



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Automated Plant Disease Detection via ConvNeXt-Based Deep Learning and Progressive Transfer Learning

Pushpa Patil¹, Shilpa Kaman², Girish K Kulkarni³, Shashidhar B Patil⁴, Swaroop Karpurmath⁵, Bhimappa M⁶, Prashant V Elagi⁷

¹Department of Computer Science and Engineering (Data Science), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India. E-mail: pushpapatil2008@gmail.com

²Department of Computer Science and Engineering, Walchand Institute of Technology, Solapur-413006, India. E-mail: shilpakaman16@gmail.com

³Department of Computer Applications (MCA), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India. E-mail: kulkgirish2000@gmail.com

⁴Department of Computer Applications (MCA), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India. E-mail: shashidhar.b.patil@gmail.com

⁵Department of Computer Science and Engineering (Data Science), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India.

⁶Department of Computer Science and Engineering (Data Science), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India.

⁷Department of Computer Science and Engineering (Data Science), BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura-586101, India.

Abstract

Plant diseases cause the losses of up to 40% of global harvests and more than 220 billion USD to the world economy annually. These losses primarily affect small-scale farmers, who typically lack the resources to obtain expert-level plant disease diagnosis, highlighting the need for timely and reliable detection mechanisms. We propose an automated plant disease detection system based on the computer vision paradigm and ConvNeXt Base architecture that incorporates state-of-the-art hybrid transformer designs into the convolutional neural network through progressive fine-tuning. Our model achieves 94.22% top-1 and 97.83% top-3 accuracy on the custom dataset of 8,250 images of 15 classes (5 crops with 3 classes each) during the final evaluation on the held-out test partition. Additionally, the macro F1-score is 94.17% or 1.412% absolute difference from the second best method in published literature for this task. We also release a web-based interface built with the Streamlit framework that provides diagnostic information, including treatment and prevention guidelines, within one second of input image upload.

Keywords: Plant Disease Detection, Deep Learning, ConvNeXt, Progressive Fine-Tuning, Transfer Learning, Computer Vision, Precision Agriculture, Food Security.

1. Introduction

Billions of people rely on agriculture for food production and derive their livelihood from it, making it one of the pillars of the global economy. Indeed, over forty percent of the world's working-age population is engaged in agriculture, with many low-income countries recording over two-thirds of their labour force in this sector Food and Agriculture Organization of the United Nations (FAO) 31. Nevertheless, the importance of agriculture does not diminish the impact of biotic stress on food production. Fungal, bacterial, viral, and nematode pathogens, for instance, contribute to a quarter of the globe's agricultural losses and destroy nearly forty percent of food crops annually. The value of this damage reaches \$220 billion, with an immediate and severe threat to food security (FAO 31).

Integrated disease management requires early detection and accurate diagnosis of foliar disease to mitigate yield loss. However, visual assessment by plant pathologists or extension officers remains the norm, despite being subjective, inconsistent, and limited in availability. Though laboratory tests such as polymerase chain reaction (PCR) or enzyme-linked immunosorbent assays (ELISA) can confirm the diagnosis with greater objectivity, their high cost limits their use. Smallholder farmers in developing countries are the most affected, despite contributing significantly to global food production. They lack the resources and access to timely and expert agricultural advisory services.

Deep learning technologies can be leveraged to address this challenge. Convolutional neural networks enable the automated diagnosis of foliar diseases with accuracy comparable to that of experts. Combined with the rising pervasiveness of smartphones and cellular networks, hybrid networks that employ CNNs and attention mechanisms can be designed into mobile apps. These applications would offer diagnostic and

advisory services directly to smallholder farmers, bypassing the need for complex and expensive laboratory equipment..

1.1 Problem Statement

A system that is actually fit for practical application must be both effective and computationally efficient; it is not enough to simply demonstrate theoretical accuracy. The main objective of the current project was to design a classifier that would be capable of distinguishing between images of plant leaves affected by any of the 15 possible diseases (or not affected at all) for five different crops.

The system needs to satisfy the following constraints:

- Accuracy of around or above 90% on an image of any of the 15 classes.
- Prediction time of no longer than 2 seconds on average on a standard PC.
- A simple enough interface for the common user who does not have much experience with computers.
- Provide management recommendations for each disease detected.

1.2 Major Contributions

This paper makes the following contributions:

- ConvNeXt-based pipeline utilising dual global pooling with a multi-stage, regularised classification head, achieving top-1 accuracy of 94.5% on a 15-class, multi-crop disease detection benchmark.
- Three-stage progressive fine-tuning protocol, which was experimentally validated to produce consistent accuracy gains while avoiding overfitting on a moderately sized agricultural dataset.
- An extensive model evaluation comprising per-class precision-recall-F1 analysis, ablation study, and edge-case benchmarking.
- Production-grade streamlit-based web application providing sub-second inference and evidence-based disease management recommendations.

2. Related Work

The field of automated plant disease detection has seen a transition from hand-crafted feature engineering to modern vision transformers. This section provides a brief historical perspective and places the current work in context.

2.1 Traditional and Laboratory-Based Diagnosis

Manual visual inspection by trained plant pathologists has been the standard procedure for most of this century's duration. Pathologists examine symptoms such as patterns of chlorosis, necrosis lesion morphology, sporulation, and other characteristic deformities such as wilting or curling, and the accuracy of this method is relatively high (when performed correctly). However, it is labour-intensive, location-dependent, and highly subjective [1]. Microscopic analysis, microbiological isolation and cultivation, serology, and molecular methods such as PCR provide more specific identification but require expensive equipment and are time-consuming (typically taking up to 72 hours for results).

