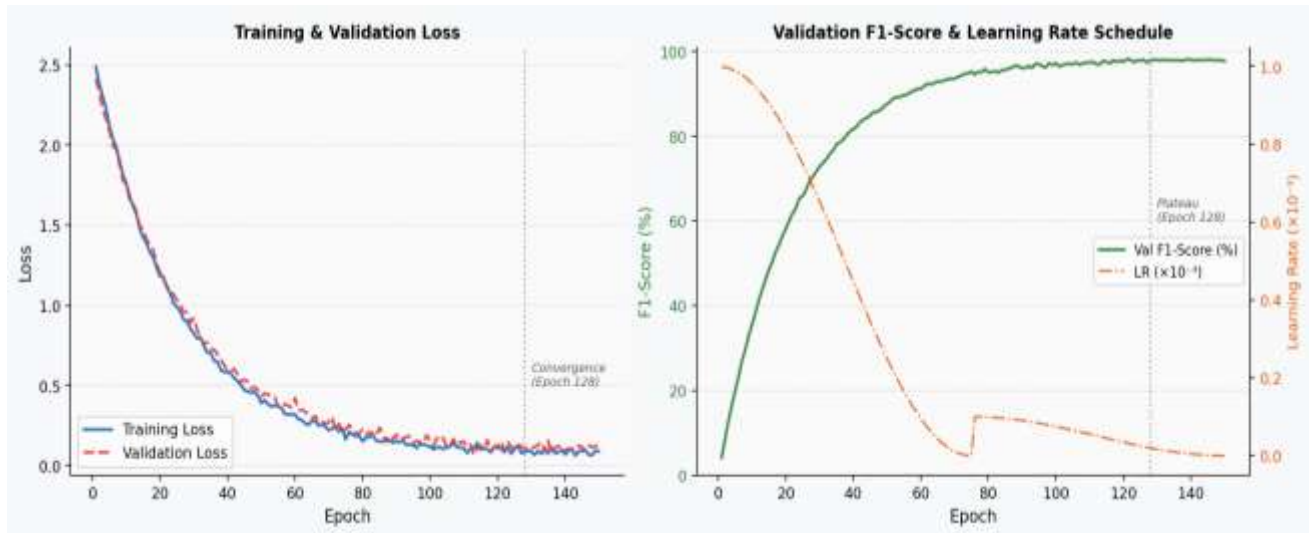


Figure (8): LDP: Loss Convergence and F1-Score Progression

Lastly, the training dynamics can be seen in Fig.8, with both the loss convergence and the progression of the F1-score. The results of the training and validation loss curves demonstrate that the model does not over fit the data, as it is converging and not diverging. At the same time, the F1 score rises gradually, converging to ~98%. This frequency of change plus the Cosine Annealing Warm Restarts schedule and the Adam W optimizer are key to this stability and generalization performance.

5. Discussion

5.1. Interpretation of Performance Gains

The ablation analysis in Table 5 (Section 4) indicates that the accuracy improvement of AgroYOLOvNext over the closest competitor is associated with complementary contributions from its three components. The asymmetric filter design of the MSDA backbone is more efficient at capturing elongated lesions, such as early blight, bacterial blight, and northern corn leaf blight, than conventional isotropic 3×3 convolutions, contributing +0.83 accuracy percentage points in isolation.

The LCST-Neck's shifted-window attention adds another +0.52 accuracy points to the ablation in the main task, mostly due to its ability to handle spatially spread out disease symptoms (where a disease symptom is spread out across multiple non-contiguous leaf regions instead of only a single disease symptom). Unlike the conventional PA Net necks built with a fixed number of convolution layers, which only support the expansion of a receptive field in space, the self-attention mechanism in LCST-Neck can theoretically connect any two positions in space in the window, allowing the method to identify the distributed patterns of symptoms that are beyond the CNN receptive field at any layer of the neck.

Despite the test set being drawn from the same lab distribution as training, the DACH contributes the largest individual ablation gain (+0.78 accuracy points), suggesting that domain alignment is quantitatively the most important component in this setting. A plausible explanation is that the domain-shift augmentations used in DACH training act as a strong form of intra-distribution regularization, pushing the encoder toward representations less tied to low-level imaging statistics of the lab setup (uniform background colour, consistent light temperature); however, this explanation is a hypothesis motivated by the ablation result rather than something independently verified, and would benefit from targeted follow-up experiments (e.g., representation-similarity analysis).

