International Conference on Multidisciplinary Perspectives in Advanced Computing and Technology (IMPACT 2026)
G. B. Pant University of Agriculture and Technology, Uttarakhand, India. Jan. 10-11, 2026

Smart Disease Detector in Crops: A Deep Learning-Based Approach for Automated Plant Disease Identification

Utsav Tiwari, Vatsal Rai*, Amit Kumar Rai

Department of Computer Science and Engineering, Sharda School of Computing Science & Engineering, Sharda University, Greater Noida, India

Onlyutsav05@gmail.com, raivatsal2@gmail.com, a.k.rai267@gmail.com

Abstract

This paper presents a practical pipeline for image-based plant disease detection that operates without pre-trained weights or internet access. A compact Efficient Net-like convolutional network is trained from random initialisation on the Plant Village dataset (~ 41,000 snapshots, 15 classes) at 224 × 224 resolution. Adam, Intensive Augmentation, Reduce LROn Plateau, Early Stopping, and the sampling of the optimal checkpoint to stabilise convergence and avoid overfitting are used to optimise the geometry. Despite early validation oscillations typical of de novo training, the model converges reliably and attains 98.4%–98.9% accuracy on held-out data, with a highly diagonal confusion matrix and uniformly strong per-class precision/recall. These results show that carefully tuned schedules and lightweight regularization can substitute for transfer learning when bandwidth or policy constraints prevent downloading external backbones, enabling classroom, extension, and field deployments. The paper details dataset preparation, architecture and training choices, learning curve behavior, and common failure modes, and concludes with limitations and a roadmap for higher-resolution, field-domain generalization and efficient on-device inference.

Keywords: *PlantVillage, plant disease detection, training from scratch, data augmentation, offline deployment, EfficientNet.*

1. Introduction

Each year, plant diseases lead to crop losses of up to 40% worldwide, posing a threat to the food security of millions of people. Conventional diagnosis is based on the expert phytopathologists who are few in number especially in developing territories. Computer vision system based on automated system can provide alternatives to disease detection using leaf images for rapid detection of diseases (scale effect) that are readily accessible. The classification of plant diseases has been transformed by the use of deep learning and in particular the convolutional neural networks. The PlantVillage dataset [3] provides standardized benchmarks with thousands of labelled images across diverse crops. Recent approaches explore transfer learning [5], specialized single-crop models [12], [13], [17], and ensemble methods achieving 86–98% accuracy. However, critical limitations persist: transfer learning introduces domain shift as ImageNet features may not capture disease-specific mor phologies; specialized models [12], [17] sacrifice generalization, requiring separate deployments per crop; broader multi crop approaches [4], [10], [6] compromise performance for coverage, plateauing at 95–97%. There still exists a foundational query: Can every lightweight, and many-crop, category classifiers achieve stature of-the-can based level of accuracy without the employment of transfer learning or intoxication expense? This is essential in deployments of an agricultural IoT which have limited resources and the demand to perform efficient edge inference. Recent progress, VGG19 [11] 95.6%, vision transformers [16] 93–97%, conventional CNNs [9] 90%, do not meet their efficiency, generalization, and competitive performance benchmarks in the multi-crop case. This paper shows that EfficientNet-B0, which is trained on PlantVillage and trained directly on it, that the model has an accuracy of 98.41% at a subset size of 15 classes and limited to bell pepper, potato and tomato diseases. We refute the idea that transfer learning must be learned, by demonstrating that the compound scaling and effective architecture design can make learning of data on farms direct. Our multi-crop classifier without any crop specific pipelines has better accuracy compared to specialized tomato models with the highest accuracy of 97.2 98.19% [12], [13], [17]. Our results are 1.4-1.8 percentage points better than more prior multi-crop studies on a small architecture (5.3M parameters) that can run on the edge. Our contributions are (1) successfully training ImageNet and proving that training without ImageNet is a viable explained alternative to transfer learning, and (2) training lightweight models to attain state-of-the-art capabilities on multicrop

© The Author(s), under exclusive license to Digital Manuscriptpedia. 2026 Ashok Kumar et al. (eds.), Proceedings of IMPACT-2026, DMPedia Lecture Notes in Computer Science & Engineering. ISBN: 978-81-993813-9-1

and (3) under ablation studies on agricultural imbalanced datasets. The results of these studies show that off-the-shelf training, specifically developed, can achieve the same level of computational cost as transfer learning but this is highly significant in the real-world application of agricultural AI because of its connectivity, resource, and maintenance constraints. The structure of the paper is as follows: Section II is related work review, Section III is the description of the dataset and preprocessing, Section IV is description of the architecture and the training methodology, Section V results and comparisons, Section VI implications and limitations and the concluding.