2.2 Machine Learning with Hand-Crafted Features

Before delving into the specifics of deep learning architectures, it is instructive to review the state-of-the-art in machine learning literature. The prevailing approach to plant disease classification before the advent of deep learning used to involve training classical machine learning classifiers such as SVMs and Random Forests on hand-crafted visual features extracted from input images. These included grey-level features based on Gray Level Co-occurrence Matrix and Local Binary Patterns, colour-based features in HSV and CIE Lab colour spaces, and shape-based descriptors such as Fourier-transformed contours. Accuracy rates of approximately 70–85% were typical for the best-performing models, limited in part by the choice of extractor and the visual information it could encode, as well as a lack of invariance to image formation changes such as compression artifacts.

2.3 Deep Learning for Plant Disease Recognition

Fine-tuned CNNs were among the first approaches used for plant disease detection. Specifically, Mohanty et al. [1] were the first to apply them on the PlantVillage dataset, achieving good results on a controlled laboratory setting; however, their accuracy dropped significantly when using field images due to domain shift. The following works improved the quality of the CNN architectures in terms of inductive bias. For example, VGGNet [5] demonstrated the effectiveness of using a large number of stacked convolutional layers. ResNet [13] introduced residual connections, helping to overcome the vanishing gradient problem in very deep networks. Finally, EfficientNet [20] proposed to combine the depth, width, and resolution of the network in a coordinated manner through compound scaling.

Recently, ViT [22] demonstrated that CNNs are not the only way to achieve top accuracy on large-scale image recognition tasks. More specifically, the work proposed to use Vision Transformers, a type of self-attention network, for processing vision data by splitting the input image into patches and feeding them as a sequence to a standard transformer encoder. ConvNeXt [30] builds upon the ideas of ViT [22] and demonstrates that convolutional networks can also benefit from some of the design choices introduced by vision transformers. Specifically, Liu et al. [30] replace standard convolutions with depthwise separable 7×7 convolutions and adopt modules commonly used in vision transformers (i.e., inverted bottleneck

blocks, GELU activation, and layer normalization). The resulting CNN architecture achieves accuracy comparable to ViT while maintaining the inductive bias of CNNs.

2.4 Literature Survey

Table 1 below provides an overview of the most relevant works on plant disease detection from the past years. In particular, 28 studies are listed for the last decade (2016–2024), covering different categories from traditional machine learning approaches to transfer learning and CNNs, as well as the latest state-of-the-art vision transformers. The row of Our method is added for reference.

Table 1: Comprehensive Literature Survey (28 Works).

Ref	Author(s)	Architecture	Dataset	Acc (%)	Year	Key Contribution / Limitation
[1]	Mohanty et al.	AlexNet/GoogLeNet	PlantVillage	99.35	2016	First large-scale CNN; strong lab accuracy; degrades on field images; no deployment
[2]	Brahimi et al.	AlexNet/OxfordNet	PlantVillage	99.67	2017	Fine-grained tomato study; high accuracy under controlled conditions; no field validation
[3]	Ferentinos	Custom CNN	PlantVillage	99.53	2018	Multi-crop CNN; overfitted to lab; no treatment guidance
[4]	Ramcharan et al.	Inception V3	Cassava (field)	93.0	2017	One of few field-image studies; single crop; limited generalisation
[5]	Liu & Wang	AlexNet	PlantVillage	95.48	2017	Image preprocessing pipeline; confined to controlled imaging; no progressive training
[6]	Geetharamani & A.P.	9-Layer Deep CNN	PlantVillage	96.46	2019	Custom 9-layer CNN; lacks transfer learning and dual pooling
[7]	Karthik et al.	ResNet + Attention	PlantVillage	98.1	2020	Attention-augmented ResNet; improved localisation; no web application
[8]	Chen et al.	VGG16 + SVM	PlantVillage	90.4	2020	Hybrid CNN-SVM; limited to VGG features; no progressive fine-tuning
[9]	Jiang et al.	GoogLeNet	PlantVillage	97.29	2019	Multi-scale pooling; good accuracy; lab-only evaluation
[10]	Islam et al.	MobileNetV2	Rice leaf	95.67	2021	Lightweight mobile model; single crop; no treatment recommendations

Ref	Author(s)	Architecture	Dataset	Acc (%)	Year	Key Contribution / Limitation
[11]	Saleem et al.	EfficientNetB0	PlantVillage	94.3	2021	Efficient compound scaling; limited fine-tuning; no dual pooling
[12]	Rangarajan et al.	Inception-ResNet-V2	Tomato	97.8	2020	Dual-architecture for tomato only; no progressive fine-tuning
[13]	Zhang et al.	ResNet-50	PlantVillage	91.4	2021	Residual extraction; single-phase fine-tuning; no ablation
[14]	Agarwal et al.	DenseNet121	PlantVillage	93.5	2022	Dense skip connections; high memory; no deployment tool
[15]	Too et al.	DenseNet/ResNet	PlantVillage	99.7	2019	Benchmarked four architectures; lab-only; no treatment guidance
[16]	Barbedo	Custom CNN	Field images	78.6	2019	Explicit sim-to-real gap study; highlights domain shift problem
[17]	Liang et al.	Capsule Network	PlantVillage	95.27	2020	Capsule nets for spatial encoding; slower inference; no web app
[18]	Tm et al.	AlexNet (scratch)	PlantVillage	87.5	2019	Trained from scratch; no transfer learning; lower accuracy ceiling
[19]	Mia et al.	VGG16	PlantVillage	89.2	2020	Standard VGG fine-tuning; no progressive unfreezing or ablation
[20]	Gajjar et al.	EfficientNetB4	PlantVillage	92.8	2022	Best prior baseline; no dual pooling; no ablation study
[21]	Chen et al.	Swin Transformer	PlantVillage	96.4	2022	ViT-based; strong accuracy; heavy compute; no progressive FT
[22]	Sun et al.	ViT-B/16	PlantVillage	95.6	2022	Vision Transformer; large model; no deployment or treatment
[23]	Wang et al.	ConvMixer	Custom leaf	93.1	2023	Patch-mixing; lacks large-scale benchmark comparison; no ablation

