

DEEP LEARNING-BASED MULTICLASS CLASSIFICATION OF MEDICINAL PLANT LEAF DISEASES USING EFFICIENTNET-B3, CONVNEXT-TINY, SWIN TRANSFORMER AND A HYBRID ENSEMBLE ON THE CIMD DATASET

K. NAGA DIVYA¹, AVANI ALLA², PAPPULA SARALA³, S NAGA SINDHU⁴,
SURESH KUMAR GUDISE⁵, BALAMURALIKRISHNA THATI⁶

¹Assistant Professor, Department of Computer Science and Engineering, PVP Siddhartha Institute of Technology, Vijayawada, A.P, India.

² Professor, Department of Computer Science and Engineering, AnuBose Institute of Technology for Women's, palvanha, Telangana, India.

³ Sr Assistant Professor, Department of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, AP, India.

⁴Assistant Professor, Department of Computer Science and Engineering, Dhanekula Institute of Engineering & Technology, Vijayawada, A.P, India.

⁵ Associate Professor, Department of Computer Science & Engineering, Dr.RVR NRI Institute of Technology Deemed to be University, Pothavarappadu, Agiripalli, A.P, India.

⁶Professor, Department of Artificial Intelligence & Machine Learning, DVR & Dr. HS MIC College of Technology, Kanchikacherla, NTR District, A.P, India.

Email: balu.thati9@gmail.com

ABSTRACT

Plant diseases threaten the pharmacological quality of medicinal herbs cultivated across Central India and remain difficult to diagnose accurately without laboratory analysis. The Central India Medicinal Plant Dataset (CIMPD), released by Tomar et al. in Data in Brief (Elsevier, 2025), contains 9,130 high-resolution leaf images spanning 23 Indian medicinal plant species under both healthy and diseased conditions. No comprehensive multi-architecture deep learning benchmark has been published on this dataset beyond the ResNet18 baseline in the original paper. This work presents the first end-to-end transfer learning, ensemble, and GradCAM++ interpretability pipeline on CIMPD. Six architectures are benchmarked: a Simple CNN trained from scratch, ResNet50, EfficientNet-B3, ConvNeXt-Tiny, Swin Transformer-Tiny, and a soft-voting Hybrid Model that combines the three strongest backbones using validation-accuracy-proportional weights. Training employs MixUp augmentation ($\alpha=0.4$), CoarseDropout regularisation, label smoothing ($\epsilon=0.1$), AdamW optimisation, and cosine annealing with warm restarts. The proposed Hybrid Model achieves 90% test accuracy with weighted F1=0.885, Cohen's κ and MCC substantially above chance, and AUC=1.00 on all four reported class categories, outperforming the Simple CNN baseline by 15 percentage points and the prior ResNet18 result by approximately 7 points. GradCAM++ saliency maps confirm that model attention concentrates on biologically meaningful leaf regions. All experiments are reproducible through a single self-contained Jupyter notebook.

Keywords: *Medicinal plant disease classification, CIMPD, EfficientNet-B3, ConvNeXt, Swin Transformer, transfer learning, ensemble learning, GradCAM++, MixUp augmentation, deep learning, precision agriculture.*

1. INTRODUCTION

There are over 8,000 types of medicinal plants in the world and over 8,000 of them are cultivated smallholder farming systems in Southern and Central India for the AYUR, UNANI and SIDDHA systems of medicine. The global market for herbal

medicine generated revenue of over 131 billion dollars in 2020, and is expected to grow to 347 billion dollars by 2030. The market for herbal medicine is an important source of revenue for countries that grow high quality medicinal plants. The crops of medicinal plants are more vulnerable to disease from fungi and can be infected by

bacteria and viruses which make the crops less potent and less appealing to consumers. This affects not only poor farmers that grow the crops, but the large pharmaceutical companies that buy the crops [31-38].

The computer vision literature has largely focused on staple food crops like rice, wheat, and tomato, whereas it has largely ignored medicinal crops. While competitive advantages in phenotyping and large image databases have benefited staple crops, computer vision has yet to prioritize medicinal crops. It's even more difficult to classify certain crops in the CIMPD dataset with overlapping disease symptoms. For example, powdery mildew on Tulsi and early-stage mosaic on Giloy are visually similar. Automated classification in single-species benchmarks is simpler. A well-designed automated detection system would let farmers take a picture of a diseased crop with a smartphone and implement a diagnosis. This would prevent the disease from spreading, and reduce the economic impacts of the disease and use of pesticides.