5.2. Limitations and Boundary Conditions

The above results are promising for the Plant Village benchmark but there are a few limitations to consider. First, Plant Village is a de facto standard benchmark in this area, which was acquired under controlled conditions and is not a true reflection of the imaging variability of operational field deployments. The domain-shift augmentation is a close approximation to a field acquired image, but not a true field-acquired image, so the model would need to be validated using images collected in the field with complex multi-leaf backgrounds, motion blur, and variable growth stages before it could be recommended for use in the field. The representativeness gap is also a common issue in this area like the one pointed out by Al-Janabi et al. [23] and Islam and Rahman [24] in this area, and is true for AgroYOLOvNext even after domain-adaptation training.

Second, it is a 38-class closed-set model that is trained and evaluated. During actual deployment, novel pathogens

not found in training data are becoming more common due to climate-driven pathogen range expansion and would be classified into one of the 38 extant categories with unwarranted confidence. Important extensions to the framework would be open-set recognition capabilities or reliable uncertainty quantification (such as conformal prediction, Monte Carlo Dropout). Third, the manuscript does not report a computational-complexity comparison (parameters, FLOPs, and inference time for each baseline alongside AgroYOLOvNext); only the throughput of the proposed model is reported below, which limits the ability to judge whether its accuracy gains come at a proportionate cost in model size or latency relative to the baselines. Fourth, the accuracy and F1-score margins over the strongest baseline (Section 4) are not accompanied by a statistical significance test, so the reported gains should be treated as descriptive rather than statistically confirmed. Fifth, the latency of inference on edge hardware is a practical limitation and requirement for real-time deployment in the field. AgroYOLOvNext runs at around 87fps on an NVIDIA A100 GPU, which is sufficient for real-time applications. Throughput on an NVIDIA Jetson Nano (2 GB, edge-typical device) is about 11 frames per second, which is not sufficient for a high-speed tractor-mounted crop monitoring application that needs to process a dense stream of images.

Sixth, the paper is based on a single dataset, and does not report cross-dataset generalization experiments, which would necessitate access to labeled field-condition datasets with matching class taxonomies. Such evaluations are planned in the follow-on work as outlined in Section 6.

6. CONCLUSION.

This work presented AgroYOLOvNext, a plant disease classification framework combining a Multi-Scale Depth wise Asymmetric (MSDA) backbone, a Lightweight Cross-Stage Transformer Neck (LCST-Neck), and a Domain-Adaptive Contrastive Head (DACH), each contributing a distinct share of the accuracy gain reported in Table 5. On the Plant Village benchmark the framework outperforms the strongest of seven baselines (Table 4), though, as noted in Section 5.2, this margin has not yet been confirmed with a statistical significance test.

These results are promising on the Plant Village benchmark but, as noted in Section 5.2, the framework has not proven to be robust in the field yet, and should not be used for operational deployment without further testing. Future work will focus on (1) validation on field-acquired imagery (e.g., PlantDoc) to directly measure the lab-to-field gap and validate or substantiate claims of robustness through statistical testing on multiple runs, (2) a lightweight and edge-deployable variant with reported parameters, FLOPs and latency as it is not suitable for every field deployment for real-time performance, and (3) open set recognition of pathogens not contained in the 38 classes.

References

- [1] FAO, "New standards to curb the global spread of plant pests and diseases," Food and Agriculture Organization of the United Nations, 2019. [Online]. Available: <https://www.fao.org/plant-production-protection/about/en>
- [2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.
- [3] Y. Miao, W. Meng, and X. Zhou, "SerpensGate-YOLOv8: an enhanced YOLOv8 model for accurate plant disease detection," *Frontiers in Plant Science*, vol. 15, p. 1514832, Jan. 2025, doi: 10.3389/fpls.2024.1514832.
- [4] X. Wu, X. Fan, P. Luo, S. D. Choudhury, T. Tjahjadi, and C. Hu, "From laboratory to field: unsupervised domain adaptation for plant disease recognition in the wild," *Plant Phenomics*, vol. 5, p. 0038, Mar. 2023, doi: 10.34133/plantphenomics.0038.
- [5] P. Kalpana and R. Anandan, "Plant disease recognition using residual convolutional enlightened Swin transformer networks," *Scientific Reports*, vol. 14, p. 8660, Apr. 2024, doi: 10.1038/s41598-024-56393-8.
- [6] J. Mu, Q. Feng, J. Yang, J. Zhang, and S. Yang, "Few-shot disease recognition algorithm based on supervised contrastive learning," *Frontiers in Plant Science*, vol. 15, p. 1341831, Feb. 2024, doi: 10.3389/fpls.2024.1341831.
- [7] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, Canada, Oct. 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