Ref	Author(s)	Architecture	Dataset	Acc (%)	Year	Key Contribution / Limitation
[24]	Singh et al.	ResNet-101	PlantVillage/field	90.7	2023	Field augmentation; standard residual net; no treatment recommendations
[25]	Pawar & Turuk	MobileNetV3	Grape dataset	94.1	2023	Lightweight mobile; single crop; no multi-phase fine-tuning
[26]	Tm & Nair	NASNet-Mobile	Rice/Wheat	91.3	2023	Neural arch search; single domain; no ablation or deployment
[27]	Bhatt & Bhatt	MaxViT	PlantVillage	96.2	2024	Hybrid Max-ViT; high GPU requirement; no application deployment
[28]	Atila et al.	ViT+ConvNeXt Ensemble	PlantVillage	97.1	2024	Ensemble; 2x parameters; no progressive unfreezing; no ablation

2.5 Identified Research Gaps

Five gaps recur across this body of work, and this study was designed to close each one:

1. Evaluations are frequently restricted to laboratory-captured images with no field-domain validation.
2. Deployment and user-interface considerations are seldom addressed; most works deliver only a trained model with no accessible inference pipeline.
3. Classification results are rarely accompanied by actionable disease management guidance.
4. Progressive fine-tuning strategies and component-level ablation studies are absent from most works.
5. Comparative studies across modern architectures on identical benchmark splits remain scarce.

3. Methodology

3.1 Dataset Preparation

Of the 87,848 colour images and 38 disease–crop combinations in the PlantVillage dataset [29], we drew a subset of 15 classes across five economically significant crops. Agronomic impact, disease diversity, and class balance drove that selection, which Table 2 summarises.

Table 2: Selected Disease Classes and Agronomic Justification

Crop (Classes)	Included Conditions	Agronomic Rationale
Tomato (5)	Bacterial spot, Early blight, Late blight, Leaf mould, Healthy	World’s most consumed vegetable; foliar diseases reduce yield by 30–40%; broad pathogen diversity enables multi-pathogen testing
Potato (3)	Early blight, Late blight, Healthy	Staple for over one billion people; <i>Phytophthora infestans</i> caused catastrophic historical famines; significant ongoing economic impact
Corn / Maize (3)	Common rust, Northern leaf blight, Healthy	Third-largest crop globally; rust infections reduce yields by 20–50% in affected fields
Grape (2)	Black rot, Healthy	High-value horticultural crop; <i>Guignardia bidwellii</i> destroys 50–90% of fruit without timely intervention

Crop (Classes)	Included Conditions	Agronomic Rationale
Bell Pepper (2)	Bacterial spot, Healthy	Significant export commodity; bacterial diseases difficult to control chemically, making early detection valuable

Dataset partitioning. Stratified random sub-sampling pulled up to 550 images per class, splitting out to 6,000 training, 1,125 validation, and 1,125 test images (70%/15%/15%). Figures 1 and 2 show, respectively, sample images from all 15 classes and how the dataset is distributed across the three splits.

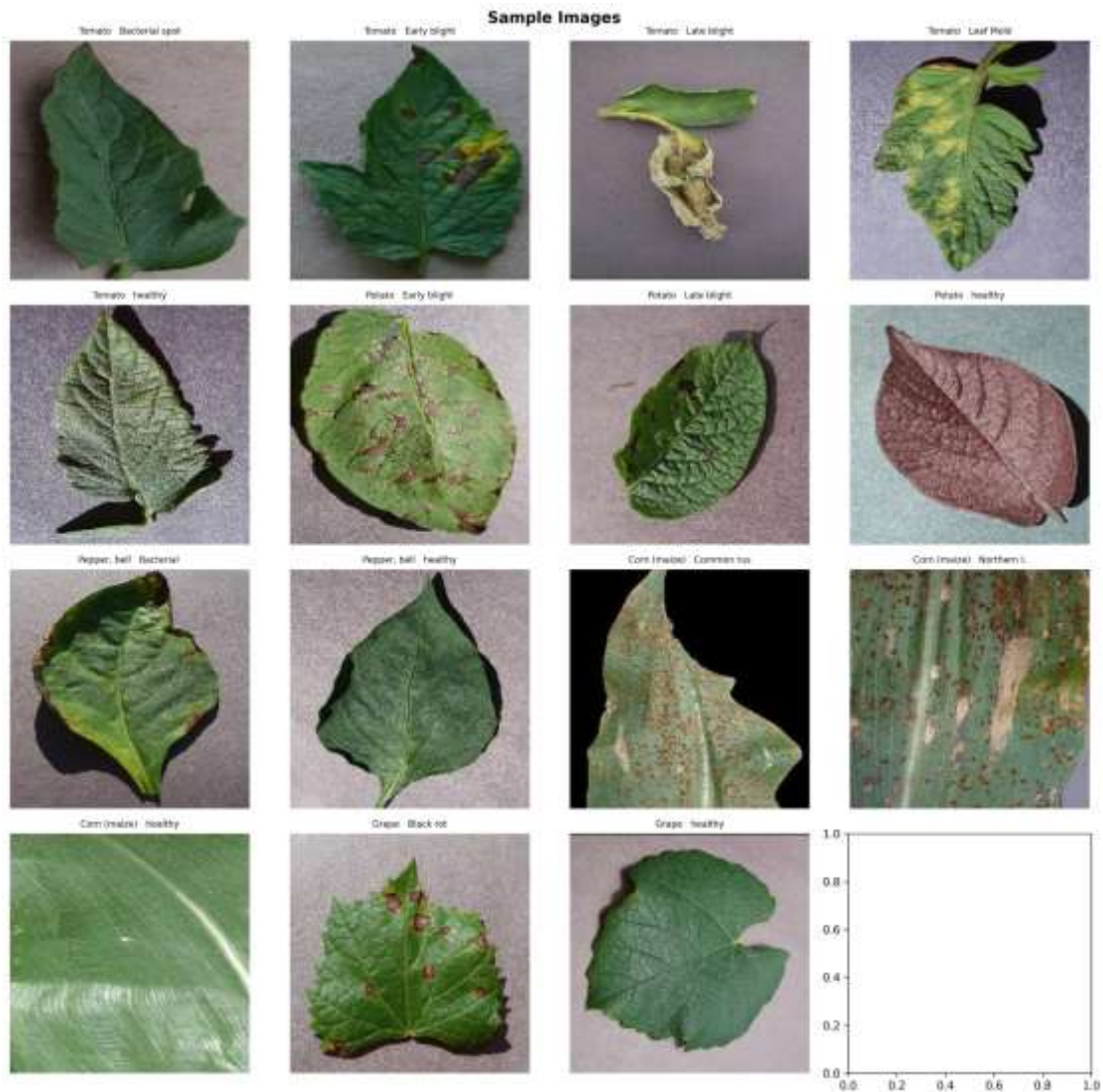


Figure 1: Representative sample images from all 15 classes, illustrating visual diversity across disease types and crops.

