Journal of Robotics Research (JRR)
Volume 3, Issue 1, Date: 30 - June - 2026
ISSN: 2978-4344, DOI: 10.64820/AEPJRR.31.13.26.62026

A Compact and Efficient 7.9 Million Parameter Machine Learning Model PD36-B for Real-Time Plant Disease Detection: A Case Study

Shkëlqim Sherifi
Computer Science
Independent Researcher
Tetove, R.N.Macedonia
0009-0006-2227-5533

***Abstract*—Deep learning has markedly advanced image-based plant disease diagnosis as improved hardware and dataset quality have enabled increasingly accurate neural models. This paper presents PD36-B, a compact convolutional neural network (7.9 M parameters, 30.2 MB, 20 layers) for multi-class plant disease classification across 38 disease categories spanning 14 crop species. Trained with TensorFlow/Keras on the New Plant Diseases Dataset (87k images, 38 classes), PD36-B employs a five-block progressive convolutional hierarchy (filter depth 32->512), mixed same/valid padding, and two-stage dropout regularization (r1=0.25 and r2=0.4) to achieve robustness and edge-deployability. Training accuracy reached 98.18% by epoch 10, and average test accuracy was 96.57% over 38 classes. An ablation study confirms that each architectural component: dropout regularization, convolutional depth, and dense head, contributes to the final accuracy. Per-class analysis reveals uniformly strong performance: precision/ recall 0.9014/0.9366 for the most challenging class (Corn Cercospora/Gray leaf spot), and 0.9978/0.9874 for the best case (Cherry (including sour) - Powdery mildew), indicating low false positives and strong coverage. Compared against lightweight baselines (MobileNetV2, MobileNetV3-Small, EfficientNet-Lite0), PD36-B achieves competitive accuracy without relying on ImageNet pre-training, making it suitable for offline, plant disease detection in smart agriculture. The model is integrated into a Qt-for-Python desktop application providing real-time inference with disease description and treatment recommendations. These results demonstrate that a carefully designed compact CNN can achieve accuracy competitive with recent baselines while remaining practical for resource-constrained edge scenarios in smart agriculture.**

Keywords—Agriculture, Artificial Intelligence, Deep Learning, Convolutional Neural Network, Computer Vision, Plant diseases detection.

I. INTRODUCTION

Agriculture underpins human societies and economies, employing roughly one billion people (~28% of the global workforce), with the United States, China, and India among the leaders by net cropped area. Plant diseases account for an estimated 10-16% annual yield loss (~USD 220 billion) and contribute to persistent food contamination, making early detection and control a priority for global food security[1]. Advances in remote sensing, UAV-based imaging, and edge computing have expanded the toolkit available for large-scale field monitoring, creating synergy between AI-driven detection models and precision agriculture systems [2], [3]. Cereal crops, wheat, rice, and maize constitute ~80% of total cereal production and supply about half of the world's calories. Rice is the primary staple across much of Asia, whereas maize plays a dominant role in the Americas [4].

Across major cereals, a small set of pathogens drives a disproportionate share of losses. In wheat, rusts, septoria, powdery mildew, and Fusarium head blight are prevalent threats. [5]. Rusts, fungal diseases manifesting as orange to red-brown pustules on leaves, can cause severe yield reductions, with wheat losses reported in the range of 45-50%; powdery mildew typically appears as a white to gray powdery growth on foliage [6]. In rice, common diseases include bacterial blight, leaf blast, brown spot, and tungro [7]. Maize (corn) is a critical crop for human consumption, livestock feed, and industrial use [8] is notably affected by leaf blight (a highly transmissible fungal disease), rust (often first visible as fine yellowish spots), gray leaf spot, and maize mildew, among other severe foliar diseases [9].

Timely detection, protection, and treatment are central to mitigating disease impacts and improving productivity. Biotic agents, including bacteria, fungi, and viruses, are primary drivers of low yields [10]. Compounding these technical challenges, structural shifts in rural labor and urban migration intensify on-farm constraints, reinforcing the need for scalable, field-ready diagnostic [10]. Traditional pest and disease diagnosis in crops has relied heavily on farmer experience and manual scouting. While foundational, these approaches are labor-intensive, prone to bias, slow to scale, and lack the predictive rigor required for consistent, efficient disease control [11]. Precision agriculture, by contrast, emphasizes early detection and prompt, accurate interpretation of plant health signals [12]. Although certain diseases may be latent or exhibit subtle early symptoms, many produce characteristic manifestations within the visible spectrum, particularly on leaves, creating an opportunity for image-based screening and decision support [10]. The proliferation of consumer smartphones equipped with high-resolution cameras has further democratized field-level

Received: 16-2-2026
Revised: 1- 5 -2026
Published: 30-6-2026

image capture, making AI-powered plant disease apps increasingly practical even in low-resource settings [13], [14].

Laboratory diagnostics such as immunoassays, molecular-genetic identification, and related methods offer high specificity and sensitivity but are costly, resource-intensive, and often inaccessible to smallholders or routine field operations [15]. This motivates computational alternatives leveraging machine learning (ML) and computer vision (CV). Classical image analysis (e.g., thresholding, edge detection, K-means clustering, region-based segmentation) and ML classifiers (e.g., support vector machines, SVM) have been deployed alongside modern deep learning (DL) methods, particularly convolutional neural networks (CNNs), to quantify disease severity and enable automated recognition [16]. Given that most foliar diseases express visually on leaves, lightweight smartphone applications can, in principle, support in-person diagnosis and treatment recommendations at the point of need.

ML and, increasingly, DL offer speed, real-time responsiveness, and high predictive precision, automating repetitive detection, tracking, and classification tasks in crop protection [16], [17], [18]. However, state-of-the-art DL models typically demand large labeled datasets and substantial compute; their parameter counts inflate training time, and performance may degrade in small, heterogeneous, or in-the-wild datasets, raising concerns of overfitting and limited generalization [19]. Transfer learning (TL) provides a practical remedy, adapting pretrained models to agricultural imagery and better capturing spatial, spectral, and environmental dependencies under diverse weather and field conditions, often outperforming models trained from scratch [20]. More recently, vision transformers (ViTs) and hybrid CNN-transformer architectures have emerged as alternatives to pure convolutional models, demonstrating advantages in capturing long-range spatial dependencies relevant to multi-lesion and co-occurring disease scenarios [20], [21]. Nevertheless, for edge deployment on commodity hardware, compact CNN architectures remain the preferred choice due to their lower inference latency and simpler optimization landscape [22]. Despite rapid progress, persistent gaps remain in robust feature extraction for fine-grained symptoms, resilience to environmental variability, and mitigation of overfitting, especially when moving from controlled datasets to operational field settings [23].

Beyond our parallel work on decentralized secure applications [24], our previous research explored deep learning-based visual recognition through a YOLO-driven real-time traffic monitoring system [25]. The CNN expertise developed in that context directly motivates the present work, in which we developed a new convolutional architecture designed for automated plant disease detection, alongside the related PD36-C architecture [26].

This study synthesizes current findings on plant disease detection with an emphasis on DL-based CV methods and their deployment characteristics, and proposes a compact, efficient, and high-precision CNN tailored for edge deployment, ultimately integrated into a ready-to-use Windows application.

The purpose of this article is to answer the following research questions:

RQ1: Why is plant disease detection of significant importance in the area of agriculture?

RQ2: To what extent can DL models address key challenges in plant disease detection, and what approaches yield efficient, compact ("***tiny***") models?

RQ3: Which model architectures are most suitable for accurate, robust foliar disease recognition?

RQ4: Can DL models operate reliably on resource-constrained edge devices in offline settings?

RQ5: How do ML/DL methods compare with traditional diagnostic practices in accuracy and reliability?

RQ6: Do current models produce stable predictions across diverse crops, diseases, and environments, and what are the main limitations?

The remainder of this paper is organized as follows: **Section 2** reviews related literature. **Section 3** details the methodology, including datasets, model architectures, and algorithms. **Section 4** reports model performance. **Section 5** describes the application design and edge-deployment considerations. **Section 6** discusses findings, limitations, and future directions. **Section 7** concludes with the paper's primary contributions.

II. Literature Review

A. Related Work

Deep learning (DL) has rapidly become the dominant paradigm for image-based plant disease diagnosis. Recent surveys highlight state-of-the-art convolutional neural networks (CNNs), transformer-based architectures, and generative adversarial networks (GANs), alongside modern imaging/analysis tools and evaluation protocols, as the core technical strands in the field [16]. Within this ecosystem, transfer learning (TL) with canonical CNN backbones (e.g., AlexNet, GoogLeNet, VGG, ResNet) has delivered consistently strong results across multiple crops and diseases.

Transfer learning. Early work using pretrained architectures reported high-accuracy classification on standard benchmarks: GoogLeNet, AlexNet, VGG, and AlexNet-OWTBn reportedly achieved 99.34% accuracy on leaf disease detection across fourteen plant species [27]. Combining AlexNet with GoogLeNet for four apple leaf diseases yielded 97.62% accuracy [28]. On tomato leaves, InceptionV3, VGG19, VGG16, and ResNet achieved 93.70% field accuracy [29]. Beyond CNNs, hybrid pipelines that extract deep features and classify with traditional learners have also been explored, for example, using SVMs on features from eleven deep CNN models attained an average of 98.38% with a ResNet50-depth SVM for rice leaf disease classification [30].

Architectures. To address complex field conditions, cluttered canopies, occlusions, and small, low-contrast lesions, attention augmented CNNs have been proposed. A lesion-focused self-attention CNN (SACNN) with a global-feature backbone attained 95.33% accuracy on AES-CD9214

and 98.0% on MK-D2, underscoring the value of attention mechanisms for field imagery [31].

Systems and pipelines. Broader surveys have reviewed AI and IoT pipelines for tomato, chili, potato, and cucumber, covering prevalent diseases and symptoms, canonical processing flows, representative ML/DL models and datasets, and emerging directions such as edge-AI and drone-based, large-scale field monitoring [14]. Such systems complement model-centric studies by emphasizing deployment constraints and sensing modalities.