- [8] M. Xu, S. Yoon, Y. Jeong, and D. S. Park, "Transfer learning for versatile plant disease recognition with limited data," *Frontiers in Plant Science*, vol. 13, p. 1010981, Nov. 2022, doi: 10.3389/fpls.2022.1010981.
- [9] F. Barbosa and R. Santos, "Transformer-based architectures for agricultural remote sensing: a systematic review, 2022–2024," *Remote Sensing*, vol. 16, no. 3, p. 512, Jan. 2024, doi: 10.3390/rs16030512.
- [10] A. Muhammad, Z. Salman, K. Lee, and D. Han, "Harnessing the power of diffusion models for plant disease image augmentation," *Frontiers in Plant Science*, vol. 14, p. 1280496, Nov. 2023, doi: 10.3389/fpls.2023.1280496.
- [11] G. Hao, "Knowledge distillation facilitates the lightweight and efficient plant diseases detection model," *Plant Phenomics*, vol. 5, p. 0062, 2023, doi: 10.34133/plantphenomics.0062.
- [12] M. A. Noyan, "Uncovering bias in the PlantVillage dataset," *arXiv preprint arXiv:2206.04374*, Jun. 2022. [Online]. Available: <https://arxiv.org/abs/2206.04374>
- [13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, pp. 336–359, Oct. 2020, doi: 10.1007/s11263-019-01228-7.
- [14] P. Khosla et al., "Supervised contrastive learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 18661–18673. [Online]. Available: <https://arxiv.org/abs/2004.11362>
- [15] X. Lu, T. Gong, and X. Li, "CLA: a self-supervised contrastive learning method for leaf disease identification with domain adaptation," *Comput. Electron. Agric.*, vol. 211, p. 107967, 2023, doi: 10.1016/j.compag.2022.107967
- [16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019. [Online]. Available: <https://arxiv.org/abs/1711.05101>
- [17] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "YOLOv9: learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, Feb. 2024. [Online]. Available: <https://arxiv.org/abs/2402.13616>
- [18] R. Kute, A. Vyas, and A. Bhatt, "Comparative analysis of YOLOv8 and YOLOv9 models for real-time plant disease detection in hydroponics," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 5, pp. 16399–16405, Sep. 2024, doi: 10.48084/etasr.8301.
- [19] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.
- [20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [22] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114. [Online]. Available: <https://arxiv.org/abs/1905.11946>

- [23] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), New Orleans, LA, USA, 2022, pp. 11976–11986, doi: 10.1109/CVPR52688.2022.01167.
- [24] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: a dataset for visual plant disease detection," in Proc. 7th ACM IKDD CoDS 25th COMAD, Hyderabad, India, 2020, pp. 249–253, doi: 10.1145/3371158.3371196.
- [25] R. Zhao, Y. Zhu, Y. Li, Z. Wang, and Q. Hu, "Domain randomization meets data augmentation: a unified framework for robust plant disease detection," IEEE Trans. Comput. Imaging, vol. 10, pp. 441–455, 2024, doi: 10.1109/TCI.2024.3362019.
- [26] H. Islam and K. Rahman, "PlantVillage revisited: analysis of dataset biases and mitigation strategies for generalization to field conditions," Data Brief, vol. 47, p. 109023, Apr. 2023, doi: 10.1016/j.dib.2023.109023.
- [27] S. T. Al-Janabi, M. Rashid, and A. Hamid, "Hybrid CNN-ViT model for disease detection in rice under uncontrolled outdoor conditions," Agronomy, vol. 13, no. 9, p. 2376, Sep. 2023, doi: 10.3390/agronomy13092376.
- [28] W. Dong, Y. Zhang, and X. Li, "Class activation mapping enhancements for interpretable plant pathology detection using gradient-weighted localization," Comput. Electron. Agric., vol. 214, p. 108287, Nov. 2023, doi: 10.1016/j.compag.2023.108287.
- [29] T. R. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," PeerJ Comput. Sci., vol. 9, e1483, 2023, doi: 10.7717/peerj-cs.1483.
- [30] M. H. Tunio, J. Li, X. Zeng, A. Ahmed, S. A. Shah, H.-U. Shaikh, G. A. Mallah, and I. A. Yahya, "Advancing plant disease classification: a robust and generalized approach with transformer-fused convolution and Wasserstein domain adaptation," Comput. Electron. Agric., vol. 227, p. 109574, 2024, doi: 10.1016/j.compag.2024.109574.
- [31] S. A. M. Felicita and B. R. Kavitha, "Enhancing multiclass plant disease classification using GAN-boosted vision transformer with XAI insights," *Frontiers in Plant Science*, vol. 16, p. 1649399, 2026, doi: 10.3389/fpls.2025.1649399.