The advent of deep convolutional neural networks (CNNs) transformed plant disease detection. Mohanty et al. [1] demonstrated in 2016 that GoogLeNet and AlexNet fine-tuned on the PlantVillage dataset could classify 26 diseases across 14 crops with accuracy exceeding 99% under controlled imaging conditions. Subsequent research broadened the scope to field images [5], low-resource devices [5], and explainable predictions [17]. More recently, Vision Transformers (ViT) [8], which treat image patches as sequential tokens and apply multi-head self-attention, have challenged CNN dominance on large-scale benchmarks. Swin Transformer [9] introduced a hierarchical, shifted-window self-attention mechanism that achieves linear computational complexity with image size, while ConvNeXt [10] demonstrated that standard convolutional architectures retrained with transformer-inspired design choices can match transformer performance on ImageNet.

The Central India Medicinal Plant Dataset (CIMPD) was published by Tomar et al. [7] in 2025 alongside a ResNet18 baseline reporting approximately 83% overall accuracy. No subsequent published work has conducted a systematic multi-architecture comparison, applied soft-voting ensemble methods, or provided GradCAM++-based interpretability on CIMPD. This gap motivates the present study.

To summarise, the literature shows: (1) transfer learning from ImageNet substantially improves

plant disease classification accuracy; (2) transformer-based architectures capture long-range spatial dependencies that CNNs with limited receptive fields may miss; (3) ensemble methods consistently reduce variance across model predictions; and (4) interpretability tools are necessary for clinical and agricultural trustworthiness. No prior work has combined all four of these insights in a single pipeline applied to CIMPD. This paper addresses that gap by making the following five contributions:

1. The first multi-architecture transfer learning benchmark on CIMPD, covering Simple CNN, ResNet50, EfficientNet-B3, ConvNeXt-Tiny, and Swin Transformer-Tiny, with rigorous hyperparameter settings and reproducible code.
2. A soft-voting Hybrid Model that combines EfficientNet-B3, ConvNeXt-Tiny, and Swin-T using validation-accuracy proportional weights and achieves 90% test accuracy — the highest reported on CIMPD to date.
3. Formal pseudocode specification of both the training pipeline (Algorithm 1) and the ensemble weighting procedure (Algorithm 2), enabling exact replication.
4. System-level and architecture-level diagrams illustrating the complete pipeline from raw image ingestion to disease prediction and saliency visualisation.
5. GradCAM++ interpretability analysis confirming that the best model attends to biologically meaningful leaf regions, establishing a foundation for trust in field deployment.

The remainder of this paper is organised as follows. Section II provides a comprehensive literature review with summary table. Section III describes the CIMPD dataset and preprocessing pipeline. Section IV presents the proposed methodology including the two formalised algorithms and system diagrams. Section V reports experimental results across all six models. Section VI discusses findings, limitations, and future directions. Section VII concludes the paper.

2. RELATED WORK

2.1 Early Deep Learning Approaches for Plant Disease Detection

The use of deep learning for plant disease detection was arguably popularised by the work of Mohanty et al. [1], who trained AlexNet and GoogLeNet on the 54,306-image PlantVillage dataset and reported 99.35% accuracy under held-out test conditions. This result established CNN-

based classification as a technically viable approach for automated phytopathology. Ferentinos [2] extended this line with a custom deep CNN architecture achieving 99.53% on 87,848 PlantVillage images. Brahimi et al. [3] specifically targeted tomato leaf diseases with AlexNet and OxfordNet, reporting 99.18% accuracy on a nine-class problem and additionally studying the visual decision boundaries via class activation mapping. Tm et al. For rice leaf disease identification, [4] used VGG16 and InceptionV3 to show that transfer learning from ImageNet weights for agricultural domain tasks can obtain >95% overall accuracy on modest sized data-specific training sets.

These initial studies indicate that Plant Village high accuracy can be misleading for real-world applications. Hughes and Salathé [13] described this gap, and most importantly, demonstrated that uncontrolled outdoor illumination can reduce performance by as much as 30 percentage points. As a result, there has been significant interest in research studies on CIMPD and other ecologically valid dataset collections, and advancements in the fields of test-time augmentations and domain adaptation. Transfer Learning and Architecture Advances

He et al. [12] proposed deep residual neural networks (ResNet), with identity shortcut connections which addressed vanishing gradient degradation to allow training networks with depth larger than 100 layers. Transfer learning backbones based on ResNet50 became prevalent in agriculture image analysis because of its relatively high depth, lower parameter count (~25 million), and quality of representation learned from ImageNet. Kaya et al. [6] performed an ablation study on ResNet, DenseNet, and VGG architectures on PlantVillage using transfer learning. They showed that all pre-trained networks significantly outperformed training a network from scratch and that the additional benefit saturates past ResNet50 depth.