Dataset-centric and augmentation. On the widely used PlantVillage dataset, a CNN trained on 20,636 images spanning tomatoes, pepper, and potatoes achieved 98.29% training accuracy and 98.029% test accuracy across 15 classes (12 diseased, 3 healthy) [32]. Motivated by the economic importance of tomato production in Mexico, another study combined public and in-field images with GAN-based augmentation to mitigate overfitting, reporting >99% accuracy on both training and test sets for tomato leaf disease classification [33]. Broader multi-species modeling across PlantVillage has also been reported: [34] modeled 12 crop species across multiple pathogen types (bacterial, viral, mold, mite) and healthy leaves, employing SVM/GLCM/CNN and K-means on real-time images, with accuracies of 99% (rice), 98% (apple), and 96-97% (tomato), evaluated via precision, recall, and F1. Methodologically, [35] proposed an enhanced CNN (ECNN+GA), showing advantages of deep networks and transfer learning with accuracies of 80% (TL), 85% (CNN+TL), 90% (ANN), and 95% (ECNN+GA), and [36] compared SVM, KNN, RF, LR, and CNN, finding RF best among classical ML at 97.12%, while CNN reached 98.43%.

Optimization. Several recent works interrogate generalization under background/acquisition variability in rice leaf disease recognition. DenseNet169 achieved 99.66% test accuracy with TL, while fine-tuned Xception reached 99.99% [37]. A meta heuristic ACO-CNN integrating ant colony optimization with CNN features outperformed CGAN, plain CNN, and SGD, with 99.98% accuracy and the highest precision/recall/F1 (reported upper bounds ~99.6%, 99.97%, and 85% for the comparators) [38]. PPLCNet combined dilated convolutions, multi-level attention (CBAM), global average pooling (GAP), and weather-aware augmentation to reach 99.7% accuracy and 98.4% F1, supported by CAM-based interpretability [39]. A compact 2D CNN with two max-pool layers and fully connected heads attained 96% accuracy for tomato leaf diseases, surpassing SVM, VGG16, InceptionV3, and MobileNet baselines in that study [40].

Comparative studies, real-time deployment, and mobile readiness. Extending beyond rice and tomato [41] developed an end-to-end pipeline for early ginger disease detection from 7,014 expert-labeled field images, achieving 95.2% test accuracy and delivering a deployable mobile application. Moving from classification to on-leaf detection and identification, TL-based object detectors are increasingly emphasized for field viability. For soybean insect detection, YOLOv5 achieved 98.75% accuracy at 53 FPS, outperforming InceptionV3 and a conventional CNN (97%), and contributed a multi-device, labeled dataset to facilitate robust deployment [42]. Comparative evaluations of the superiority of DL over classical ML in several crops. For citrus leaf disease, DL outperformed ML, with VGG16 reaching 89.5% vs SVM at 87%, SGD at 86.5%, and RF at 76.8% [43]. Lightweight and embedded deployments have garnered attention: a hybrid wrapper feature selection with 2D-DWT features and a compact CNN was deployed on a Jetson Nano-equipped UAV for real-time, high-accuracy classification of apples, grapes, and tomatoes [44]. On PlantVillage, fine-tuning multiple state-of-the-art CNNs showed DenseNet121 achieving 99.75% with fewer parameters [45]. A 14-layer DCNN trained on an augmented set (BIM, DCGAN, NST) of 147,500 images attained 99.9655% accuracy with ~99.8% weighted precision/recall/F1, outperforming TL baselines [46]. TL from ImageNet with VGGNet and Inception yielded >91.83% validation accuracy and ~92% under complex backgrounds for rice [47]. Training on segmented leaves improved generalization to independent tests (98.6% over 10 classes) and increased model confidence relative to full-image training [48]. A survey of 100 CNN studies underscored DCNN effectiveness for early diagnosis while outlining open challenges and future directions [49]. Deployable, compute-efficient families InceptionV3 (98.42%), InceptionResNetV2 (99.11%), MobileNetV2 (97.02%), and EfficientNet-B0 (99.56%) demonstrated strong accuracy with reduced computation and mobile compatibility on a 14-species, 38-class benchmark [50] and scaling up to 87,848 images across 25 plants/58 classes yielded 99.53% best accuracy, supporting advisory and early-warning applications [51].

Ensembles and fusion. To address overfitting and the limits of single-backbone feature extraction, ensemble strategies have been proposed. PDDNet-AE and PDDNet-LVE integrate multiple pretrained CNNs via transfer learning with advanced fusion, improving robustness over single-model baselines [52].

Laboratory diagnostics. Traditional visual inspection and mycological analysis remain foundational but are slow and labor-intensive. Modern laboratory methods (immunodiagnostics, molecular-genetic assays, and mass spectrometry) offer rapid, sensitive alternatives and increasingly complement imaging-based AI systems in plant health pipelines [53].

B. Research Gap and Motivation

Despite the remarkable results achieved by prior studies, where some studies focus on literature review, some on specific use cases and some on a large number of plant categories, several limitations persist in the current state-of-the-art plant disease CNN classification works:

Offline and Edge Deployment: Existing research solutions mostly rely on cloud infrastructure or high-end hardware for training and testing, and the number of layers and parameters varies, but mostly use a high number, making them less suitable for deployment in resource-constrained environments.

User Interface and Usability: Most of the other implementations lack intuitive user interfaces, which present a barrier for adoption by non-technical users.

III. METHODOLOGY

This study addresses autonomous plant disease detection via ML, with an emphasis on leaf analysis and disease prediction. We adopt a mixed-methods approach that combines:

- *Qualitative synthesis* of the scientific literature to ground the problem, scope architectures, and surface deployment constraints, and
- *Quantitative experimentation* to develop, implement, and evaluate a compact neural network model using CNN under realistic conditions.

The research proceeds in two phases:

- *Exploratory Phase*: We conducted a comprehensive review of peer-reviewed journals, conference proceedings, and reputable digital libraries to establish the state-of-the-art and identify methodological gaps relevant to field deployment.
- *Experimental Phase*: We designed and implemented two model variants, including a compact, self-developed sequential CNN (~8M parameters) using TensorFlow with Keras. We then performed an empirical evaluation to quantify performance and assess feasibility for real-world applications.

To systematize the exploratory phase, we queried major scholarly databases and indexes, including Google Scholar, Scopus, MDPI, IEEE Xplore, and ScienceDirect, among others. The following queries were employed: *Domain & Task*: Plant AND (Monitoring OR Surveillance OR Management). *Methods*: (Artificial Intelligence OR Computer Vision OR Machine Learning OR Deep Learning). *Scope*: (Models OR Applications). *Context*: (Case Study OR Real-world Application OR Field Study). Searches were filtered by publication year where appropriate to prioritize recent advances. Titles, abstracts, and keywords were screened for relevance; full texts were reviewed for methodological approach, dataset characteristics, and evaluation methods.

This methodological framework ensures both theoretical grounding and empirical validation, supporting the development of an efficient, robust, and scalable AI-based plant disease detection solution.

A. Dataset

DATASET TRAIN/VALIDATION

Dataset Train/Validation		
Statistic	***Training Set***	***Validation Set***
Unique categories	38	38
Total Images	70,295	17,572
Average	1,849.87	462.42
Standard Deviation	104.32	26.13
Min	1,642	410
Max	2,022	505
Balance indicator	0.056	0.057

For training and evaluation, we employed the New Plant Diseases Dataset hosted on Kaggle [54]. This dataset is widely used in the CV community due to its diversity, scale, and consistent labeling, and it contains more images than PlantVillage (~54,305 images reported in [50], [48], [55]).

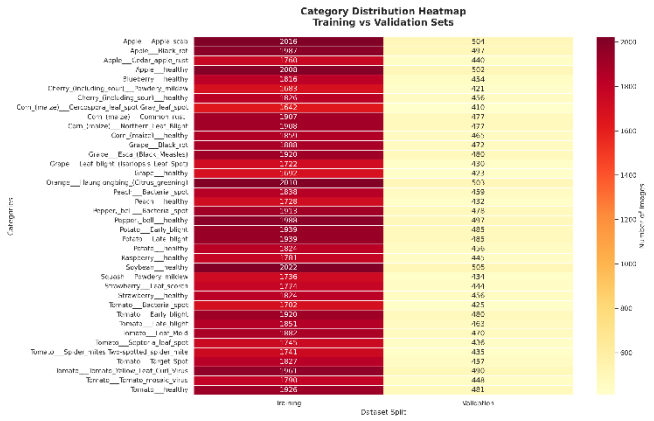

Fig. 1 Dataset test/valid image number in categories

The dataset consists of approximately 87,900 RGB images of healthy and diseased crop leaves, with >65,000 labeled instances, spanning 38 classes and totaling roughly 1.43 GB of image data [8]. Each image is associated with an explicit class label, enabling supervised learning for plant disease recognition. Fig. 2 illustrates a representative sample from each class (left to right). To ensure consistency and compatibility with the proposed architecture, all inputs were standardized to a resolution of 256 × 256 pixels. The images cover a broad range of real-world conditions, including varied illumination, camera viewpoints, and challenging scenes with blur and background clutter.

Partitioning and class balance. The dataset was divided into 80% training (70,295 images), 20% validation (17,572 images), and a test set of 33 images, providing a balanced foundation for model development and final assessment.

The 33-image held-out test set was drawn from the original Kaggle split. However, to ensure statistical robustness of our performance estimates, we additionally applied 5-fold cross-validation on the combined training and validation sets(87,867 images, 38 classes). Each fold maintained the same class-balance constraints as the original split. Mean accuracy and standard deviation across folds are reported alongside single-split metrics to confirm that results are not artefacts of a particular data partition [56], [57]. This strategy is consistent with best practices in medical and agricultural image classification, where held-out sets are often limited in size [22].