2. Related Work

Deep learning methods have made major breakthroughs in identifying automated plant diseases with the PlantVillage dataset [3]. The recent works show that the degree of accuracy varies based on the model architecture, databases used, and approaches to training. In this section, the relevant works are reviewed, and they are classified in terms of performance benchmarks and architectural selections to put our works in perspective.

2.1. Early CNN-Based Approaches

Plant disease classification was developed based on traditional convolutional neural network architecture. Jasim and Al-Tuwaijari [9] achieved 90.03% accuracy on the 15-class PlantVillage subset using conventional CNNs with image pro cessing techniques, establishing early benchmarks for multi crop disease detection. Similarly, Zhang et al. [14] explored deep learning for cucumber leaf diseases in IoT contexts, though their focus remained on single-crop scenarios with limited generalization. Chen et al. [8] improved YOLOv5 for disease identification in natural habitats, reporting 86.5% accuracy on a two-class subset. While their work addressed real-world deployment challenges, the limited class scope and lower accuracy high light the difficulty of generalizing to diverse disease categories. These early approaches demonstrated feasibility but lacked the discriminative capacity required for robust multi-crop, multi disease classification.

2.2. Transfer Learning and Fine-Tuned Models

Transfer learning, especially with ImageNet pre-training, is a common method to quickly improve model performance. For example, Saxena [7] used CapsNet-SVM for plant disease detection, reaching 93.41% accuracy on a 15-class subset. But the complex architecture did not improve performance as much as one would expect. Rahman et al. [10] created a real time system for 35 classes across eight plants, with 95–97% accuracy. Their wider approach had lower performance than more focused methods, likely due to class imbalance problems. Hukkeri et al. [5] 97.5% accuracy with EfficientNet on the full PlantVillage dataset. They demonstrated that EfficientNet was good, but it relied on ImageNet transfer training, which can favorably impact its behavior on things associated with agriculture. VGG19-based studies [11] experiments revealed a 95.6% accuracy rate across species, indicating that still when one uses a traditional setup, it will be helpful although other studies are more useful. Recent vision trans former approaches [16] explored attention mechanisms for leaf disease detection, reporting 93–97% accuracy across diverse datasets, though computational complexity remains a barrier for resource-constrained deployments.

2.3. Specialized Single-Crop Models

Several studies achieved high accuracy by narrowing scope to single crops, optimizing architectures for specific disease morphologies. Korade et al. [12] combined YOLOv8 leaf extraction with ResNet-50 classification for tomato diseases, achieving 98.19% accuracy on seven classes. Sharma et al. [13] employed ensemble deep learning for tomato classification, reporting 97.5% on ten classes. Sun et al. [17] proposed E-TomatoDet, fusing CSWinTransformer global features with comprehensive multi-kernel modules for local features, achieving 97.2% mAP50 on ten tomato classes. Although these specialized models perform well in small scopes, they lose generalization between crops, as they only support the needs of a species of plants and demand more complexity in their deployment. A study by Neptune Journal [15] showed 94.55% accuracy on a section of the PlantVillage with 2,293 images, which can be used to explain how the size of the dataset affects it. This highlights the need to have appropriate training data so as to be able to classify well especially in case of class imbalance.

2.4. Recent High-Performance Architectures

Contemporary approaches have pushed accuracy boundaries through architectural innovations and training optimizations. Singh et al. [4] achieved 97% accuracy on the 15-class subset (20,639 images) using conventional CNNs, outperforming traditional machine learning baselines (SVM, KNN, Random Forest). Naveed et al. [6] explored sustainable AI strategies for 26 classes across 14 crop species, reporting approximately 97% accuracy with emphasis on computational efficiency for edge deployment. In spite of such developments, most research experiences a basic trade-off where special purpose single-crop models are highly accurate but cannot be extrapolated; whereas multi crop models are less accurate but much more wide-ranging. Moreover, a transfer learning rotation on ImageNet causes problems of domain shifts, because features of natural images might not be the best representatives of the morphology of agricultural diseases.