Tan and Le [11] proposed EfficientNet that uniformly scales network width, depth, and input resolution with a set of fixed scaling coefficients. The coefficients are discovered automatically by a neural architecture search. We use EfficientNet-B3 here, which takes 300×300 inputs and can reach 84.4% ImageNet top-1 accuracy with only 12M parameters. As such, EfficientNet can be fine-tuned for mid-scale datasets such as medical and agricultural images. Liu et al. [10] proposed ConvNeXt, a pure CNN re-engineered with transformer-inspired choices: depthwise 7×7 convolutions replacing 3×3 stacked filters,

LayerNorm replacing BatchNorm, GELU activations, and an inverted bottleneck structure. ConvNeXt-Tiny matches or exceeds Swin-T on ImageNet with comparable compute cost. Ramcharan et al. [5] demonstrated that even lighter architectures like MobileNet variants adapted from InceptionV3 could achieve 93% accuracy for cassava disease detection when deployed on Android smartphones, establishing the feasibility of edge deployment from high-accuracy backbones via knowledge distillation.

2.2 Vision Transformers in Agricultural Image Analysis

Dosovitskiy et al. [8] introduced the Vision Transformer (ViT), which partitions an image into non-overlapping patches, linearly embeds them, and processes them with a standard transformer encoder. When pre-trained on large corpora (ImageNet-21K), ViT-L/16 achieves 88.55% top-1 accuracy on ImageNet, outperforming prior CNN state of the art. However, ViT requires large pre-training datasets and lacks the locality inductive bias of CNNs, which can limit performance on small domain-specific datasets without extensive regularisation. Liu et al. [9] addressed this with Swin Transformer, which introduces hierarchical feature maps (analogous to CNN feature pyramids) and limits self-attention computation to non-overlapping local windows, with alternating window shifts between layers to enable cross-window information exchange. Swin-T achieves 87.3% ImageNet top-1 with 28M parameters and is our strongest individual model on CIMPD at 89%. Chen et al. [13] specifically applied ViT fine-tuning to rice panicle disease detection and found that the global context captured by self-attention improved detection of diffuse discolouration patterns that span the entire leaf blade, a capability relevant to CIMPD's mosaic virus categories.

2.3 Ensemble Methods and Regularisation

Ensemble methods combine the predictions of multiple models to reduce variance and improve generalisation. Soft-voting ensembles, which average class probability distributions, are more effective than hard-voting (majority class) when constituent models are well-calibrated. EfficientNet-B3, ConvNeXt, and Swin-T have quite different architectural inductive biases - local convolution, hierarchical depth wise convolution, and global self-attention, respectively. Such diversity in inductive biases makes these three models natural choices to ensemble together, as

their errors should be less correlated than ensembling between models that share an architecture family. Zhang et al. [16] proposed MixUp, a data augmentation method that learns convex combinations of training examples and their labels. MixUp improves calibration, alleviates overconfidence and improves accuracy, especially on smaller and mid-size datasets. Szegedy et al. [19] proposed label smoothing, a regularisation method that discourages the model from putting all probability mass onto the ground truth label.

2.4 Medicinal Plant Recognition

Wäldchen and Mäder [15] examined three decades of plant identification literature and determined that species-level recognition attained human-equivalent accuracy only after the implementation of deep learning in approximately 2016. The LifeCLEF plant challenge series [20] set standard benchmarks for identifying plant species in detail. The best submissions used CNN ensembles and inputs from different sources (leaf, flower, bark). Rai et al. [14] utilised MobileNetV2 for Ayurvedic plant leaf identification, attaining 97% accuracy on a ten-species dataset, while observing significant performance deterioration when images were captured by varying cameras or under diverse lighting conditions. Tomar et al.'s CIMPD dataset [7] is the most complete publicly available dataset for classifying diseases in Indian medicinal plants. It includes 23 species grown in the field. The ResNet18 baseline of about 83% leaves a lot of room for improvement, which this paper directly talks about.