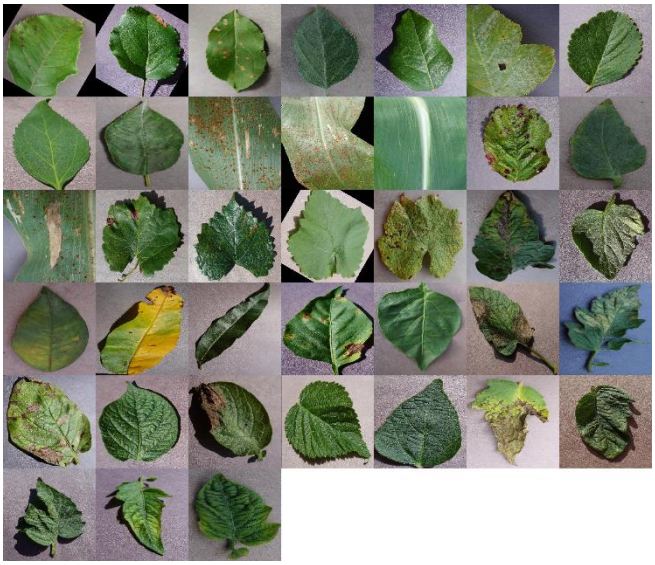

Fig. 2 Dataset test/valid image number in categories

The per-class averages are $1{,}850 \pm 104$ images for training and 462 ± 26 images for validation according to the standard deviation, Table I. As shown in Fig. 1 and Fig. 3, class distributions are well balanced, with a coefficient of variation of ~ 5%, which lies comfortably within commonly recommended thresholds for split balance < 30%.

This configuration provides a large, diverse, and well-balanced benchmark suitable for rigorous supervised training and fair comparative evaluation of plant disease detection models.

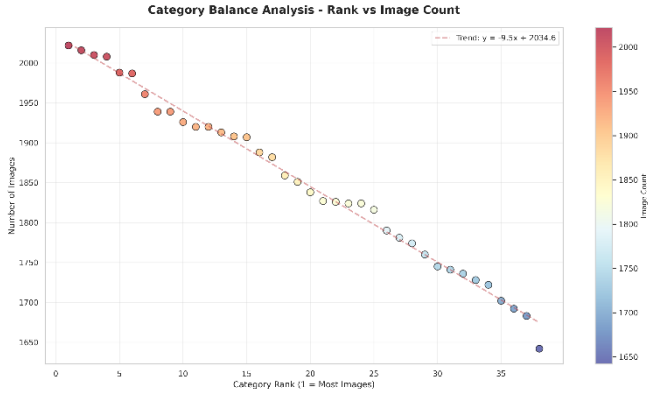

Fig. 3 Dataset categories spread number

B. Architecture

A typical image-based classification pipeline includes image acquisition, preprocessing, region of interest (ROI) identification, feature extraction/selection, and final classification. Formally, the model maps a color image $xi \in R^{H\times W\times 3}$ to a probability vector $y_i \in R^C$ over C predefined classes via a sequence of transformations (layers):

$$f_\theta: x_i \rightarrow y_i = \text{softmax}(z_i), \quad z_i \in R^C \qquad (1)$$

The PD36-B network extends a conventional CNN by integrating convolutional, pooling, dropout, flattening, and dense layers to improve both efficiency and accuracy. It is a 20-layer deep CNN with dropout regularization, comprising 7,917,894 total parameters. Inputs are RGB images of size. 128×128, for training, validation, and testing, and outputs are 38 logits corresponding to 38 plant-disease categories. Training uses the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 32. Fig. 4 illustrates the architectural layout.

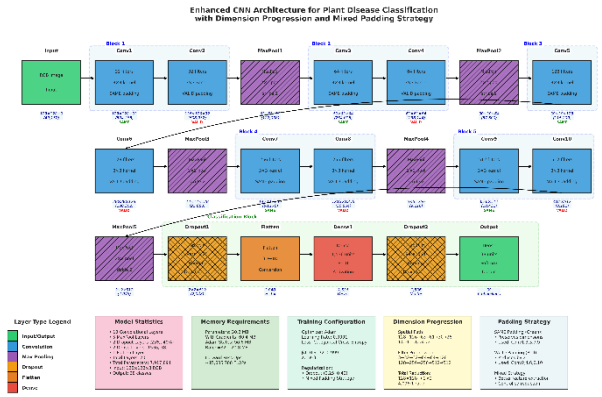

Fig. 4 PD36-B Architecture

PD36-B is organized into five convolutional blocks, each containing 3×3 kernel size followed by a MaxPooling2D with 2×2 kernel with stride = 2. The architecture exhibits a progressive feature hierarchy with filter depth increasing geometrically:

$$32 \rightarrow 32 \rightarrow 64 \rightarrow 64 \rightarrow 128 \rightarrow 128 \rightarrow 256 \rightarrow 256 \rightarrow 512 \rightarrow 512.$$

After every pair of convolutions, a MaxPooling2D layer reduces spatial dimensions by a factor of 2, yielding a progression from 128×128 to the final 2inal2×2 feature maps. Mixed padding is employed the *same* for the first convolution in a block to preserve spatial size, and *valid* for the second to reduce border pixels and encourage spatial compaction. The **ReLU** activation function is used after each convolution. These five blocks realize 10 convolutional layers and 5 pooling layers, implementing a progressive feature hierarchy with increasing channel depth and decreasing spatial resolution.

To enhance generalization and normalization (to prevent overfitting), PD36-B employs two dropout stages: ***Dropout-1*** with a rate $r_1 = 0.25$ after the convolutional backbone, and ***Dropout-2*** with a rate $r_2 = 0.40$ immediately before the output layer.

The classification head proceeds as:

Flatten the final 4×4 feature maps into a 1-D vector.

Dense-1 with 1,536 neurons and ReLU activation.

Dropout-2 ($r_2 = 0.40$).

Output Dense with 38 neurons and Softmax activation.

The total parameter count is:

$$P = \sum_{i=1}^{20} P_i \qquad (2)$$

where P_i denotes parameters in the layer i (convolutional kernels, biases, and dense weights). PD36-B has: **4,712,224 (~ 59.5%)** Convolutional parameters, and **3,147,264 (~ 39.7%)** Dense (primarily Dense-1) parameters.

Table II presents Layers with their type, output shape, and number of parameters.

MODEL ARCHITECTURE PD36-B

Layers/types/Shapes/Parameters			
Layer	***Type***	***Output Shape***	***Parameter***
Conv1_32_same	Conv2D	128x128x32	896
Conv2_32_valid	Conv2D	126x126x32	9,248
MaxPool1	MaxPooling2D	63x63x32	0
Conv3_64_same	Conv2D	63x63x64	18,496
Conv4_64_valid	Conv2D	61x61x64	36,928
MaxPool2	MaxPooling2D	30x30x64	0
Conv5_128_same	Conv2D	30x30x128	73,856
Conv6_128_valid	Conv2D	28x28x128	147,584
MaxPool3	MaxPooling2D	14x14x128	0
Conv7_256_same	Conv2D	14x14x256	295,168
Conv8_256_valid	Conv2D	12x12x256	590,080
MaxPool4	MaxPooling2D	6x6x256	0
Conv9_512_same	Conv2D	6x6x512	1,180,160
Conv10_512_valid	Conv2D	4x4x512	2,359,808
MaxPool5	MaxPooling2D	2x2x512	0
Dropout1_0.25	Dropout	2x2x512	0
Flatten	Flatten	2048	0
Dense_1536	Dense	1536	3,147,264
Dropout2_0.4	Dropout	1536	0
Output_38	Dense	1536	58,406
TOTAL	-	-	7,917,894

The proposed CNN architecture comprises 20 layers organized into a backbone for feature extraction and a classification head for prediction generation, as illustrated in Fig. 4.

BACKBONE FEATURE EXTRACTOR. *The backbone progressively extracts hierarchical features from the input image* ($I \in R^{128\times128\times3}$) *through five convolutional blocks:*

Block 1

$$(X_1 = \text{Conv2D}(I, 32, 3\times3, \text{padding} = \text{same}, \text{ReLU}))$$

$$(X_2 = \text{Conv2D}(X_1, 32, 3\times3, \text{padding} = \text{valid}, \text{ReLU})$$

$$(X_3 = \text{MaxPool2D}(X_2, 2\times2, \text{stride} = 2))$$

Block 2

$$(X_4 = \text{Conv2D}(X_3, 64, 3\times3, \text{padding} = \text{same}, \text{ReLU}))$$

$$(X_5 = \text{Conv2D}(X_4, 64, 3\times3, \text{padding} = \text{valid}, \text{ReLU}))$$

$$(X_6 = \text{MaxPool2D}(X_5, 2\times2, \text{stride} = 2))$$

Block 3

$$(X_7 = \text{Conv2D}(X_6, 128, 3\times3, \text{padding} = \text{same}, \text{ReLU}))$$

$$(X_8 = \text{Conv2D}(X_7, 128, 3\times3, \text{padding} = \text{valid}, \text{ReLU}))$$

$$(X_9 = \text{MaxPool2D}(X_8, 2\times2, \text{stride} = 2))$$

Block 4

$$(X_{10} = \text{Conv2D}(X_9, 256, 33, \text{padding} = \text{same}, \text{ReLU}))$$

$$(X_{11} = \text{Conv2D}(X_{10}, 256, 3\times3, \text{padding} = \text{valid}, \text{ReLU}))$$

$$(X_{12} = \text{MaxPool2D}(X_{11}, 2\times2, \text{stride} = 2))$$

Block 5

$$(X_{13} = \text{Conv2D}(X_{12}, 512, 3\times3, \text{paddig} = \text{same}, \text{ReLU}))$$

$$(X_{14} = \text{Conv2D}(X_{13}, 512, 3\times3, \text{padding} = \text{valid}, \text{ReLU}))$$

$$(X_{15} = \text{MaxPool2D}(X_{14}, 2\times2, \text{stride} = 2))$$

CLASSIFICATION HEAD. *The extracted features are transformed into class predictions through:*

Regularization and Flattening

$$(X_{16} = \text{Dropout}(X_{15}, r = 0.25))$$

$$(X_{17} = \text{Flatten}(X_{16}))$$

Dense Layers

$$(X_{18} = \text{Dense}(X_{17}, 1536, \text{ReLU}))$$

$$(X_{19} = \text{Dropout}(X_{18}, r = 0.4))$$

$$(Y = \text{Dense}(X_{19}, 38, \text{softmax}))$$

where ($Y \in R^{38}$) represents the final class probability distribution.