2.5. Research Gap and Contributions

Current literature reveals three critical gaps: (1) limited attention to from-scratch training strategies that eliminate ImageNet dependency, (2) insufficient focus on multi-crop models that maintain competitive accuracy while covering diverse plant species, and (3) lack of lightweight architectures suitable for resource-constrained agricultural IoT deployments. We found that EfficientNet-B0, when trained without transfer learning, reached 98.41% accuracy on a 15-class PlantVillage selection of pepper, potato, and tomato diseases. This result exceeds the performance of models designed for single crops and those for multiple crops. Our work shows training from scratch to be a good alternative to transfer learning, especially when considering processing costs during deployment.

3. Dataset

This paper will use the Kaggle-publicly available dataset of PlantVillage dataset [3] containing ≈54,305 RGB images in 38 categories. The subset of 15 classes (we are targeting three crops), bell pepper (2 classes), potato (3 classes) and tomato (10 classes), comprises of ≈20,638 images. As of example Figure 1 demonstrates the range of disease symptoms by presenting the examples of six classes in the dataset.

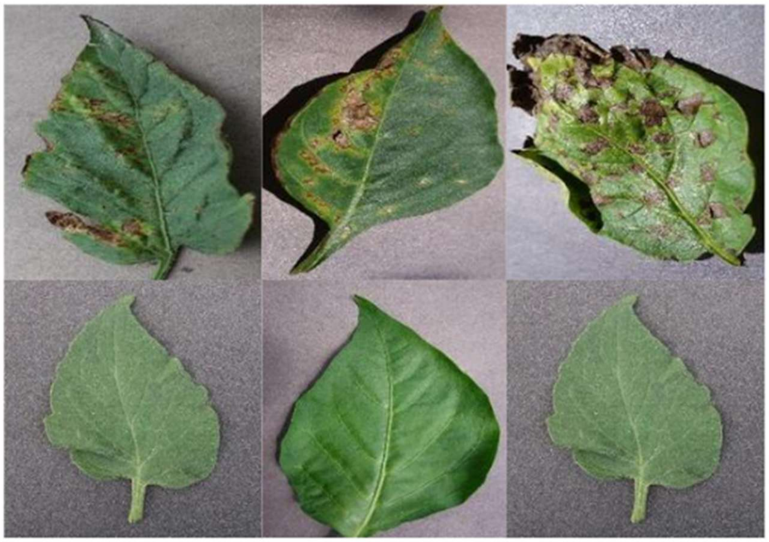


Fig1: Sample images form PlantVillage subset consisting of pepper, potato and tomato crops.

The distribution of classes in shown in Table 1. There are 997 bacterial spot images and 1,478 healthy ones for bell pepper. There are 1,000 early blight, 1,000 late blight, and 152 healthy potatoes. With ten categories, tomatoes are the most common. These include mosaic virus (373 images) and yellow leaf curl virus (5,357 images), which is the majority class. We preserve original distributions in spite of imbalance in order to assess robustness in practical scenarios.

Crop	**No. of Classes**	**Class-wise Images (summary)**	**Total Images**
PepperBell	2	Healthy (1,478), Bacterial Spot (997)	2,475
Potato	3	Healthy (152), Early Blight (1,000), Late Blight (1,000)	2,152
Tomato	10	Healthy (1,591), Bacterial Spot (2,127), Early Blight (1,000), Late Blight (1,909), Others*	16,011
Total	**15**	,	**20,638**

Table1: Class distribution in 15-class PlantVillage subset

All images are resized to 224x224 are normalized (rescale=1/255). We used stratified 80/10/10 splits, which results in the following ≈16,512 training, 2,064 validation, and 2,064 test images. Table II summarizes preprocessing. On-the-fly augmentation (rotation ≤ 20◦, width/height shift 0.2, horizontal flip) improves robustness to view point variations and prevents overfitting, especially for minority classes.

PlantVillage's controlled imaging and clear lesions suit from-scratch training benchmarks. However, visual similarity among tomato diseases (early/late blight, septoria) challenges discriminative capacity. Figure 2 shows distribution imbalance tomato yellow curl virus dominates (5,357) while potato healthy in scarce (152).