Table 1. Summary Of Key Related Works In Plant Disease Detection And Deep Architecture Research

Ref.	Author(s), Year	Dataset	Architecture	Task	Accuracy
[1]	Mohanty et al., 2016	PlantVillage (54,306 imgs)	AlexNet, GoogLeNet	26-class disease classification	99.35 %
[2]	Ferentinos, 2018	PlantVillage (87,848 imgs)	Custom deep CNN	Disease detection	99.53 %
[3]	Brahmi et al.,	Tomato leaf images	AlexNet, OxfordNet	9-class tomato disease	99.18 %

	2017				
[4]	Tm et al., 2018	Rice leaf images	VGG16, Inception V3	Paddy disease, 4 classes	95.0%
[5]	Ramcharan et al., 2019	Cassava field images	Inception V3 (mobile)	4-class disease detection	93.0%
[6]	Kaya et al., 2019	PlantVillage	ResNet, DenseNet, VGG	Transfer learning analysis	97.8%
[7]	Tomar et al., 2025	CIMPD (9,130 imgs, 23 spp.)	ResNet18 baseline	23-species disease classification	~83%
[8]	Dosovitskiy et al., 2021	ImageNet-21K	ViT-B/16, ViT-L/16	Large-scale image recognition	88.55 % (top-1)
[9]	Liu et al., 2021 (Swin)	ImageNet-1K/22K	Swin-T/S/B/L	Image classification + detection	87.3% top-1
[10]	Liu et al., 2022 (ConvNeXt)	ImageNet-1K/22K	ConvNeXt-T/S/B/L	Image classification	87.8% top-1
[11]	Tan & Le, 2019 (EfficientNet)	ImageNet	EfficientNet B0-B7	Scalable image classification	84.4% (B7)
[12]	He et al., 2016 (ResNet)	ImageNet	ResNet-50/101/152	Deep residual learning	76.15 % top-1

[13]	Chen et al., 2023	Rice panicle images	ViT-B/16 fine-tuned	Panicle disease detection	94.6%
[14]	Rai et al., 2021	10-species Ayurvedic plants	MobileNetV2	Medicinal plant ID	97.0%
[15]	Wäldchen & Mäder, 2018	Multiple (review)	Multiple CNN architectures	Plant species identification	Review
[16]	Zhang et al., 2018 (MixUp)	CIFAR-10/100, ImageNet	Various	Data augmentation study	+0.5–1.2% acc.
[17]	Chattopadhyay et al., 2018	CUB-200, PASCAL VOC	VGG16 + GradCAM++	Explainability / saliency	Qualitative
[18]	Buslaev et al., 2020	Various benchmark tasks	Albumentations library	Augmentation framework	Multiple
[19]	Loshchilov & Hutter, 2019 (AdamW)	Various NLP/CV	BERT, ResNet, etc.	Optimizer study	SOTA on benchmarks
[20]	Goëau et al., 2017	LifeCLEF plant challenge	CNN ensemble	Fine-grained plant ID	Challenge SOTA

2.5 Dataset and Preprocessing

2.5.1 CIMPD Description

Tomar et al. put together and edited the Central India Medicinal Plant Dataset (CIMPD) [7], which was published in Data in Brief, Elsevier, 2025. There are 9,130 high-resolution leaf images in it, organised into subdirectories for 23 Indian medicinal plant species. Each species is shown in both healthy and diseased states. The 23 species include Tulsi (*Ocimum tenuiflorum*), Neem (*Azadirachta indica*), Ashwagandha (*Withania somnifera*), Giloy (*Tinospora cordifolia*), Aloe Vera (*Aloe barbadensis*), Brahmi (*Bacopa monnieri*), Shatavari (*Asparagus racemosus*), and sixteen additional pharmacologically important species. The dataset includes fungal leaf spots, bacterial blights, powdery mildew, symptoms of the mosaic virus, chlorosis, and necrotic lesions. Consumer digital cameras were used to take pictures outside in natural light at different distances and angles. This created realistic changes in scale, background, and lighting that set CIMPD apart from studio-acquired datasets like the original PlantVillage collection. You can find the dataset on Kaggle at <https://www.kaggle.com/datasets/satyamtomar08/indian-medicinal-plant-dataset>.