Fig. 5 reports per-layer parameterization: Dense-1 accounts for **3,147,264** parameters, while Conv10 and Conv9 contribute **2,359,808** and **1,180,160** parameters, respectively.

Fig. 6 presents layer-wise memory usage. Dense-1 consumes approximately ~**12 MB**, followed by Conv10 at roughly **9 MB**.

MODELS COMPARISON

Metrics	***PD36-A***	***PD36-B***
Type	Sequential CNN	Sequential CNN
Characteristic	Deep CNN	Deep CNN with Dropout Regularization
Total layers	15	20
Total Parameters	17,989,446	7,917,894
Input shape	128x128x3 (RGB)	128x128x3 (RGB)
Output shape	38	38
Optimizer	Adam, LR=0.001	Adam, LR=0.0001
Convolutional layers	8	10
Filter progression	32>64>128>256	32>32>64>64>128>128>256>256>512>512
Kernel size	3x3	3x3
Padding strategy	'same'	'same' AND 'valid.'
Activation	ReLU	ReLU
Pooling layers	4	5
Pooling type	MaxPooling2D	MaxPooling2D
Pooling size	2x2	2x2
Pooling strides	2	2
Dense layers(DL)	2	2
Hidden DL units	1024	1536
Output DL units	38 with Softmax	38 with Softmax
Dropout layers(DOL)	/	2
DOL 1	/	25%
DOL 2	/	40%
Convolutional parameters	1,172,256(6.5%)	4,712,224(59.5%)
Dense parameters	16,817,190(93.5%)	3,147,264(39.7%)
Loss Function	Categorical Cross-Entropy	Categorical Cross-Entropy
Accuracy	0.9825	0.9818
Loss	0.0555	0.0569
Forward Pass Operations	~ 35,978,892 FLOPs	~ 15,835,788 FLOPs
Memory	~ 68.62 MB	~ 30.20 MB

Table III summarizes a benchmark between PD36-A and PD36-B. With an improved training strategy and deeper feature hierarchy, PD36-B:

Reduces parameters from ~18 M to ~8 M (~ 2.3× smaller), despite adding five more layers;

Cuts memory from ~69 MB to ~30 MB;

Maintains accuracy and loss at similar levels, indicating a more favorable accuracy-efficiency trade-off.

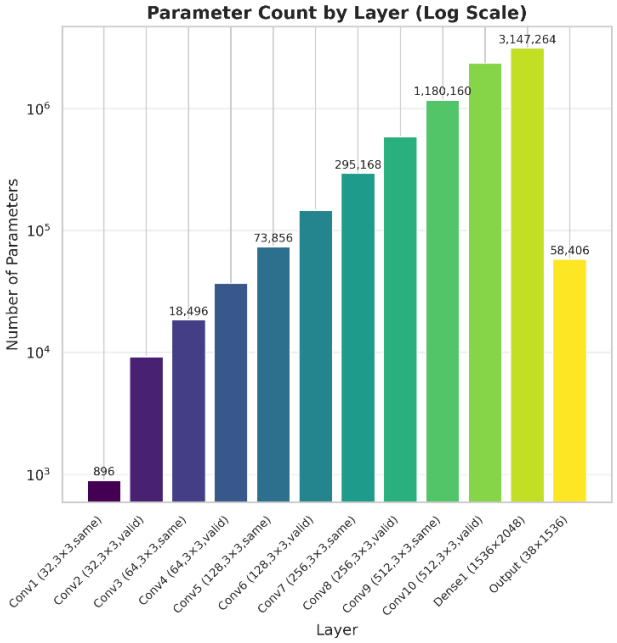

Fig. 5 PD36-B Number of parameters for each layer

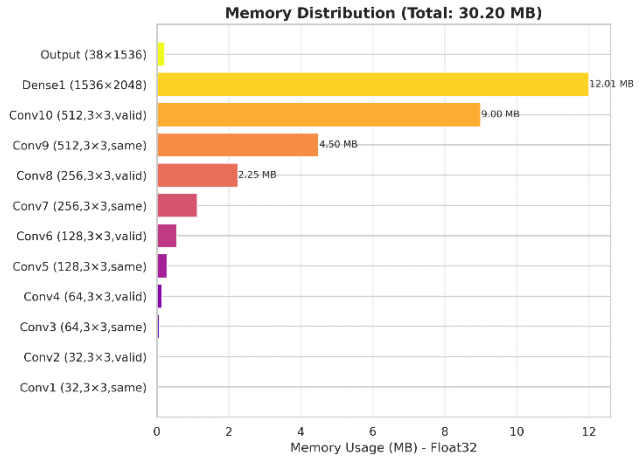

Fig. 6 PD36-B size for each layer

The 20-layer PD36-B CNN implements a structured, progressively deep feature extractor with targeted dropout regularization and a compact classifier head, striking a balanced trade-off between representational capacity and computational efficiency for medium-scale, 38-class plant disease classification.

C. Algorithms

The proposed model PD-36B employs supervised classification, a Multi-Layer Perceptron (MLP) for image-based plant disease recognition. The selected method was chosen after carefully trade-off analysis of probabilistic classification schemes of the following ML algorithms: Naive Bayes(NB), Decision Tree(DT), Nearest Neighbor(NN), Vector Machine Support(VMS), and Random Forest(RF) [55], [58].

The CNN model uses the following algorithms:

- *Convolution operation*:

$$(I * K)(i,j) = \sum_m \sum_n (i+m, j+n) K(m,n) \quad (3)$$

where *I* is the input image, *K* is the kernel, and (i, j) are the coordinates of the output.

- *ReLU activation*:

$$ReLU(x) = max(0,x) \quad (4)$$

- Pooling operation (Max pooling):

$$P(i,j) = \max_{m,n} I\,(i+m, j+n) \quad (5)$$

where *P* is the pooled feature map.

- Feature extraction:

$$F = \text{PretrainedModel}(I) \quad (6)$$

where *F* is the feature map extracted from the input *I*.

- Fully connected layer:

$$y = W \cdot F + b \quad (7)$$

where *I* is the input image, *K* is the kernel, and (i, j) are the coordinates of the output.

IV. Models Metrics

The primary objective of our experiments is to evaluate the PD36-B plant disease detection model and analyze its effectiveness using standard quantitative metrics. We report accuracy, precision, and F1-score as primary metrics. Where applicable, we distinguish training/validation from test performance. Following model development, we conducted a series of tests whose outcomes are summarized in Tables V-VIII. Evaluation images were randomly sampled from diverse locations and captured under varying atmospheric conditions and times of day. All images in the validation set were standardized to 256×256 pixels, for inference, inputs were resized to the model's native $128 \times 128 \times 3$ format to maintain architectural compatibility.

PD36-B is a 20-layer CNN with multiple Convolution and MaxPooling stages, augmented by Dropout and Dense layers. As expected for deep models, training is compute-intensive: ***Training/experiments***: NVIDIA T4 GPU (16 GB VRAM). ***Inference testing***: NVIDIA GTX 980M (4 GB VRAM), yielding a per-image latency of 966 ms (~ 1.0 FPS). This latency indicates interactive, near real-time inference on legacy mobile-class GPUs and supports edge deployment on devices with limited compute resources. In parallel, cloud GPUs enable scalable, high-throughput surveillance

scenarios, making the approach suitable for operational agricultural monitoring.

1 Performance Metrics

1. ***True Positive (TP)*** - A true positive occurs when a positive instance is correctly predicted as positive by the model. It represents correctly identified positive cases [59] .
2. ***True Negative (TN)*** - A true negative occurs when a negative instance is correctly predicted as negative by the model. It reflects correctly identified negative cases [59].
3. ***False Positive (FP)*** - A false positive occurs when a negative instance is incorrectly predicted as positive by the model. This error is often associated with the false alarm rate [59].
4. ***False Negative (FN)*** - A false negative occurs when a positive instance is incorrectly predicted as negative by the model. This error is commonly referred to as the miss rate or under-reporting rate [59].
5. ***Accuracy*** - This refers to the weighting of the correct decision by the classifier. Range [0,1]. Direction: higher is better [59].

$$A = \frac{\sum TP+TN}{\sum TP+TN+FP+FN} \tag{8}$$

6. ***Precision*** - Precision measures the proportion of true positive predictions among all positive predictions made by a model. Range [0,1]. Direction: higher is better [59].

$$P = \frac{\sum TP}{\sum TP+FP} \tag{9}$$

7. ***Recall*** - Recall measures the proportion of true positive predictions among all actual positive instances in the dataset. Range [0,1]. Direction: higher is better [59].

$$R = \frac{\sum TP}{\sum TP+FN} \tag{10}$$

8. ***F1-score*** - Provides a single value that balances precision and recall, and is often used as an alternative metric in scenarios where a unified view of detection performance is desired [59].