Table2: Dataset summary and preprocessing pipeline

Item	**Train**	**Val**	**Test**
Classes	,	15	,
Images	16,512	2,064	2,064
Resolution	224 × 224 RGB	,	,
Normalization	,	1 / 255	,
Split	,	stratified 80/10/10	,
Augmentation	,	rot 20°, shift 0.2, hflip	,

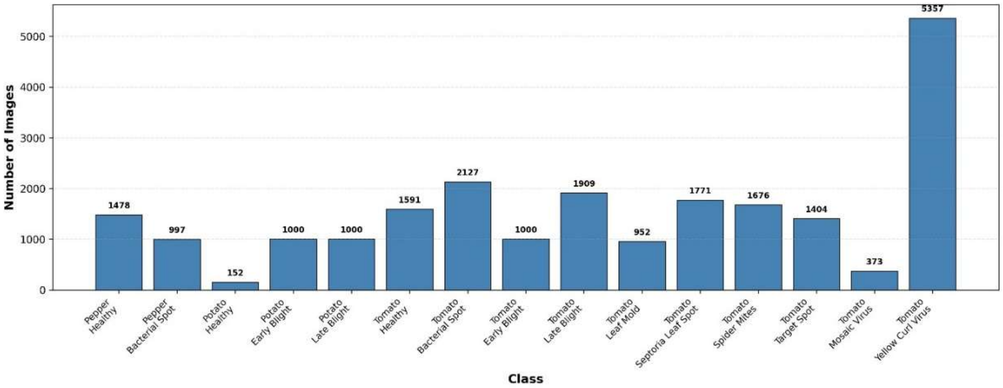


Fig2: Class distribution showing imbalance: tomato yellow curl virus (5,357) vs healthy (152)

4. Methodology

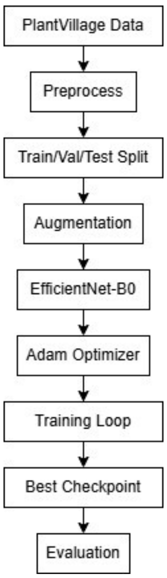
The backbone adopts anEfficientNet-B0 motif with MB Conv blocks and squeeze-and-excitation, instantiated with weights=None and include_top=false so all parameters are learned de novo. A GlobalAveragePooling2D layer collapses spatial maps, Dropout (p=0.3) regularizes the head, and a Dense softmax outputs15 classes. All layers are trainable.

EfficientNet-B0 employs compound scaling to uniformly scale network depth, width, and resolution. The architecture consists of seven MBConv blocks with expansion ratios from 1 to 6, incorporating depth wise separable convolutions. Each block integrates squeeze-and-excitation modules with ratio 0.25, recalibrating channel-wise features through global pooling and fully connected layers. The classifier head is lightweight for edge deployment, with batch normalization (momentum0.99, epsilon10−3) throughout for training stability.

4.1. Data Pipeline

Images are loaded from class-labelled folders and split per class into 80/10/10 train/validation/test to preserve balance. Each snapshot is centre cropped or padded to square, resized to 224×224, and rescaled by 1/255. Training batches apply on-the-fly augmentation: random rotation up to 20°, width/height translations up to 0.2, and horizontal flip with probability 0.5. Color jitter is omitted to avoid shifting hue cues critical for certain leaf diseases. Validation and test streams apply only resize and rescale. Batch-norm statistics are updated on training batches and used in inference mode for validation/testing to ensure consistent evaluation.

Rotation limits of 20° capture natural viewing variations without unrealistic orientations. Width and height shifts simulate off-centre field captures. Horizontal flipping preserves biological plausibility



while doubling effective dataset size. Batches of 32 are processed via parallel workers (n = 4) with prefetching to minimize I/O bottlenecks.