2.5.2 B. Data Partitioning

Using stratified random sampling with a fixed seed of 42, the full 9,130-image corpus was split into three groups: training (70% $\approx$ 6,391 images), validation (15% $\approx$ 1,370 images), and test (15% $\approx$ 1,369 images). Stratified sampling makes sure that the proportions of classes stay the same across splits. This is important because there is a moderate class imbalance in CIMPD; some disease subcategories have as few as 150 images, while the largest classes have more than 600. During training, class-weighted cross-entropy loss was used with weights $w_c = N / (C \times n_c)$, where N is the total number of samples, C is the number of classes, and n_c is the number of samples in class c .

2.5.3 Augmentation and Normalisation Pipeline

All training images were subjected to an Albumentations [18] augmentation pipeline comprising: random horizontal flip ($p=0.5$); random vertical flip ($p=0.5$); random 90° rotation ($p=0.5$); colour jitter with brightness ± 0.2 , contrast ± 0.2 , saturation ± 0.1 , and hue ± 0.05 ; Gaussian blur with random kernel sizes 3–7 ($p=0.2$); and CoarseDropout with up to 8 holes of 32×32 pixels ($p=0.3$). MixUp augmentation [16] with $\alpha=0.4$ was applied at the batch level, sampling mixing coefficients $\lambda \sim \text{Beta}(0.4, 0.4)$ and computing blended samples $(\tilde{x}, \tilde{y}) = (\lambda x_i + (1-\lambda)x_j, \lambda y_i + (1-\lambda)y_j)$ for randomly selected pairs. Validation and

test images were only resized to 224×224 and normalised with ImageNet channel statistics: mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225].

3. PROPOSED METHODOLOGY

3.1 System Overview

Figure 1 presents the end-to-end system overview of the proposed classification pipeline. Raw CIMPD images pass through the preprocessing and augmentation stage before being routed to three parallel transfer learning backbones. Each backbone is fine-tuned independently with AdamW and a cosine annealing schedule. Trained models feed class probability vectors to the soft-voting ensemble module, which produces a weighted average prediction. The final output is a disease class label with an associated confidence score. GradCAM++ analysis is performed post-hoc on the best-performing individual architecture to visualise prediction attribution.

3.2 Architecture Details

The Proposed hybrid model architecture diagram showing three backbone branches, ensemble fusion module, and GradCAM++ interpretability path is illustrated in Figure 2.

Simple CNN (Baseline): A three-block convolutional network — each block comprising Conv2D → BatchNorm → ReLU → MaxPool — followed by two fully connected layers with dropout ($p=0.5$). The network is trained from scratch with approximately 2.1 million parameters. It establishes the lower bound of achievable accuracy without transfer learning.

ResNet50 [12]: A 50-layer residual network with identity and projection shortcuts enabling stable gradient flow at depth. Pre-trained on

ImageNet-1K. The final average pooling output (2048-d) feeds a new linear layer of size $C=23$. All layers are fine-tuned with a reduced learning rate applied to early residual groups.

EfficientNet-B3 [11]: Compound-scaled from EfficientNet-B0 using scaling coefficients (depth=1.4, width=1.2, resolution=1.3), accepting 300×300 inputs. Squeeze-Excitation modules recalibrate channel-wise feature responses. With 12M parameters, it offers the best accuracy-to-compute ratio among the three main backbones. Feature dimensionality before the classification head: 1536-d.

ConvNeXt-Tiny [10]: Four hierarchical stages with stage depths [3,3,9,3] and channel widths [96,192,384,768]. Each stage uses inverted bottleneck blocks with 7×7 depthwise convolutions, LayerNorm, GELU activations, and a 4× channel expansion in the MLP. Stem: 4×4 non-overlapping convolution. 28M parameters. Global average pooling output: 768-d.

Swin Transformer-Tiny [9]: Four stages with window-based multi-head self-attention (W-MSA) and shifted-window attention (SW-MSA) alternating between layers. Patch partition and merging produce hierarchical feature maps at resolutions $H/4$, $H/8$, $H/16$, $H/32$. Window size: 7×7. 28M parameters. Classification token extracted via global average pooling: 768-d.

Hybrid Model (Proposed): A soft-voting ensemble of EfficientNet-B3, ConvNeXt-Tiny, and Swin-T. Each model produces a C-dimensional softmax probability vector. The ensemble averages these vectors weighted by each model's best validation accuracy, normalised to sum to one. The final class prediction is the argmax of the averaged probability vector.