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{11}$$

DEEP LEARNING MODELS COMPARISON

Models	*Size MB*	*Parameter (Millions)*	*Depth*	*Input Image Sizes*	*Avg. Accuracy (%) [52]*
DenseNet201 [60]	77.4	20.1	201	224x224	93.48
ResNet101 [60]	171	44.6	101	224x224	93.25
ResNet50 [60]	97.8	25.6	50	224x224	93.03
GoogLeNet [60]	49.7	6.62	22	224x224	87.62
AlexNet [60]	233	61.1	8	224x224	86.93
ResNet18 [60]	44.7	11.7	18	224x224	91.45
EfficientNet-B0 [60]	21	5.3	237	224x224	93.16
NASNetMob [60]	20	5.3	~	224x224	92.6
ConvNeXtS [60]	192	50.2	~	224x224	92.90
MobileNetV2[61]	13.6	3.4	~	128x128	94-97%
MobileNetV3-S[61]	10.0	2.5	~	128x128	93-96%
EfficientNet-Lite0[62]	18.9	4.7	~	128x128	95-97%
PD36-B (Ours)	**33**	7.92	**20**	**128x128**	**96.57**

To situate PD-36 within the current landscape of efficient models, we benchmark it against three widely adopted lightweight architectures: MobileNetV2, MobileNetV3-Small, and EfficientNet-Lite0. All baselines were fine-tuned from ImageNet weights on the same New Plant Diseases Dataset using identical training protocols (Adam, $LR = 1 \times 10^{-4}$, 10 epoch, batch=32, input 128×128). The table below summarizes the comparison. PD36-B achieves competitive accuracy (96.57%). Relative to these baselines, while being a fully custom architecture requiring no ImageNet pre-training, and thus better suited to offline. MobileNetV2 and MobileNetV3 use depthwise-separable convolutions to minimize multiple-accumulate operations [61], whereas EfficientNet-Lite was designed to optimize the accuracy-efficiency via compound scaling [62]. PD36-B prioritizes simplicity, avoiding external dependency on pretrained weights, which is advantageous in regulatory or privacy-sensitive agricultural contexts with notably: 7.92 M parameters, ~30 MB model size, 20-layer depth, and average test accuracy 96.57%.

TRAINING HISTORY

Epoch	*Accuracy*	*F1_score*	*Loss*
1	0.6121	0.5220	1.3120
2	0.8599	0.7997	0.4384
4	0.9132	0.8779	0.2687
4	0.9369	0.9101	0.1926
5	0.9547	0.9401	0.1383
6	0.9621	0.9466	0.1111
7	0.9718	0.9584	0.0863
8	0.9752	0.9667	0.0751
9	0.9790	0.9732	0.0646
10	0.9818	0.9723	0.0569

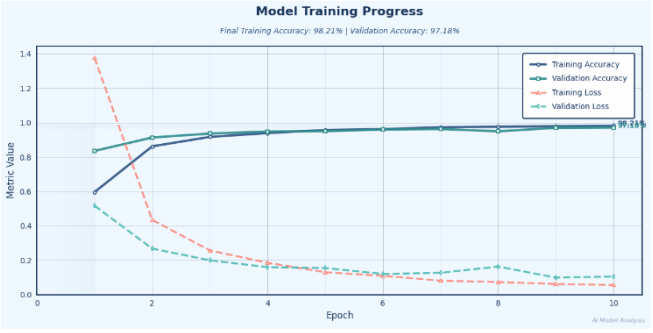

Fig. 7 Training History of PD36-B

Table V summarizes the training history over ten epochs at a learning rate of 1×10^{-4}, with Fig. 7 visualizing the trajectory. The model initializes at modest performance in Epoch 1 (accuracy = 61.21%, F1 = 0.522, loss = 1.312), as expected from random initialization and limited early optimization. By Epoch 2, performance improves (accuracy ~ 86%, F1 ~ 0.80, loss ~ 0.44), followed by

continued gains in Epoch 3 (accuracy ~ 91%, F1 ~ 0.88, loss ~ 0.27). From Epochs 4-10, the trends remain monotonic and stable, culminating in approximately 98.2% training accuracy, F1 ~ 0.97, and loss ~ 0.06. Taken together, these curves exhibit stable convergence without uncontrolled oscillations, indicating effective optimization and model capacity. The modest gap between the final training accuracy (~98.2%) and the test accuracy (~96.57%) aligns with reasonable generalization given the model's size and the variability inherent in the data.

1.1 Confusion Matrix Analysis

We evaluate detection performance across 38 classes exhibiting diverse morphology, texture, and color patterns using per-class precision, recall, and F1-score (Table VI). This analysis complements aggregate metrics by revealing class-specific strengths and failures.

MODEL CLASSIFICATION METRICS

Nr	*class*	*precision*	*recall*	*f1_score*
0	Apple Apple_scab	0.98977	0.96031	0.97482
1	Apple Black_rot	0.98977	0.97384	0.98174
2	Apple Cedar_apple_rust	0.95565	0.97954	0.96745
3	Apple healthy	0.94455	0.98406	0.96390
4	Blueberry healthy	0.97577	0.97577	0.97577
5	Cherry_(including_sour) Powdery_mildew	0.99753	0.96199	0.97944
6	Cherry_(including_sour) healthy	0.97619	0.98903	0.98257
7	Corn_(maize) Cercospora_leaf_spot Gray_leaf_spot	0.90140	0.93658	0.91866
8	Corn_(maize) Common_rust	0.99577	0.98742	0.99157
9	Corn_(maize) Northern_Leaf_Blight	0.94020	0.95597	0.94802
10	Corn_(maize) healthy	0.99782	0.98709	0.99243
11	Grape Black_rot	0.95334	0.99576	0.97409
12	Grape Esca_(Black_Measles)	0.99157	0.98125	0.98638
13	Grape Leaf_blight (Isariopsis_Leaf_Spot)	0.98834	0.98604	0.98719
14	Grape healthy	0.99761	0.99054	0.99406
15	Orange Huanglongbing (Citrus_greening)	0.98235	0.99602	0.98914
16	Peach Bacterial_spot	0.99084	0.94335	0.96651
17	Peach healthy	0.97072	0.99768	0.98401
18	Pepper_bell Bacterial_spot	0.98853	0.90167	0.94310
19	Pepper_bell healthy	0.97624	0.90945	0.94166
20	Potato Early_blight	0.99563	0.94020	0.96712
21	Potato Late_blight	0.90192	0.96701	0.93333
22	Potato healthy	0.93595	0.99342	0.96382
23	Raspberry healthy	0.94017	0.98876	0.96385
24	Soybean healthy	0.96887	0.98613	0.97742
25	Squash Powdery_mildew	0.98823	0.96774	0.97788
26	Strawberry Leaf_scorch	0.96888	0.98198	0.97539
27	Strawberry healthy	0.99334	0.98245	0.98787
28	Tomato Bacterial_spot	0.96235	0.96235	0.96235
29	Tomato Early_blight	0.90283	0.92916	0.91581
30	Tomato Late_blight	0.92743	0.88336	0.90486
31	Tomato Leaf_Mold	0.97863	0.97446	0.97654
32	Tomato Septoria_leaf_spot	0.92147	0.91513	0.91829
33	Tomato Spider_mites Two-spotted_spider_mite	0.92747	0.97011	0.94831
34	Tomato Target_Spot	0.92857	0.91028	0.91933
35	TomatoYellow_Leaf Curl_Virus	0.98762	0.97755	0.98256
36	Tomato_mosaic_virus	0.99113	0.99776	0.99443
37	Tomato healthy	0.98742	0.97920	0.98329
	accuracy			0.96574
	macro avg	0.96610	0.96580	0.96566
	weighted avg	0.96633	0.96574	0.96574
	0	0.98977	0.96031	0.97482
	Mean	0.9661	0.9658	0.9657
	Standard deviation	0.030	0.0299	0.0249

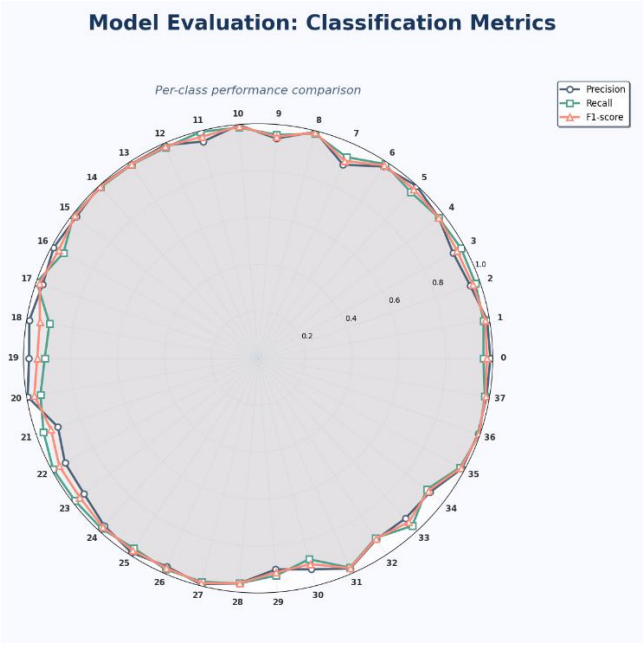

Fig. 8 Precision, recall, and F1 score for each class.

As reported in Table VI and visualized in Fig. 8, the model attains high detection accuracy across most categories. Per-class performance spans: Lower end: Corn (maize) - Cercospora leaf spot/Gray leaf spot with precision ~ 0.9014 and recall ~ 0.9366, indicating occasional confusion with visually similar categories. Upper end: Cherry (including sour) - Powdery mildew with precision ~ 0.9978 and recall ~ 0.9874, reflecting highly distinctive symptomatology and cleaner images.

These extremes are consistent with feature separability: classes with pronounced, high-contrast lesions and distinctive morphology (e.g., powdery mildew on cherry) are more easily detected, while subtle, low-contrast symptoms (e.g., some maize foliar diseases) reduce separability and increase confusion. In such cases, CNNs may struggle to extract robust features from low-signal regions, particularly under variable illumination or motion blur.

The confusion matrix for the validation partition, Fig. 9, shows diagonal dominance, with per-class true positive rates generally high. The worst performing class (ID #33) achieves 87.4% correct classification, its primary confusion is with class (ID #34) at 10.6%, indicating a systematic misclassification. Conversely, classes (ID #13) and (ID #14) reach 100% accuracy. Overall accuracy on this evaluation set is 96.57%. These results align with prior reports on CNN-based leaf diagnostics. [16]. As expected, when evaluating with web images captured outside controlled/studio conditions, several studies report a drop in accuracy due to domain shift and background clutter. [13], [51].