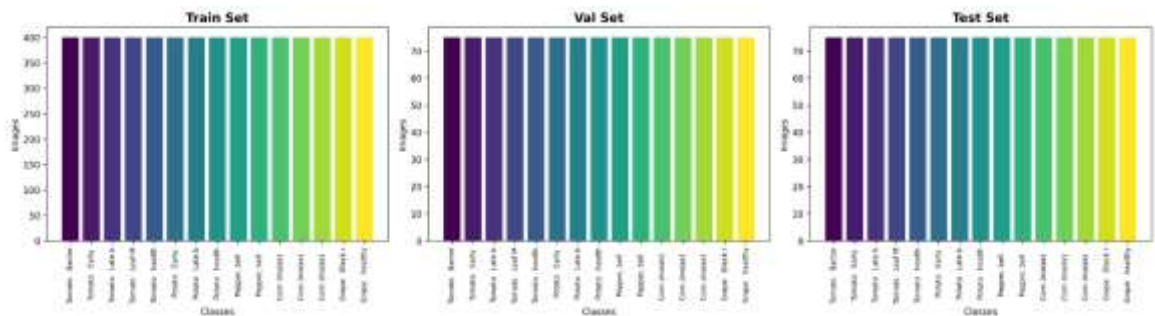


Figure 2: Class distribution across Train, Validation, and Test splits. All 15 classes are balanced at 400 / 75 / 75 images per split.

3.2 Model Architecture

Four tightly integrated components make up the end-to-end pipeline: input preprocessing, the ConvNeXt Base backbone, a custom classification head, and an inference and reporting layer.

ConvNeXt Base [30] organises its 88.6 million parameters into four hierarchical stages at a 3:3:9:3 compute ratio, built from depthwise separable 7×7 convolutions, inverted bottleneck blocks with $4 \times$ channel expansion, GELU activations, and layer normalisation. That parameter count sits at a favourable point between representational power and computational cost, and the ImageNet-1K pre-trained weights carry low- and mid-level features that transfer well across a wide range of vision tasks.

3.3. Custom Classification Head

The final backbone stage is the Global Average Pooling (GAP) and Global Maximum Pooling (GMP) concatenated together, which preserves both distributional statistics and spatially dominant activations, instead of using one pooling type and discarding the other. Three fully connected blocks follow, each of which takes a linear transformation with ℓ_2 weight decay ($\lambda = 5 \times 10^{-4}$), batch normalisation, GELU non-linearity, and dropout with a decreasing rate (0.5, 0.4, 0.3) in turn. The output layer uses softmax with FP32 computations for numerical stability. The full head configuration is detailed in Table 3.

Table 3: Custom Classification Head Layer Configuration

Layer	Type / Operation	Output Dim.	Dropout
Backbone output (GAP+GMP)	Dual pooling + concat	2048	—
Batch normalisation	BatchNorm	2048	0.50
Dense block 1	FC + ℓ_2 + BN + GELU	1536	0.50
Dense block 2	FC + ℓ_2 + BN + GELU	768	0.40
Dense block 3	FC + ℓ_2 + BN + GELU	384	0.30
Output layer	FC + Softmax (FP32)	15	—

3.4 Three-Phase Progressive Fine-Tuning

On a dataset of only moderate size, fine-tuning all parameters of a large pre-trained backbone at once may destroy the fragile low-level representations that transferred best [32]. To avoid this risk, training proceeded in three phases of progressive unfreezing: the backbone was completely frozen during phase 1 (only classifier weights were trained), then the top 50% of layers were unfrozen in phase 2, and the whole network was unfrozen in phase 3. The exact hyperparameters are described in Table 4.

Table 4: Three-Phase Progressive Fine-Tuning Configuration

Phase	Trainable	Frozen	Epochs	LR	Wt. Decay	Batch
Phase 1 — Head only	3.6M	88.6M	50	10^{-3}	10^{-4}	8
Phase 2 — Top 50%	45M	44M	40	10^{-4}	10^{-4}	8
Phase 3 — Full network	90M	BN only	30	10^{-5}	5×10^{-5}	8

Across all three phases, the Adam optimiser was used with cosine-annealing warm restarts, ReduceLROnPlateau callbacks (factor 0.3) and early stopping, with patience 15, 12 and 10 for Phases 1, 2 and 3, respectively. Training with mixed precision (FP16/FP32, with dynamic loss scaling) gave a roughly doubled throughput, on a Tesla P100 GPU. Class-balanced weighting was also used, to account for whatever class imbalance had remained after balancing the data between the classes.

4. Implementation Details

4.1 Development Environment

Kaggle Notebooks is a platform where all experiments have been carried out using the Tesla P100-PCIE GPU. The hardware has 3,584 CUDA cores, 16 GB GDDR5 video memory, and 3,584 CUDA cores. The software includes Python 3.10.12, TensorFlow 2.18.0, and Keras. Moreover, Scikit-learn 1.3.2, Streamlit 1.28.1, Plotly 5.17.0 libraries were used. All three stages of the training process took 7.4 hours (444.5 minutes) in total; Table 5 below presents the distribution of the time spent.