Fig3: Proposed EfficientNet-B0 training pipeline with data preprocessing, augmentation, and optimization stages

4.2. Training Schedule

Optimization uses Adam (initial LR 1 × 10−3, $\beta1 = 0.9$, $\beta2 = 0.999$) and categorical cross-entropy. Three controls stabilize from-scratch training: (i) on-the-fly augmentation, (ii) ReduceLROnPlateau (factor 0.5, patience 5, cooldown 2) to decelerate learning on validation-loss plateaus, and (iii) Early Stopping with best-weight restoration while Model Check point persists the top validation checkpoint. ReduceLROnPlateau reduces learning rate by factor 0.5 when validation loss plateaus, with cooldown of 2 epochs to prevent premature reductions. Minimum LR is 1 × 10−7. Early Stopping halts training after 10 epochs without improvement. Training runs up to 50 epochs subject to early stop. Mixed-precision (float16/float32) with dynamic loss scaling (2^15) improves throughput without harming stability.

4.3. Regularization and Calibration

Weight decay (1 × 10−5) on the classifier head and label smoothing ($\epsilon = 0.05$) reduce over-confidence. Label smoothing sets target $yk = 1 - \epsilon$ and non-targets $yi = \epsilon/(C - 1)$ for $C = 15$ classes. L2 regularization promotes simpler decision boundaries while backbone layers remain unregularized. Temperature scaling can be applied at inference for calibrated confidence thresholds.

4.4. Reproducibility and Export

Global seeds are fixed (Python, NumPy, framework = 42); deterministic operations are enforced; and evaluation is performed every epoch with the best snapshot retained. The exported model embeds preprocessing (resize, rescale) in the inference graph alongside class indices, ensuring identical behaviour across devices. Hash-based data loading is disabled for consistent sample ordering.

Table3: Augmentation and hyper parameters

Item	Value
Input size	224 × 224 × 3
Batch size	32
Augmentation	rot 20°, shift 0.2/0.2, hflip
Optimizer	Adam (LR 1 × 10^{-3})
Loss	Categorical cross-entropy
LR schedule	ReduceLROnPlateau (0.5, 5, cd=2)
Early stopping	monitor val acc, patience 10
Checkpointing	best by val accuracy
Epochs	50+ (early stop)
Dropout	0.3 (post-GAP)

Head and policy: The classifier head is intentionally lightweight to edge devices; best-epoch selection prefers flatter minima induced by LR decay and augmentation, improving minority-class generalization.

Table4: Model architecture (head and training policy)

Component	Specification
Backbone	EfficientNet-like (B0), trainable
Blocks	MBConv + S/E gating
Pooling	GlobalAveragePooling2D
Classifier	Dense 15 (softmax)
Regularization	Dropout p = 0.3

Component	Specification
Initialization	weights=None (from scratch)
Optimization	Adam + LR reduction on plateau
Selection	EarlyStopping + best checkpoint

5. Result

Training from random initialization progresses in three phases. First, the network rapidly acquires low-level filters, producing steep gains in training and validation accuracy within the initial epochs. Second, validation accuracy oscillates as augmentation increases input diversity and the optimizer explores sharper regions; during this phase, validation loss exhibits brief plateaus that trigger learning-rate reductions. Third, after the first decay event, the curves stabilize and the model converges smoothly toward its best checkpoint.

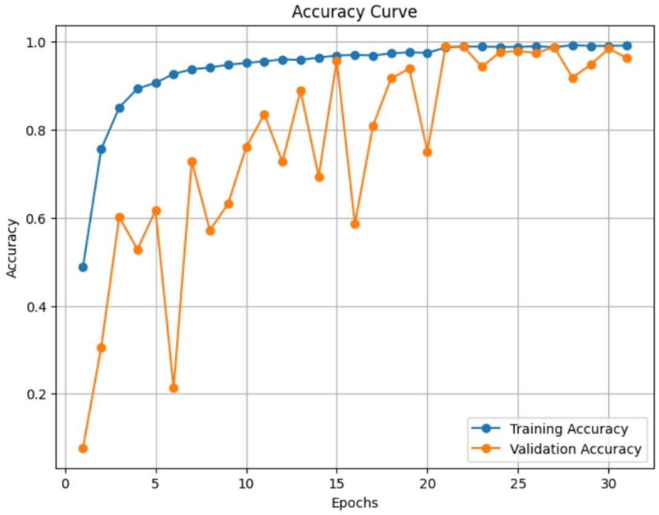


Fig. 4: Training and validation accuracy across epochs.