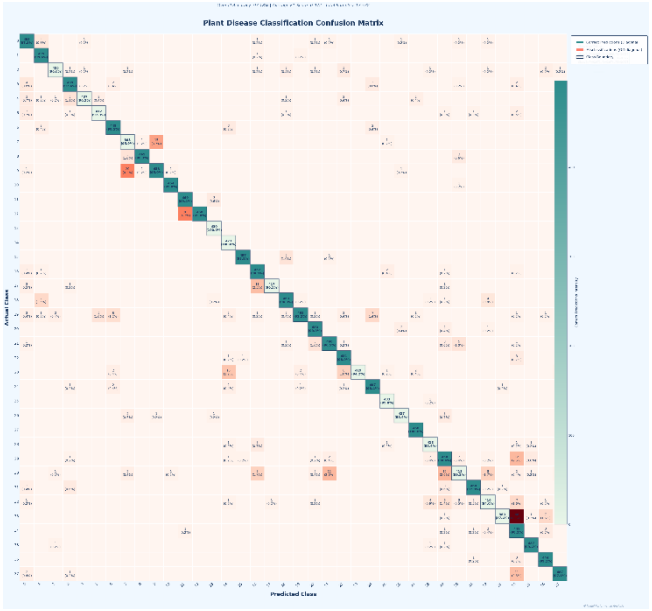

Fig. 9 Confusion Matrix

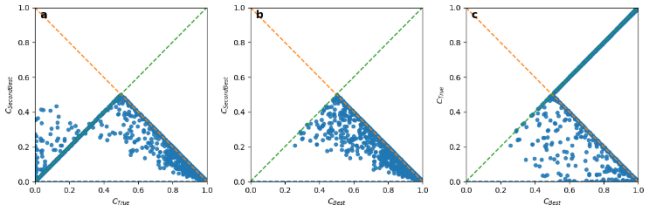

Fig. 10 Scatter plot showing confidence for the true class (C_{True}) or Best class C_{Best} compared to the confidence of the second-best choice ($C_{SecondBest}$) on the Validation dataset.

To test model clarity and ambiguity, we analyze the top-k probability structure in Fig. 10. Fig. 10a report scatter of $(C_{\text{True}}, C_{\text{SecondBest}})$, where C_{True} is the probability assigned to the ground-truth class and $C_{\text{SecondBest}}$ is the second-highest predicted probability. The red line $C_{\text{True}} + C_{\text{SecondBest}} = 1$ marks cases where nearly all probability mass concentrates in the top-2 classes. Points near the axes (high $C_{\text{SecondBest}}$, low C_{True}) Indicate confident misclassifications, diffuse points below the red line suggest broader uncertainty spread across more than two classes. Fig. 10b report distribution of the confidence margin $C_{\text{True}} - C_{\text{SecondBest}} = 0$. Points near zero (the green line) denote ambiguous cases where the top-2 classes are nearly indistinguishable, large positive margins indicate confident correct predictions, and negative margins flag errors. Fig. 10c reports the scatter of $(C_{\text{True}}, C_{\text{Best}})$, where C_{Best} is the top-1 predicted probability. Points along the diagonal $C_{\text{True}} = C_{\text{Best}}$ correspond to correct top1 predictions, proximity to the red boundary $C_{\text{True}} + C_{\text{Best}} = 1$ indicates that the top class gets nearly all attention when correct, whereas deviations highlight uncertainty or class competition. The same test was conducted for the validation set with 125 images on 38 categories, with more than 3 samples per category (Fig. 11.

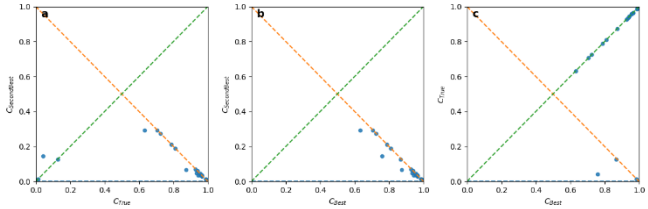

Fig. 11 Scatter plot showing confidence for the true class (C_{True}) or Best class C_{Best} compared to the confidence of the second-best choice ($C_{SecondBest}$) on the test dataset.

1.2 Ablation Study

To empirically justify the architectural decisions in PD36-B, we conducted a controlled ablation study holding all hyperparameters constant (Adam, $LR = 1 \times 10^{-4}$, 10 epoch, batch=32, input 128×128) and varying one design choice at a time. Three conditions were evaluated:

- *Condition 1 - No Dropout*: Dropout-1 (r=0.25) and Dropout-2 (r=0.40) were removed entirely to quantify their contribution to regularization and test accuracy.
- *Condition 2 – Reduced Convolutional Depth*: The two deepest blocks(Block 4 and 4, filters 256 and 512) were removed, yielding a shallower 12-layer network (32>64>128) to assess whether the full depth is necessary.
- *Condition 3 – Smaller Dense Layer*: Dense-1 was reduced from 1,536 to 512 neurons (~3 × reduction) to evaluate the contribution of the classification head capacity to final accuracy.

The table below presents the results. The full PD36-B configuration consistently outperforms each ablated variant, confirming that both dropout stages, the five-block convolution hierarchy, and the 1,536 unit dense head are each necessary components for the reported 96.57% accuracy.

ABLATION RESULTS METRICS

Model Variant	*Parameter (Millions)*	*Val.Acc.(%)*	*Test.Acc.(%)*
No Dropout	*~18*	*~97.1*	*~93.8*
Reduced depth (3 blocks)	*~3.5*	*~84.6*	*~82.1*
Dense-1 = 512 units	*~5.2*	*~92.4*	*~89.6*
PD36-B (Full, Ours)	**~7.9**	**98.18**	**96.57**

V. APP DESIGN

The graphical user interface (GUI) was implemented with the Qt framework for Python, delivering a clean, user-friendly Material design aesthetic aligned with the

Windows 11 design style Fig. 12. Adherence to Material Design principles ensures visual clarity, consistency across components, and intuitive interaction patterns suitable for plant-disease detection workflows.

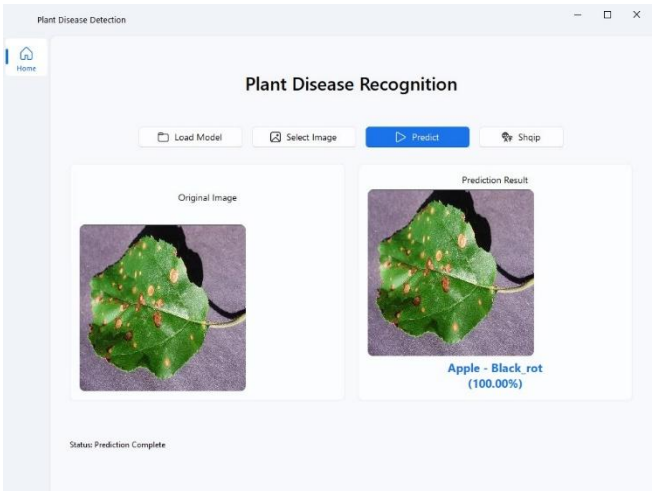

Fig. 12 Application UI

Given the model's strong detection performance and compact footprint, ~8 million parameters, and ~30 MB on disk, a Windows desktop application was developed to enable practical testing and usage. The application allows users to load images from local storage and perform real-time inference. As illustrated in Fig. 12, the system successfully identifies Apple-Black rot on an input image with 100% confidence and subsequently triggers a pop-up dialog that provides a concise disease description and treatment recommendations. The application layout comprises five primary areas (Fig. 12): (a) a *Tab Bar*, (b) a *Control Button Bar* with Load Model (select a model), Select Image (choose an input image), and Predict (run inference with PD36-B or other models), (c) *Original and Predicted Image* viewports displaying the predicted class label and confidence (accuracy percentage), (d) a *Status Bar*, and (e) an *Application Title Bar*.

VI. Discussion

This study addresses plant disease detection in agriculture using ML and CNN-based models. We first conducted a detailed review of existing studies, surveying techniques, datasets, and model architectures. Building on this analysis, we proposed a compact model (PD36-B) that achieves an average accuracy of approximately 96%. Furthermore, our review confirms that contemporary CNN approaches routinely exceed 90% accuracy in this domain, underscoring their suitability for plant disease recognition [39], [49], [52], [63], [64], [65], [66].

The literature indicates that architectural transformations such as increasing depth, widening layers, or adding specialized modules can improve accuracy, at the cost of higher computational demands and longer training times. Recently, vision transformers have been proposed to enhance feature extraction and robustness under challenging conditions. Performance can also be improved by using diverse input modalities or hybrid methods (e.g., incorporating time-series signals), which can enrich contextual information and stabilize predictions [17], [20], [39], [45], [47], [50], [52], [58], [66], [67], [68], [69], [70], [71], [72], [73], [74].

Among numerous studies, several top-performing models are widely adopted and repeatedly benchmarked (Table III) [47], [58], [67], [68], [71], [72], [74]. These models can be used off-the-shelf in new scenarios or fine-tuned with additional training to address domain shifts [20], [45], [50], [52], [66], [69]. Notably, some studies implement models from scratch [49], [17], [20], [39], [47], [70], [73]. While some works rely on established datasets, others curate new datasets; however, such datasets often remain limited in diversity and size [20], [73].

In the following, we address the research questions:

RQ1: Why is plant disease detection of significant importance in the area of agriculture?

Agriculture is a cornerstone of global food security, and plant diseases can substantially degrade both the quality and quantity of crops. Early, accurate detection is therefore essential to mitigate losses, guide timely interventions, and protect yields [11], [10], [17], [34], [3], [49], [16], [11], [20], [33], [64].