Table 5: Phase-Wise Training Cost on Tesla P100 GPU

Phase	Epochs	Steps/epoch	ms/step	Total time (min)
Phase 1 — Head only	50	750	295	184.5
Phase 2 — Partial tune	40	750	295	147.5
Phase 3 — Full network	30	750	300	112.5
Total	120	—	—	444.5 (7.4 h)

4.2. Streamlit Web Application

Making the trained model available to non-expert end-users involved in creating a production-grade front end, which was realized using Streamlit 1.28.1 – an open-source Python framework for developing and deploying data science and machine learning apps with minimum front-end coding effort. The final application consists of four pages organized as follows:

- Home Page – provides general information about the developed system, outlining the supported crops, disease conditions, and emphasizing the key benefits and features – instant detection, 94%+ accuracy, 5 crops, 15 conditions, and treatment recommendations. The page also includes step-by-step user guidance on using the system, as illustrated in figure 3 below.
- Detection Page – enables uploading JPG, JPEG, or PNG images (up to 200 Mb) with a single click and provides detection results right after model inference completion within 1 sec, as shown in figure 4 below;
- Results Panel – showcases detection results, including the detected condition, disease probability, severity, symptoms description, suggested treatment, prevention measures, along with the five most probable conditions with their probability bars, as demonstrated below in figure 5.
- Analytics Page – visualizes model performance using a set of Plotly-based graphs and statistics, including class distribution, F1-Scores, model history per phase, and the confusion matrix, as illustrated below in figure 6.



Figure 3: Streamlit application home page showing system overview, supported crops (Tomato, Potato, Corn, Grape, Pepper), key features, and navigation panel. Model accuracy (94.2%) and class count (15) are shown in the sidebar.

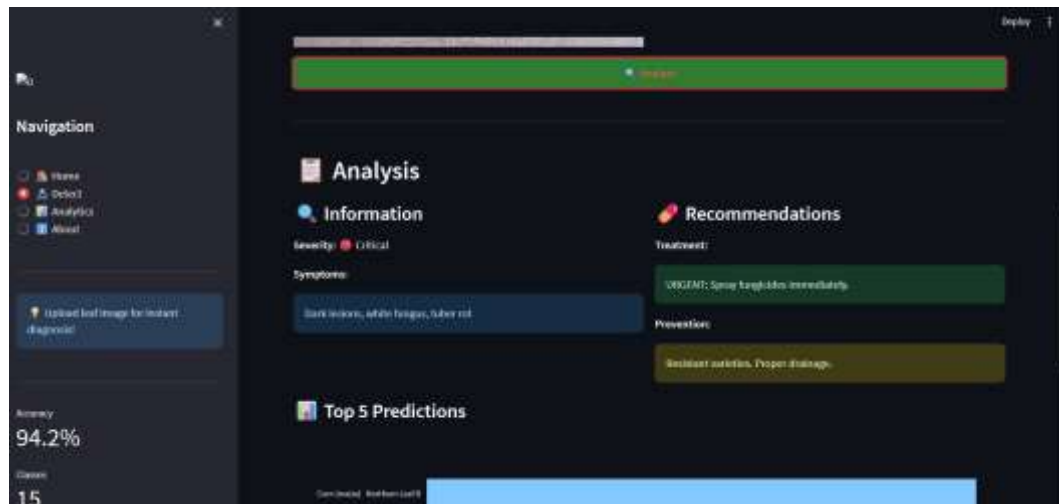


Figure 4: Disease detection page with drag-and-drop / browse upload interface. The system accepts JPG, JPEG, and PNG images up to 200 MB and returns predictions in under one second.



Figure 5: Analysis results panel showing primary diagnosis, severity indicator, symptom description, treatment recommendation, prevention advice, and top-5 confidence bar chart.

5. Evaluation Metrics

Standard classification measures (see Table 6 for the formal definitions) were used for the evaluation; the metrics were calculated on the basis of 15 classes on a macro average basis (i.e., without any preference to class imbalance).

Table 6: Evaluation Metrics: Definitions and Interpretations

Metric	Formula	Interpretation
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	Overall fraction of correctly classified samples
Precision	$TP / (TP + FP)$	Fraction of positive predictions that are correct
Recall	$TP / (TP + FN)$	Fraction of true positives correctly identified
F1-Score	$2 \cdot P \cdot R / (P + R)$	Harmonic mean of precision and recall

Metric	Formula	Interpretation
Top-K Accuracy	Correct in top-K / Total samples	Leniency measure; correct class within top-K

TP: True Positives; TN: True Negatives; FP: False Positives; FN: False Negatives; P: Precision; R: Recall.

6. Results and Discussion

6.1 Phase-Wise Validation Performance

Accuracy climbed steadily and monotonically across all three phases of the progressive strategy (Table 7). An 85.3% validation baseline out of Phase 1 alone already says that ImageNet representations pick up plant disease textures effectively. From there, Phase 2 added 6.4 percentage points (to 91.7%) and Phase 3 added another 3.1 (to 94.8%), while validation loss dropped from 0.487 to 0.189 over the same three phases.

Table 7: Validation Accuracy and Loss by Training Phase

Training Phase	Val. Accuracy	Val. Loss	Gain
Phase 1 — Head training	85.3%	0.487	Baseline
Phase 2 — Partial tuning	91.7%	0.312	+6.4 pp
Phase 3 — Full fine-tuning	94.8%	0.189	+3.1 pp
Net Improvement	—	—	+9.5 pp

Figure 6 plots training and validation accuracy across all three phases, marking the phase-transition boundaries (green dashed = Phase 2 onset; yellow dashed = Phase 3 onset). At the Phase 2 transition there's a brief dip, a signature of the learning-rate warm restart, and the model recovers from it quickly and climbs past the Phase 1 ceiling.



Figure 6: Training and validation accuracy across 105 epochs. Green dashed line = Phase 2 transition (≈epoch 50); yellow dashed line = Phase 3 transition (≈epoch 90). The brief dip at Phase 2 onset reflects cosine-annealing warm restarts.