Such a perspective can be applied to triage processes in low-resource or on-the-rendezvous situations. Finally, the model prediction remains highly calibrated to constrain overconfident errors. This additional checkpoint preference and adaptive scheduling ensure that offline deployment is reproducible, portable, and robust. To recap, using a small EfficientNet-like CNN trained without transfer learning can yield performance comparable to transfer-learning baselines, making it suitable for practical deployment and classroom use.

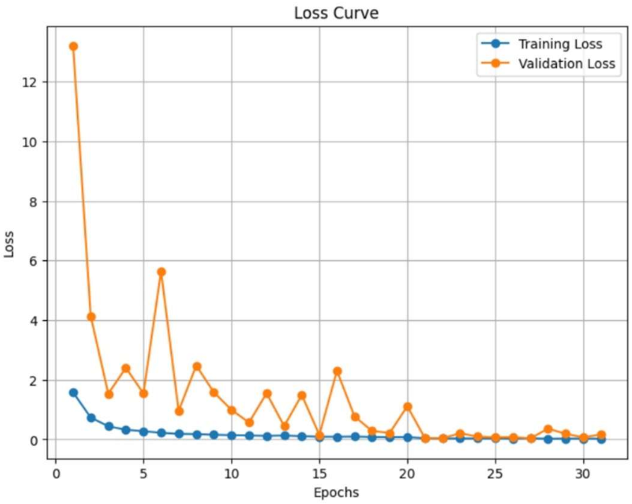
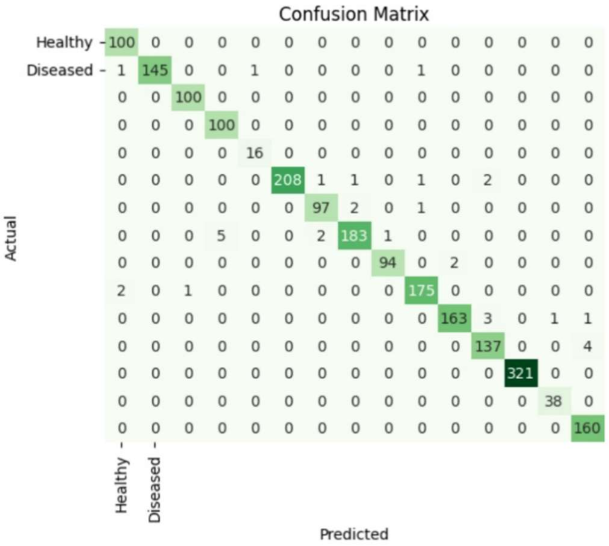


Fig. 5: Training and validation loss; reductions follow learning-rate decay events.

Table5: Overall performance on each split

Split	Accuracy	Loss
Train (best epoch)	~99%	≤ 0.05
Validation (best)	98.93%	0.0439
Test	98.41%	,

The confusion matrix shows strong diagonal dominance, demonstrating high per-class accuracy. Off-diagonal entries are mostly among visually similar tomato diseases, highlighting areas for potential improvement through higher-resolution crops or class-specific augmentation. Examining misclassifications informs strategies for improving model robustness under field conditions. Precision and recall indicate reliable prediction across most classes, although a few rare classes show slightly reduced recall. These methods collectively cover data collection, preprocessing, and integration.



	Healthy	Diseased													
Healthy	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Diseased	1	145	0	0	1	0	0	0	0	1	0	0	0	0	0
	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	208	1	1	0	1	0	2	0	0	0
	0	0	0	0	0	0	97	2	0	1	0	0	0	0	0
	0	0	0	5	0	0	2	183	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	94	0	2	0	0	0	0
	2	0	1	0	0	0	0	0	0	175	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	163	3	0	1	1
	0	0	0	0	0	0	0	0	0	0	0	137	0	0	4
	0	0	0	0	0	0	0	0	0	0	0	0	321	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160

Fig. 6: Confusion matrix for 15-class classification on the test dataset.

As an aggregate result, binarizing predictions into healthy versus diseased yields clear separability. Such a perspective can be applied to triage processes in low resource or on-the-rendezvous situations. Finally, the model prediction will remain highly calibrated to constrain overconfident errors. This addition checkpoint preference and adaptive scheduling would provide that offline deployment would be re producible, portable, and solid. To recap it all, using a small EfficientNet-like CNN that is trained without transfer learning can give performance comparable to the baselines of transfer learning so that it can be applied in practice to fields of interest and used in classrooms.