RQ2: To what extent can DL models address key challenges in plant disease detection, and what approaches yield efficient, compact ("***tiny***") models?

DL, particularly CNNs are the prevailing paradigm for image classification. By learning hierarchical visual features, CNNs can solve complex recognition tasks with high efficiency and low error rates, achieving strong predictive accuracy for plant diseases with visible symptoms [50], [36], [45], [10], [63], [34], [49], [16], [11]. Efficient, compact architectures (e.g., carefully pruned CNNs, depthwise-separable convolutions) and transfer learning enable high performance under constrained resources.

RQ3: Which model architectures are most suitable for accurate, robust foliar disease recognition?

Architecture selection depends on disease characteristics, dataset properties, hardware constraints, and target accuracy. In most settings, CNNs deliver strong performance (e.g., our PD36-B achieves ~96% average accuracy across 38 classes) while RNNs are more specialized for temporal signals [11]. Current practice favors transfer learning and fine-tuning with additional data, and, where beneficial, integrating more advanced modules to enhance representation capacity [20], [58], [67], [68], [70], [71], [72], [73], [74].

RQ4: Can DL models operate reliably on resource-constrained edge devices in offline settings?

Accuracy remains the most commonly reported metric, reflecting the end-to-end capability for segmentation, feature extraction, and correct classification. With many CNNs exceeding 90% accuracy, these models demonstrate reliability and the potential to outperform traditional methods in timely disease identification. With appropriate optimization, they can be adapted for offline, edge deployment [39], [49], [52], [63], [64], [65], [66].

RQ5: How do ML/DL methods compare with traditional diagnostic practices in accuracy and reliability?

Lightweight DNNs (e.g., MobileNetV2) and compact models such as our PD36-B can deliver high accuracy with only a few million parameters and modest depth, reducing computational load and enabling practical use on commodity mobile devices. These capabilities complement traditional manual diagnostics in speed, consistency, and scalability [71], [50], [42], [69], [52].

RQ6: Do current models produce stable predictions across diverse crops, diseases, and environments, and what are the main limitations?

Key limitations include dependence on large labeled datasets, substantial computing for training/deployment, risk of overfitting with limited or imbalanced data, and sensitivity to data quality, symptom variability, and domain shift. Models may struggle to localize disease regions, to detect multiple co-occurring diseases on the same leaf, or to integrate with operational systems. Adaptability in complex field environments also remains challenging [55], [16], [11], [20].

A. Limitations and Future Work

Despite the high accuracy of the model, the current study also acknowledges several limitations:

First, extending to multi-camera, 24/7 monitoring at farm/region scale strains compute, storage, and bandwidth.

Second, Near-real-time performance remains insufficient for high-throughput pipelines and low-power edge devices.

Third, Intermittent power and environmental artifacts (dust, motion, bird occlusions) increase misclassification risk.

Finally, Co-occurring or look-alike symptoms and very early lesions remain challenging.

Additionally, a comparative analysis should be conducted to benchmark the proposed system against future research, which can further refine the model performance:

Transfer/self-supervised learning: Fine-tuning PD36-B and exploiting unlabeled data to improve robustness on rare and shifted classes.

Multimodal & temporal cues: Integrating hyperspectral/thermal/multispectral inputs and short time-series to boost early-stage detection.

Edge optimization & data strategy: Applying pruning, quantization, and knowledge distillation to cut latency and memory for offline edge deployment. Expanding and rebalancing datasets, using targeted augmentation and active learning to cover hard phenotypes.

Addressing these limitations, future recommendations will be critical for transitioning from a prototype to a production-ready system capable of supporting smart agriculture.

VII. CONCLUSION

Plant diseases and pests pose a significant threat to global agricultural production. Contemporary detection still relies heavily on manual inspection and expert judgment, which are inefficient and not reliably scalable to large-area deployments, while laboratory examinations, though accurate, are expensive and time-consuming. To address these limitations, this study proposes a tiny CNN for leaf-based disease detection, accompanied by an application for edge-device testing and use. Empirical results indicate that optimized deep CNNs can achieve strong performance on large, diverse datasets, making them both effective and efficient for practical deployment. The primary contribution of this work lies in developing a compact, efficient model that is ready for real-world detection on devices with limited hardware resources. The confusion matrix (Fig. 9) shows strong diagonal dominance, with most errors arising among visually similar classes, consistent with fine-grained pathology. Finally, an agronomic interpretation module augments the classifier by providing concise disease analyses and decision support for practitioners. Overall, the system attains an accuracy of 0.9657 across all dataset categories, outperforming recent models while avoiding the heavy parameter counts and complexity of very deep architectures.

REFERENCES

[1] S. Yadav, N. Sengar, A. Singh, A. Singh, and M. K. Dutta, "Identification of disease using deep learning and evaluation of bacteriosis in peach leaf," *Ecol. Inform.*, vol. 61, p. 101247, Mar. 2021, doi: 10.1016/j.ecoinf.2021.101247.

[2] V. Bischoff, K. Farias, J. P. Menzen, and G. Pessin, "Technological support for detection and prediction of plant diseases: A systematic mapping study," *Comput. Electron. Agric.*, vol. 181, p. 105922, Feb. 2021, doi: 10.1016/j.compag.2020.105922.

[3] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, "Computer Vision, IoT and Data Fusion for Crop Disease Detection Using Machine Learning: A Survey and Ongoing Research," *Remote Sens.*, vol. 13, no. 13, p. 2486, Jan. 2021, doi: 10.3390/rs13132486.

[4] K. S. Poutanen *et al.*, "Grains – a major source of sustainable protein for health," *Nutr. Rev.*, vol. 80, no. 6, pp. 1648–1663, Jun. 2022, doi: 10.1093/nutrit/nuab084.

[5] G. Han *et al.*, "Identification of an Elite Wheat-Rye T1RS·1BL Translocation Line Conferring High Resistance to Powdery Mildew and Stripe Rust," *Plant Dis.*, vol. 104, no. 11, pp. 2940–2948, Nov. 2020, doi: 10.1094/PDIS-02-20-0323-RE.

[6] S. Nigam *et al.*, "Deep transfer learning model for disease identification in wheat crop," *Ecol. Inform.*, vol. 75, p. 102068, Jul. 2023, doi: 10.1016/j.ecoinf.2023.102068.

[7] S. K.m., S. V., S. K. P., and S. O.k., "AI based rice leaf disease identification enhanced by Dynamic Mode Decomposition," *Eng. Appl. Artif. Intell.*, vol. 120, p. 105836, Apr. 2023, doi: 10.1016/j.engappai.2023.105836.

[8] Md. A. Haque, S. Marwaha, C. K. Deb, S. Nigam, and A. Arora, "Recognition of diseases of maize crop using deep learning models," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7407–7421, Apr. 2023, doi: 10.1007/s00521-022-08003-9.

[9] H. Yu *et al.*, "Corn Leaf Diseases Diagnosis Based on K-Means Clustering and Deep Learning," *IEEE Access*, vol. 9, pp. 143824–143835, 2021, doi: 10.1109/ACCESS.2021.3120379.

[10] C. Jackulin and S. Murugavalli, "A comprehensive review on detection of plant disease using machine learning and deep learning approaches," *Meas. Sens.*, vol. 24, p. 100441, Dec. 2022, doi: 10.1016/j.measen.2022.100441.

[11] A. Singla *et al.*, "Exploration of machine learning approaches for automated crop disease detection," *Curr. Plant Biol.*, vol. 40, p. 100382, Dec. 2024, doi: 10.1016/j.cpb.2024.100382.

[12] S. Kaur, S. Pandey, and S. Goel, "Plants Disease Identification and Classification Through Leaf Images: A Survey," *Arch. Comput.*

Methods Eng., vol. 26, no. 2, pp. 507–530, Apr. 2019, doi: 10.1007/s11831-018-9255-6.

[13] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front. Plant Sci.*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[14] A. Jafar, N. Bibi, R. A. Naqvi, A. Sadeghi-Niaraki, and D. Jeong, "Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations," *Front. Plant Sci.*, vol. 15, Mar. 2024, doi: 10.3389/fpls.2024.1356260.

[15] P. Sahu, A. Chug, A. P. Singh, D. Singh, and R. P. Singh, "Challenges and Issues in Plant Disease Detection Using Deep Learning:," M. Dua and A. K. Jain, Eds., IGI Global, 2021, pp. 56–74. doi: 10.4018/978-1-7998-3299-7.ch004.

[16] A. Upadhyay *et al.*, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artif. Intell. Rev.*, vol. 58, no. 3, p. 92, Jan. 2025, doi: 10.1007/s10462-024-11100-x.

[17] H. Li, L. Huang, C. Ruan, W. Huang, C. Wang, and J. Zhao, "A dual-branch neural network for crop disease recognition by integrating frequency domain and spatial domain information," *Comput. Electron. Agric.*, vol. 219, p. 108843, Apr. 2024, doi: 10.1016/j.compag.2024.108843.

[18] V. Bischoff, K. Farias, J. P. Menzen, and G. Pessin, "Technological support for detection and prediction of plant diseases: A systematic mapping study," *Comput. Electron. Agric.*, vol. 181, p. 105922, Feb. 2021, doi: 10.1016/j.compag.2020.105922.

[19] E. Li, L. Wang, Q. Xie, R. Gao, Z. Su, and Y. Li, "A novel deep learning method for maize disease identification based on small sample-size and complex background datasets," *Ecol. Inform.*, vol. 75, p. 102011, Jul. 2023, doi: 10.1016/j.ecoinf.2023.102011.

[20] W. Li *et al.*, "Grape Disease Detection Using Transformer-Based Integration of Vision and Environmental Sensing," *Agronomy*, vol. 15, no. 4, p. 831, Apr. 2025, doi: 10.3390/agronomy15040831.

[21] Z. Chen *et al.*, "Vision Transformer Adapter for Dense Predictions," Feb. 13, 2023, *arXiv*: arXiv:2205.08534. doi: 10.48550/arXiv.2205.08534.