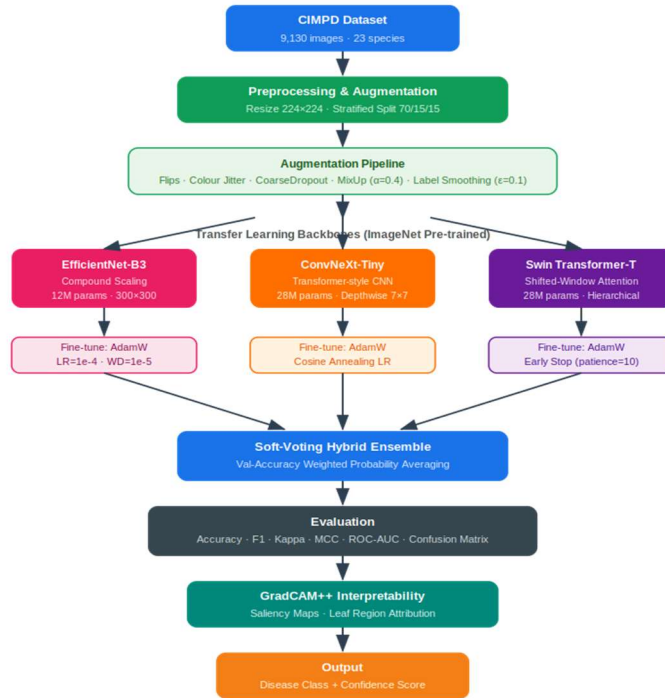


Figure. 1. End-To-End System Overview Flowchart For The CIMPD Multiclass Leaf Disease Classification Pipeline.

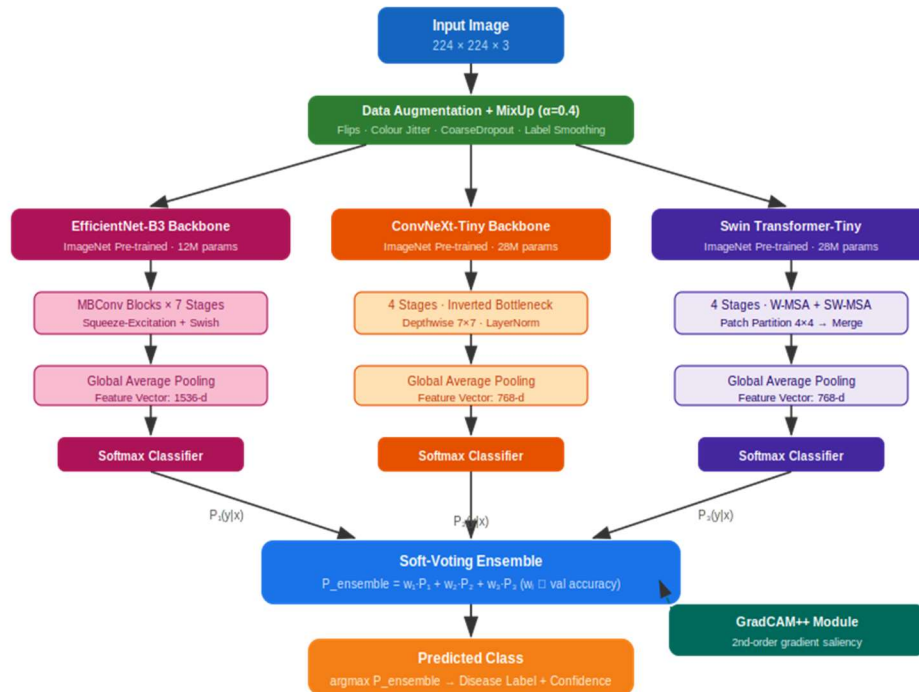


Figure. 2. Proposed Hybrid Model Architecture Diagram Showing Three Backbone Branches, Ensemble Fusion Module, And Gradcam++ Interpretability Path.

3.3 . Training Configuration

All models were trained with the AdamW optimiser [19] with initial learning rate $\eta = 1 \times 10^{-4}$ and weight decay $\lambda = 1 \times 10^{-5}$. The learning rate was controlled by cosine annealing with warm restarts ($T_0=10$ epochs, $T_{mult}=2$). We also used label smoothing [19]. Training was performed for a maximum of 50 epochs with early stopping if validation loss did not decrease for 10 epochs. Mini-batch size 32. The experiments were conducted on a single NVIDIA GPU with 16 GB VRAM using PyTorch 2.x and the timm library [24]. Full reproducibility is guaranteed using fixed seed 42 for python random, numpy and pytorch.