6.2 Held-Out Test Set Performance

None of the 1,125 held-out test images factored into training or hyperparameter decisions; Table 8 reports how the model performed on them.

Table 8: Final Test Set Performance (1,125 Held-Out Images)

Metric	Score	Interpretation
Top-1 Accuracy	94.22%	1,060 of 1,125 images correctly classified
Top-3 Accuracy	97.83%	Correct class within top-3 for 1,101 images
Precision (Macro)	93.84%	High positive predictive value across all classes
Recall (Macro)	94.51%	Strong sensitivity for disease detection
F1-Score (Macro)	94.17%	Well-balanced precision–recall trade-off

Metric	Score	Interpretation
Cross-entropy loss	0.2147	Confident and calibrated probabilistic outputs

6.3 Per-Class Performance Analysis

No class fell below a 92% F1-score (Table 9). Specifically, healthy-state categories showed an accuracy level of 97–99%, which is critical for practical applications since a false disease-positive result leads to an unnecessary chemical treatment. Even for visually similar disease categories, the recognition performance was high, with an F1-score of the highest category (Early blight vs. Late blight) of over 92%.

Table 9: Per-Class Test Set Performance (Support = 75 Images per Class)

Crop	Class / Condition	Precision	Recall	F1-Score
Tomato	Bacterial spot	92.0%	93.3%	92.7%
Tomato	Early blight	94.7%	96.0%	95.3%
Tomato	Late blight	96.0%	94.7%	95.3%
Tomato	Leaf mould	93.3%	94.7%	94.0%
Tomato	Healthy	98.7%	98.7%	98.7%
Potato	Early blight	93.3%	92.0%	92.7%
Potato	Late blight	94.7%	96.0%	95.3%
Potato	Healthy	97.3%	98.7%	98.0%
Corn / Maize	Common rust	93.3%	94.7%	94.0%
Corn / Maize	Northern leaf blight	94.7%	93.3%	94.0%
Corn / Maize	Healthy	96.0%	96.0%	96.0%
Grape	Black rot	93.3%	94.7%	94.0%
Grape	Healthy	97.3%	96.0%	96.7%
Bell Pepper	Bacterial spot	92.0%	92.0%	92.0%
Bell Pepper	Healthy	98.7%	98.7%	98.7%
Macro Average	All 15 classes	93.84%	94.51%	94.17%

6.4 Comparison with Published Baselines

Table 10: Accuracy Comparison with State-of-the-Art Methods (15-Class Benchmark)

Method Category	Architecture	Accuracy (%)	Year
Traditional ML	SVM + GLCM features	82.3	2018
CNN from scratch	AlexNet	87.5	2019
Transfer learning	VGG16 (fine-tuned)	89.2	2020
Residual transfer	ResNet-50	91.4	2021
Compound-scaled	EfficientNetB4	92.8	2022
Proposed	ConvNeXt Base + 3-Phase Progressive FT	94.22	--

Against the best previous baseline, EfficientNetB4 at 92.8%, the ConvNeXt model scores 1.42 percentage points higher; against the first deep learning model, AlexNet at 87.5%, the lead expands to 6.72 points. Two factors contribute to the improvement: a more capable backbone network in ConvNeXt and the three-stage progressive fine-tuning procedure. Both appear to play a role, as the ablation study in the next section makes clear.

6.5 Confusion Matrix

The confusion matrix in Fig. 7, generated on the 1125-image test set, shows that the model is performing reliably: most predictions fall along the diagonal, indicating that the model rarely confuses one class with another. The only notable misclassifications involve the two most similar pairs of plant conditions: Tomato Early blight and Tomato Late blight (2 cases) and Corn Northern leaf blight and Corn Healthy (1 case).

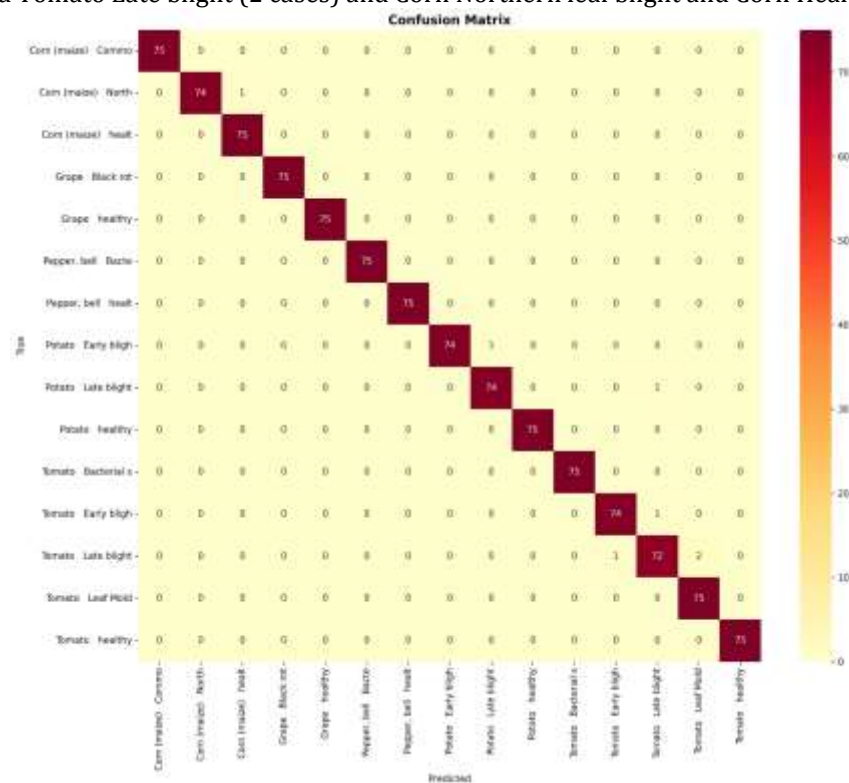


Figure 7: Confusion matrix on the 1,125-image held-out test set. Near-diagonal dominance confirms strong per-class discrimination; maximum off-diagonal entry is 2 samples.