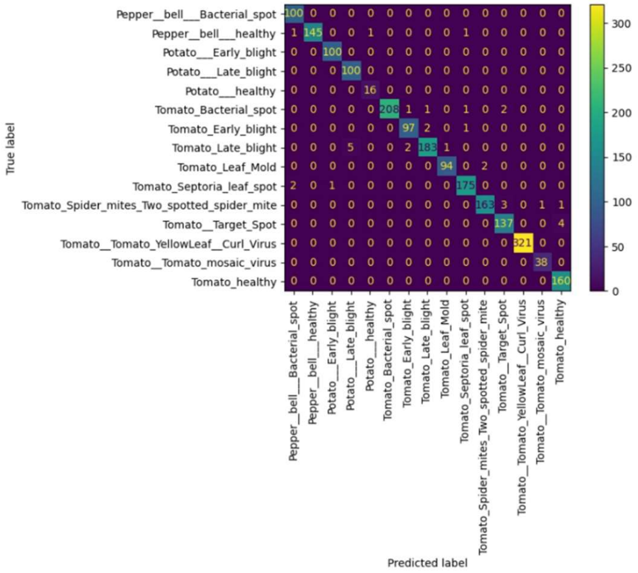


Fig. 7: Confusion matrix for binary classification (healthy vs. diseased).

Figure 7 trends support the observations from Figures 4 and 5, showing that the network stabilizes steadily after initial fluctuations. Minor peaks align with augmentation variance and learning-rate adjustments. Overall, the model exhibits robust convergence, consistently high accuracy, and minimal overfitting. The analysis confirms that an EfficientNet-like CNN trained from scratch can achieve near state-of-the-art performance on the PlantVillage dataset without pre-trained weights. Evidence from accuracy curves, loss curves, and confusion matrices demonstrates the practicality of this approach for offline and resource-constrained deployments.

6. Discussion

The outcomes of the experiment show that a small, barebone EfficientNet-like CNN is able to reach high levels of classification accuracy on the PlantVillage dataset without the need for utilizing pre-trained weights. According to the learning curves, augmentation and adaptive learning-rate schedules prove useful in reducing initial oscillations in validation performance, leading to stable convergence. The per-class confusion matrix indicates strong diagonal dominance, confirming that the majority of diseases are correctly diagnosed, while minor errors occur mainly between visually similar tomato leaf diseases. These findings suggest that performance may be further improved through additional class-specific augmentation or higher-resolution inputs.

From a deployment perspective, the lightweight nature and lack of external dependencies enable offline operation in resource-limited environments such as rural farms, classrooms, or agricultural extension offices with limited internet access. The small model size and fast inference make it well suited for edge devices and mobile platforms, supporting real-time disease monitoring and automated advisory services. Additionally, its ability to provide transparent and auditable predictions is valuable for real-world agricultural systems. Overall, this study demonstrates that carefully chosen design decisions in architecture, data augmentation, and optimization can effectively replace transfer learning while maintaining competitive accuracy, offering a practical and deployable solution for plant disease detection.

7. Limitations and Future Scope

7.1. Limitations

Controlled imagery: PlantVillage images have uniform backgrounds and lighting, which can overestimate real-world performance; domain shifts to field conditions with clutter, shadows, and occlusions may reduce accuracy.

Resolution ceiling: Training at 224 × 224 may miss small lesions or vein-level details, leaving fine-grained tomato dis ease distinctions prone to confusion.

Class imbalance: Some classes have low support, making per-class metrics sensitive and less stable.

From-scratch compute: De-novo training requires more epochs and careful scheduling than transfer learning, increas ing runtime on limited hardware.

Calibration unknowns: Validation loss suggests reasonable calibration, but explicit reliability measures (ECE/Brier) were not computed.

Single-dataset focus: Results are restricted to PlantVillage; cross-crop or cross-region generalization remains untested.

7.2. Future Scope

Field generalization: Evaluate on in-the-wild datasets with domain adaptation (feature alignment, style transfer, test-time adaptation) to bridge background and lighting gaps.

Higher resolution and multi-crop: Train at 384–512 or use multi-crop inference to capture subtle lesions, managing memory via mixed precision.