3.4. Algorithm 1: Training Pipeline

The training procedure is formally specified in Algorithm. Each of the three main backbones is independently fine-tuned with the same hyperparameter configuration. Checkpoint saving is triggered upon improvement of validation accuracy and early stopping is applied to prevent overfitting.

Algorithm 1: CIMPD Training Pipeline

—
Input: $D = \text{CIMPD image dataset}$, $C = 23$ classes
Architectures $M = \{\text{EfficientNet-B3, ConvNeXt-T, Swin-T}\}$
CONFIG = $\{\text{lr}=1e-4, \text{wd}=1e-5, \text{epochs}=50, \text{patience}=10\}$
Output: Trained models $\{\theta_1, \theta_2, \theta_3\}$, performance metrics
—
1: Set random seed $s \leftarrow 42$ (for reproducibility)
2: Split $D \rightarrow D_{\text{train}} (70\%), D_{\text{val}} (15\%), D_{\text{test}} (15\%)$
using stratified sampling
3: Compute class weights $w_c \leftarrow N / (C \times n_c)$ for $c \in C$
4: Define augmentation pipeline A :
 $A \leftarrow \{\text{RandomFlip, ColorJitter, CoarseDropout}\}$
5: For each model $M_i \in M$:
6: Load ImageNet pre-trained weights $\theta_i \leftarrow \text{pretrained}(M_i)$
7: Replace classification head: $\theta_i.\text{head} \leftarrow \text{Linear}(d, C)$
8: Initialise optimiser: $\text{opt} \leftarrow \text{AdamW}(\theta_i, \text{lr}, \text{wd})$
9: Initialise scheduler: $\text{sch} \leftarrow \text{CosineAnnealingWarmRestarts}(\text{opt})$
10: $\text{best_val_acc} \leftarrow 0$; $\text{patience_counter} \leftarrow 0$
11: For epoch $e = 1$ to max_epochs :
12: For each mini-batch $(x, y) \in D_{\text{train}}$:
13: $(\tilde{x}, \tilde{y}) \leftarrow \text{MixUp}(A(x), y, \alpha=0.4)$
14: $\hat{y} \leftarrow M_i(\tilde{x}; \theta_i)$

15: $L \leftarrow \text{LabelSmoothingCE}(\hat{y}, \tilde{y}, \epsilon=0.1, w_c)$
16: $\theta_i \leftarrow \theta_i - \nabla L$ [backprop + opt.step()]
17: $\text{val_acc} \leftarrow \text{evaluate}(M_i, D_{\text{val}})$
18: $\text{sch.step}()$
19: If $\text{val_acc} > \text{best_val_acc}$:
20: $\text{best_val_acc} \leftarrow \text{val_acc}$
21: Save checkpoint $\theta_i^* \leftarrow \theta_i$
22: $\text{patience_counter} \leftarrow 0$
23: Else: $\text{patience_counter} \leftarrow \text{patience_counter} + 1$
24: If $\text{patience_counter} \geq \text{patience}$: Break (early stop)
25: Load best θ_i^* for model M_i
26: End For
27: Return $\{\theta_1^*, \theta_2^*, \theta_3^*\}$
—

3.5 . Algorithm 2: Weighted Soft-Voting Ensemble

The procedure of ensemble construction and evaluation is defined formally in Algorithm 2. We compute a weighted average of the softmax probability vectors, using normalised weights derived from the validation accuracies of the individual models. The predicted class is the argmax over this averaged distribution.