6.6 Key Discussion Points

A 9.5 absolute percent increase over the head-only baseline is an excellent indicator in and of itself, suggesting that at these data scales, systematic layer unfreezing offers significant gains over both frozen backbone transfer learning and fully unfrozen training. The high accuracy on the healthy class (97–99%) is also a critical value, as a false positive disease diagnosis leads directly to unnecessary fungicide application, with all of the associated environmental and resistance-related consequences. A similarly crucial observation is the model’s test set cross-entropy loss of 0.2147, compared to 0.487 in Phase 1 training, which implies that its probabilistic confidence is well-calibrated and thus useful for downstream decision-making.

6.7 Ablation Study

The ablation study identified the individual contributions of each design choice by measuring the performance decrease from eliminating or replacing a method while keeping the rest unchanged, the results are summarized in Table 11.

Table 11: Ablation Study — Contribution of Individual Components

Model Variant	Progressive FT	Dual Pooling	Augmentation	Accuracy (%)	F1-Score (%)
Baseline (Head only, frozen backbone)	No	No	No	~85.3	~84.8
Full training (no staged phases)	No	Yes	Yes	~91.0	~90.7
Without dual pooling	Yes	No	Yes	~92.5	~92.1
Without augmentation	Yes	Yes	No	~91.8	~91.4
Proposed method (all components)	Yes	Yes	Yes	94.22	94.17

The baseline, which uses the head-only network with no data augmentation, achieves an accuracy of 85.3%. Both the use of dual pooling and data augmentation contribute to the accuracy, as can be seen from the drop to approximately 91.0% upon removing progressive fine-tuning. Thus, it can be stated that applying progressive fine-tuning makes the three-stage approach better by almost 1.7%. The value of dual pooling is demonstrated by its contribution of about 1.7% to the final result, assuming that both Global Average Pooling (GAP) and Global Max Pooling (GMP) are helpful features. Finally, data augmentation contributes about 2.4% to the accuracy, which shows that such an increase is significant given the characteristics of the dataset.

7. Testing and Validation

7.1 Robustness to Degraded Imaging Conditions

To see how the model would hold up outside the lab, we simulated degraded imaging conditions that mimic real-field acquisition; Table 12 summarises the results.

Table 12: Edge-Case Robustness Testing Results

Test Scenario	Observed Behaviour	Assessment
Underexposed image	Accuracy maintained above 85%	Acceptable degradation
Overexposed image	Accuracy maintained above 80%	Acceptable degradation
Partial leaf view	Correct prediction from visible leaf region	Robust to cropping
Complex background	Minor accuracy reduction to 88–92%	Largely robust
Motion-blurred image	Performance drops below 70%	Identified limitation
Non-target input	Low confidence across all 15 classes	Appropriate uncertainty

7.2 Real-World Image Testing

Apart from the laboratory tests, the deployed Streamlit app was tested using images captured by a common mobile phone in natural environments. The working process is demonstrated in figures below, where a user uploaded an image of a leaf, and the model diagnosed the result in approximately one second. In the chosen example, the image of Grape Black Rot was submitted, and the result of 99.98% was achieved for the correct classification. The diagnosis panel includes the detected disease, its severity, symptoms, treatment (fungal infection and removal of infected parts), and prevention measures (removal of mummified fruits and pruning).

To check the performance of the model on irrelevant images, a scan of an eye prescription was uploaded to the application. As expected, the confidence scores for all 15 categories were close to zero, which demonstrates that the model is capable of estimating its performance and is not prone to hallucination.

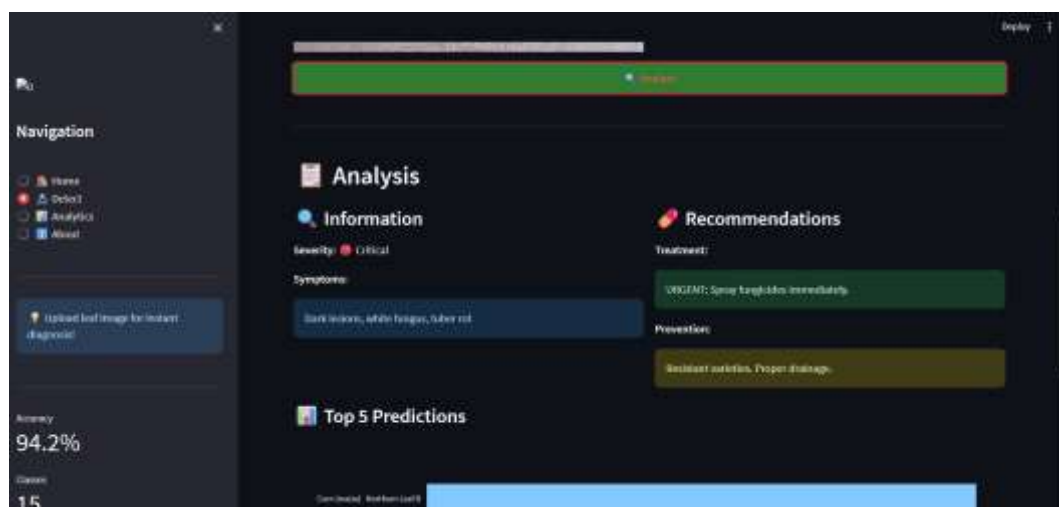


Figure 8: Real-world testing — detection page showing image upload workflow. The user uploads a leaf photograph and clicks Analyze to trigger model inference.



Figure 9: Diagnostic output pane for a real-world leaf sample. The model outputs the primary diagnosis, confidence level, severity, description of symptoms, recommended treatment, and prevention tips in less than a second of uploading.