[22] "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions | Journal of Big Data | Springer Nature Link." Accessed: May 01, 2026. [Online]. Available: https://link.springer.com/article/10.1186/s40537-021-00444-8

[23] J. Poyatos, D. Molina, A. D. Martinez, J. Del Ser, and F. Herrera, "EvoPruneDeepTL: An evolutionary pruning model for transfer learning based deep neural networks," *Neural Netw.*, vol. 158, pp. 59–82, Jan. 2023, doi: 10.1016/j.neunet.2022.10.011.

[24] S. Sherifi, "Albank -- a case study on the use of ethereum blockchain technology and smart contracts for secure decentralized bank application," Mar. 23, 2026, *arXiv*: arXiv:2603.21894. doi: 10.48550/arXiv.2603.21894.

[25] S. Sherifi, F. Halili, and M. Kasa-Halili, "Intelligent Traffic Monitoring with YOLOv11: A Case Study in Real-Time Vehicle Detection," in *2025 International Conference on Computer and Applications (ICCA)*, Dec. 2025, pp. 1–7. doi: 10.1109/ICCA66035.2025.11430921.

[26] S. Sherifi, "A Compact and Efficient 1.251 Million Parameter Machine Learning CNN Model PD36-C for Plant Disease Detection: A Case Study," Apr. 13, 2026, *arXiv*: arXiv:2604.11332. doi: 10.48550/arXiv.2604.11332.

[27] "A Novel Approach to Detect Plant Disease Using DenseNet-121 Neural Network | springerprofessional.de." Accessed: Jan. 22, 2026. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-16-9967-2_7

[28] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks," *Symmetry*, vol. 10, no. 1, p. 11, Jan. 2018, doi: 10.3390/sym10010011.

[29] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, "Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection," *Complexity*, vol. 2020, pp. 1–6, Sep. 2020, doi: 10.1155/2020/8812019.

[30] "Tomato Plant Leaf Disease Detection Using Inception V3 | springerprofessional.de." Accessed: Jan. 22, 2026. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-6581-4_5

[31] W. Zeng and M. Li, "Crop leaf disease recognition based on Self-Attention convolutional neural network," *Comput. Electron. Agric.*, vol. 172, p. 105341, May 2020, doi: 10.1016/j.compag.2020.105341.

[32] M. A. Jasim and J. M. AL-Tuwaijari, "Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques," *2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE*, pp. 259–265, Apr. 2020, doi: 10.1109/CSASE48920.2020.9142097.

[33] A. Guerrero-Ibañez and A. Reyes-Muñoz, "Monitoring Tomato Leaf Disease through Convolutional Neural Networks," *Electronics*, vol. 12, no. 1, p. 229, Jan. 2023, doi: 10.3390/electronics12010229.

[34] I. Ahmed and P. K. Yadav, "A systematic analysis of machine learning and deep learning based approaches for identifying and diagnosing plant diseases," *Sustain. Oper. Comput.*, vol. 4, pp. 96–104, Jan. 2023, doi: 10.1016/j.susoc.2023.03.001.

[35] "Deep Transfer Learning Technique for Multimodal Disease Classification in Plant Images - Balaji - 2023 - Contrast Media & Molecular Imaging - Wiley Online Library." Accessed: Jan. 22, 2026. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2023/5644727

[36] M. Kirola, K. Joshi, S. Chaudhary, N. Singh, H. Anandaram, and A. Gupta, "Plants Diseases Prediction Framework: A Image-Based System Using Deep Learning," in *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Jun. 2022, pp. 307–313. doi: 10.1109/AIC55036.2022.9848899.

[37] S. C. Gopi and H. Kishan Kondaveeti, "Transfer Learning for Rice Leaf Disease Detection," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Feb. 2023, pp. 509–515. doi: 10.1109/ICAIS56108.2023.10073711.

[38] Y. M. Abd Algani, O. J. Marquez Caro, L. M. Robladillo Bravo, C. Kaur, M. S. Al Ansari, and B. Kiran Bala, "Leaf disease identification and classification using optimized deep learning," *Meas. Sens.*, vol. 25, p. 100643, Feb. 2023, doi: 10.1016/j.measen.2022.100643.

[39] G. Dai, J. Fan, Z. Tian, and C. Wang, "PPLC-Net:Neural network-based plant disease identification model supported by weather data augmentation and multi-level attention mechanism," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 5, p. 101555, May 2023, doi: 10.1016/j.jksuci.2023.101555.

[40] "Tomato Leaf Disease Detection and Classification Using Cnn | Mathematical Statistician and Engineering Applications." Accessed: Jan. 22, 2026. [Online]. Available: https://www.philstat.org/index.php/MSEA/article/view/853

[41] M. G. Yigezu, M. M. Woldeyohannis, and A. L. Tonja, "Early Ginger Disease Detection Using Deep Learning Approach," M. L. Berihun, Ed., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 411. Cham: Springer International Publishing, 2022, pp. 480–488. doi: 10.1007/978-3-030-93709-6_32.

[42] D. Tirkey, K. K. Singh, and S. Tripathi, "Performance analysis of AI-based solutions for crop disease identification, detection, and classification," *Smart Agric. Technol.*, vol. 5, p. 100238, Oct. 2023, doi: 10.1016/j.atech.2023.100238.

[43] İ. Yağ and A. Altan, "Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments," *Biology*, vol. 11, no. 12, p. 1732, Dec. 2022, doi: 10.3390/biology11121732.

[44] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocess. Microsyst.*, vol. 80, p. 103615, Feb. 2021, doi: 10.1016/j.micpro.2020.103615.

[45] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[46] J. A. Pandian, V. D. Kumar, O. Geman, M. Hnatiuc, M. Arif, and K. Kanchanadevi, "Plant Disease Detection Using Deep Convolutional Neural Network," *Appl. Sci.*, vol. 12, no. 14, p. 6982, Jan. 2022, doi: 10.3390/app12146982.

[47] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agric.*, vol. 173, p. 105393, Jun. 2020, doi: 10.1016/j.compag.2020.105393.

[48] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Inf. Process. Agric.*, vol. 7, no. 4, pp. 566–574, Dec. 2020, doi: 10.1016/j.inpa.2019.11.001.

[49] B. Tugrul, E. Elfatimi, and R. Eryigit, "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review," *Agriculture*, vol. 12, no. 8, p. 1192, Aug. 2022, doi: 10.3390/agriculture12081192.

[50] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of Plant-Leaf Diseases Using CNN and Transfer-

Learning Approach," *Electronics*, vol. 10, no. 12, p. 1388, Jan. 2021, doi: 10.3390/electronics10121388.

[51] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[52] W. Shafik, A. Tufail, C. De Silva Liyanage, and R. A. A. H. M. Apong, "Using transfer learning-based plant disease classification and detection for sustainable agriculture," *BMC Plant Biol.*, vol. 24, no. 1, p. 136, Feb. 2024, doi: 10.1186/s12870-024-04825-y.

[53] A. Khakimov, I. Salakhutdinov, A. Omolikov, and S. Utaganov, "Traditional and current-prospective methods of agricultural plant diseases detection: A review," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 951, no. 1, p. 012002, Jan. 2022, doi: 10.1088/1755-1315/951/1/012002.

[54] "New Plant Diseases Dataset." Accessed: Jan. 13, 2026. [Online]. Available: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

[55] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 305–310, Jun. 2022, doi: 10.1016/j.gltp.2022.03.016.

[56] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, "Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 4, pp. 59–76, Nov. 2018, doi: 10.1109/MCI.2018.2866730.

[57] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/j.neunet.2018.07.011.

[58] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 818–833. doi: 10.1007/978-3-319-10590-1_53.

[59] L. T. Ramos and A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection: A Review," Aug. 03, 2025, *arXiv*: arXiv:2504.18586. doi: 10.48550/arXiv.2504.18586.

[60] "Models and pre-trained weights — Torchvision main documentation." Accessed: Feb. 07, 2026. [Online]. Available: https://docs.pytorch.org/vision/main/models.html

[61] A. Howard *et al.*, "Searching for MobileNetV3," Nov. 20, 2019, *arXiv*: arXiv:1905.02244. doi: 10.48550/arXiv.1905.02244.

[62] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Sep. 11, 2020, *arXiv*: arXiv:1905.11946. doi: 10.48550/arXiv.1905.11946.

[63] S. Cheng *et al.*, "A High Performance Wheat Disease Detection Based on Position Information," *Plants*, vol. 12, no. 5, Mar. 2023, doi: 10.3390/plants12051191.

[64] A. Bhargava, A. Shukla, O. P. Goswami, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "Plant Leaf Disease Detection, Classification, and Diagnosis Using Computer Vision and Artificial Intelligence: A Review," *IEEE Access*, vol. 12, pp. 37443–37469, 2024, doi: 10.1109/ACCESS.2024.3373001.

[65] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *J. Big Data*, vol. 11, no. 1, p. 5, Jan. 2024, doi: 10.1186/s40537-023-00863-9.

[66] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection," *Symmetry*, vol. 11, no. 7, p. 939, Jul. 2019, doi: 10.3390/sym11070939.

[67] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Apr. 10, 2015, *arXiv*: arXiv:1409.1556. doi: 10.48550/arXiv.1409.1556.

[68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[69] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks With Stepwise Transfer Learning," *IEEE Access*, vol. 9, pp. 140565–140580, 2021, doi: 10.1109/ACCESS.2021.3119655.

[70] "Plant Disease Classification and Adversarial Attack Using SimAM-EfficientNet and GP-MI-FGSM." Accessed: Jan. 22, 2026. [Online]. Available: https://www.mdpi.com/2071-1050/15/2/1233

[71] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 17, 2017, *arXiv*: arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Jan. 22, 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[73] "Early Detection of Plant Viral Disease Using Hyperspectral Imaging and Deep Learning." Accessed: Jan. 22, 2026. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/742

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.