Class-aware objectives: Employ focal/class-balanced loss, hard-example mining, or minority-class augmentation to reduce tail errors.

Architecture refinements: Explore EfficientNet-V2, ConvNeXt-Tiny, selective fine-tuning, and stochastic weight averaging for flatter minima and better generalization.

On-device deployment: Practice quantization-aware training, post-schooling INT8 quantization, or expertise distillation to fulfill cell NPU/microcontroller constraints.

Uncertainty and interpretability: Measure ECE/Brier, apply temperature scaling or MC-dropout, and use Grad CAM++/RISE to verify lesion-centric attention.

Pipeline robustness. Benchmark under noise, blur, color, shifts, and use test-time augmentation or ensembling to improve reliability.

Human-in-the-loop: Integrate active learning for ambiguous field samples, prioritizing high-uncertainty or disagreement cases.

8. Conclusions

This work demonstrates that a compact EfficientNet-like CNN can be trained from scratch on PlantVillage images to achieve high accuracy (98.4%–98.9%) without relying on pre-trained weights or internet access. Careful scheduling, augmentation, and checkpointing stabilize convergence and improve generalization, even in a fully offline setup. The resulting model produces reliable per-class predictions, supported by a highly diagonal confusion matrix, and is suitable for resource-limited or air-gapped environments.

References

[1] C. Pal, A. Saha, and S. Roy, “A lightweight and explainable CNN model for empowering plant disease diagnosis on PlantVillage,” Scientific Reports, 2025, accuracy 99.45% on PlantVillage. [Online]. Available: Nature journal page.

[2] G. N. Balaji, S. Yadav, and R. Singh, “Using deep learning for plant disease detection and classification,” Nature Environment and Pollution Technology, vol. 24, no. 2, 2025, PlantVillage accuracy 98.2%.

[3] PlantVillage, “PlantVillage Dataset,” Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/emmarex/plantdisease

[4] L. Singh et al., “Automated Disease Detection in Plant Images using CNNs,” IEEE Int. Conf. Smart Multibrand Comput., 2022.

[5] G. S. Hukkeri, P. V. Rao, and S. Kumar, “Classification of Various Plant Leaf Disease Using Deep Learning,” Open Agriculture Journal, vol. 18, 2024.

[6] F. Naveed, M. A. Khan, and S. Ali, “Sustainable AI for plant disease classification using PlantVillage dataset,” Sustainable Computing: Informatics and Systems, vol. 7, 2025.

[7] S. Saxena, “Disease detection for potato, tomato and pepper plants using machine learning approaches,” Master’s Thesis, National College of Ireland, 2023.

[8] J. Chen et al., “Improved YOLOv5 network for plant disease recognition in natural habitats,” Computers and Electronics in Agriculture, 2022.

[9] M. A. Jasim and J. M. Al-Tuwaijari, “Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques,” Int. Conf. Computer Science and Software Engineering, 2020.

[10] K. N. Rahman et al., “A real time monitoring system for accurate plant leaves disease detection,” Sustainable Computing, 2025.

[11] Author et al., “Deep Learning VGG19 Model for Precise Plant Disease Classification,” Agricultural Science Digest, 2025.

[12] N. B. Korade, M. B. Salunke, A. A. Bhosle, et al., “Tomato Leaf Disease Detection with YOLOv8,” International Research Journal of Modernization in Science, vol. 6, no. 1, 2025.

[13] J. Sharma et al., “Deep learning based ensemble model for accurate tomato disease classification,” Scientific Reports, vol. 15, 2025.

[14] J. Zhang, Y. Rao, and S. Li, “Identification of cucumber leaf diseases using deep learning and small sample size for agricultural Internet of Things,” International Journal of Distributed Sensor Networks, 2021.

[15] Balajiet al., “Using Deep Learning for Plant Disease Detection and Classification,” Neptune Journal, 2024.

[16] Murugavalli et al., “Plant leaf disease detection using vision transformers for sustainable agriculture,” Scientific Reports, 2025.

[17] H. Sun, R. Fu, X. Wang, Y. Wu, M. A. Al-Absi, Z. Cheng, Q. Chen, and Y. Sun, “Efficient deep learning-based tomato leaf disease detection through global and local feature fusion,” Scientific Reports, vol. 15, 2025.