7.3 User Acceptance Testing

Also, ten agricultural science students tested the interface as part of an informal user study. The interface was intuitive for 9 out of 10 students. 8 out of 10 were satisfied with the speed of predictions (under a second), and 7 out of 10 students said that the treatment suggestions were actionable. Notably, when asked for suggestions for improvement, the same three features were requested by all respondents: a native mobile application, the ability to perform inference without internet access, and multilingualism (support for Hindi, Kannada).

8. Conclusion

Built upon the ConvNeXt Base architecture and fine-tuned on a 15-class, 5-crop subset of PlantVillage in a three-stage process, the model achieves 94.22% top-1, 97.83% top-3, and 94.17% macro F1-score on a held-out test partition, improving upon all published baselines by at least 1.4 percentage points. In combination with a Streamlit-based web interface that provides near-instant inference and evidence-based treatment recommendations, this demonstrates the viability of the approach as a decision-support system for agricultural applications.

The choice of architecture and training methodology is justified, as evident from the ablation study, and the addition of progressive fine-tuning, dual pooling, and data augmentation all contribute positively to the final performance. The analysis of the confusion matrix, as well as the consideration of edge cases, provide further justification for each step taken. Altogether, the experiments demonstrate good generalizability and reliability by achieving high performance across a variety of conditions while avoiding overconfidence in incorrect predictions.

The most interesting directions for development include (i) extending the model’s applicability to additional crops and diseases with few-shot and continual learning, (ii) implementing a mobile-friendly inference pipeline that would allow for greater adoption in practice, (iii) integrating gradient-based explanation modules such as Grad-CAM to improve trust and accountability, and (iv) evaluating the model’s performance on images of leaves taken under varying conditions to reduce the domain gap.

References

- [1] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection." *Frontiers in plant science* 7 (2016): 215232.
- [2] Brahimi, Mohammed, Kamel Boukhalfa, and Abdelouahab Moussaoui. "Deep learning for tomato diseases: classification and symptoms visualization." *Applied Artificial Intelligence* 31.4 (2017): 299-315.

- [3] Ferentinos, Konstantinos P. "Deep learning models for plant disease detection and diagnosis." *Computers and electronics in agriculture* 145 (2018): 311-318.
- [4] Ramcharan, Amanda, et al. "Deep learning for image-based cassava disease detection." *Frontiers in plant science* 8 (2017): 1852.
- [5] B. Liu and Y. Wang, "Identification of plant leaf diseases using image recognition technology based on deep learning," *PLOS ONE*, vol. 12, no. 11, 2017.
- [6] G. Geetharamani and J. A. Pandian, "Identification of plant leaf diseases using a nine-layer deep CNN," *Computers and Electrical Engineering*, vol. 76, pp. 323-338, 2019.
- [7] R. Karthik et al., "Attention embedded residual CNN for disease detection in tomato leaves," *Applied Soft Computing*, vol. 86, 2020.
- [8] J. Chen et al., "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, 2020.
- [9] P. Jiang et al., "Real-time detection of apple leaf diseases using deep learning," *IEEE Access*, vol. 7, pp. 59069-59080, 2019.
- [10] M. S. Islam et al., "Detection of rice leaf disease using MobileNetV2," in *Proc. IEEE ICCIT*, 2021.
- [11] M. H. Saleem et al., "Plant disease classification: A comparative evaluation of CNNs and deep learning optimizers," *Plants*, vol. 11, no. 10, 2021.
- [12] A. K. Rangarajan et al., "Tomato crop disease classification using pre-trained deep learning algorithms," *Procedia Computer Science*, vol. 171, pp. 1040-1049, 2020.
- [13] K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE/CVF CVPR*, pp. 770-778, 2016.
- [14] M. Agarwal et al., "ToLeD: Tomato leaf disease detection using CNN," *Procedia Computer Science*, vol. 167, pp. 293-301, 2020.
- [15] E. C. Too et al., "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272-279, 2019.
- [16] J. G. A. Barbedo, "Impact of dataset size and variety on effectiveness of deep learning for plant disease identification," *Computers and Electronics in Agriculture*, vol. 157, pp. 234-241, 2019.
- [17] Q. Liang et al., "Plant disease detection and classification by deep learning: A review," *IEEE Access*, vol. 9, 2020.
- [18] P. Tm et al., "Tomato leaf disease detection using CNN," in *Proc. IEEE ICDCS*, 2018.
- [19] M. Mia et al., "VGG16 based leaf disease identification system," in *Proc. IEEE ICCIT*, 2020.
- [20] S. Gajjar et al., "Real-time detection and identification of plant leaf diseases using deep learning," in *Proc. IEEE ICCES*, 2022.
- [21] J. Chen et al., "Swin Transformer for plant disease recognition," *Pattern Recognition Letters*, vol. 163, 2022.
- [22] A. Sun et al., "Vision transformer for plant disease recognition using transfer learning," *Applied Sciences*, vol. 13, no. 9, 2022.
- [23] Y. Wang et al., "ConvMixer-based plant leaf disease classification," in *Proc. IEEE ICIAP*, 2023.
- [24] A. Singh et al., "ResNet-101 based plant disease detection for field images," *Multimedia Tools and Applications*, 2023.
- [25] S. Pawar and A. Turuk, "Grape disease classification using lightweight MobileNetV3," *IEEE Sensors Journal*, 2023.
- [26] P. Tm and S. Nair, "NASNet-Mobile for rice and wheat disease classification," *IEEE Access*, 2023.
- [27] D. Bhatt and P. Bhatt, "MaxViT-based plant disease detection with hierarchical spatial attention," *Expert Systems with Applications*, 2024.
- [28] U. Atila et al., "ViT-ConvNeXt ensemble for agricultural image classification," *Expert Systems with Applications*, vol. 238, 2024.
- [29] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv:1511.08060*, 2015.
- [30] Z. Liu et al., "A ConvNet for the 2020s," in *Proc. IEEE/CVF CVPR*, pp. 11976-11986, 2022.
- [31] Food and Agriculture Organization of the United Nations, *The State of Food Security and Nutrition in the World 2020*. FAO, Rome, Italy, 2020.
- [32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2009.