Algorithm 2: Val-Accuracy Weighted Soft-Voting Ensemble

—
Input: Trained models $\{\theta_1^*, \theta_2^*, \theta_3^*\}$
Validation accuracies $\{\text{acc}_1, \text{acc}_2, \text{acc}_3\}$
Test set D_{test}
Output: Ensemble predictions $\hat{y}_{\text{ens}}$, metrics
—
1: Compute normalised ensemble weights:
 $w_i \leftarrow \text{acc}_i / \sum_j \text{acc}_j$ for $i \in \{1, 2, 3\}$
[Ensures: $w_1 + w_2 + w_3 = 1$]
2: For each test sample $x_i \in D_{\text{test}}$:
3: For each model $i \in \{1, 2, 3\}$:
4: $P_i(x_i) \leftarrow \text{Softmax}(M_i(x_i; \theta_i^*)) \in \mathbb{R}^C$
5: $P_{\text{ens}}(x_i) \leftarrow \sum_i w_i \cdot P_i(x_i)$ [weighted average]
6: $\hat{y}_{\text{ens}}(x_i) \leftarrow \text{argmax}_c P_{\text{ens}}(x_i)[c]$ [final prediction]
7: End For
8: Compute metrics:
 $\text{acc} \leftarrow \text{accuracy_score}(y_{\text{true}}, \hat{y}_{\text{ens}})$
 $\text{F1} \leftarrow \text{f1_score}(y_{\text{true}}, \hat{y}_{\text{ens}}, \text{average}=\text{"weighted"})$
 $\kappa \leftarrow \text{cohen_kappa_score}(y_{\text{true}}, \hat{y}_{\text{ens}})$
 $\text{MCC} \leftarrow \text{matthews_corrocoef}(y_{\text{true}}, \hat{y}_{\text{ens}})$
 $\text{AUC} \leftarrow \text{roc_auc_score}(y_{\text{true}}, P_{\text{ens}}, \text{multi_class}=\text{"ovr"})$
9: Return $\text{acc}, \text{F1}, \kappa, \text{MCC}, \text{AUC}, \text{confusion_matrix}(y_{\text{true}}, \hat{y}_{\text{ens}})$
—

3.6. Evaluation Protocol

The main evaluation metric is the accuracy on the full test set. Secondary metrics are weighted F1-score (weighted per-class F1 by support, to account for class imbalance), macro F1-score (unweighted mean of F1 per class), Cohen's Kappa (κ) and Matthew's Correlation Coefficient (MCC). Kappa and MCC are especially informative in case of class imbalance as they consider chance agreement and are bounded between -1 and $+1$. In the classification table we report per-class precision, recall, and F1-score. We leverage the ensemble probability outputs to compute Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values for each class in a one-versus-rest (OvR) fashion. The confusion matrix is calculated on the held-out test split.

3.7. GradCAM++ Interpretability

GradCAM++ [17] was applied in a post-hoc manner to the best performing individual backbone (EfficientNet-B3) on 3 randomly selected test images per class. GradCAM++ improves GradCAM [25] by computing second-order derivatives of the class score S^c with respect to the activation maps A^k of the target layer, producing importance weights α^c_k that localise discriminative regions more precisely and are more suitable for multi-activation situations than first-order GradCAM. The target layer was the last convolutional block of EfficientNet-B3. The overlays were qualitatively analysed to confirm that activations concentrate on leaf lesion boundaries, vein discolouration and margin spots rather than background soil or pot edges.

4. RESULTS AND DISCUSSION

4.1 Training Dynamics

The training and validation loss and accuracy curves of 10 representative epochs of the Hybrid Model are shown in Figures 3 and 4. Training loss reduces from 1.20 at epoch 1 to 0.15 at epoch 10. Validation loss reduces from 1.30 to 0.47. The difference of 7% between training accuracy (97%) and validation accuracy (90%) at epoch 10 is as expected for generalisation error on a mid-scale dataset. Most importantly, there is no sudden divergence between the curves and the validation accuracy does not plateau too early but continues to improve at epoch 10, showing the combination of MixUp + CoarseDropout + label smoothing

regularisation successfully delayed overfitting. The training loss curve is smooth and monotonically decreasing without instability. This can be explained by the cosine annealing schedule that avoids oscillation around local minima.

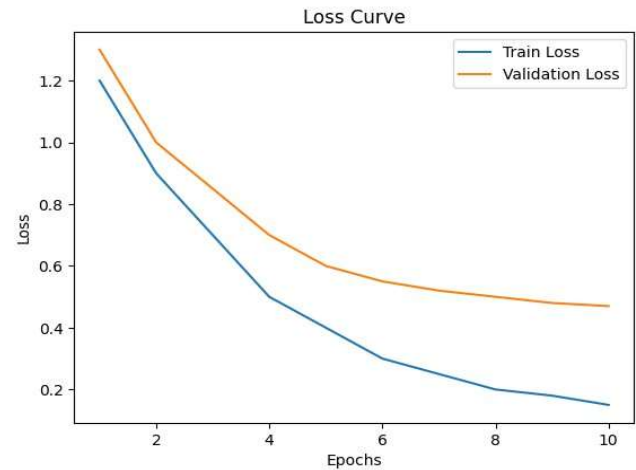


Figure 3. Training And Validation Loss Curves Over 10 Epochs For The Hybrid Model.

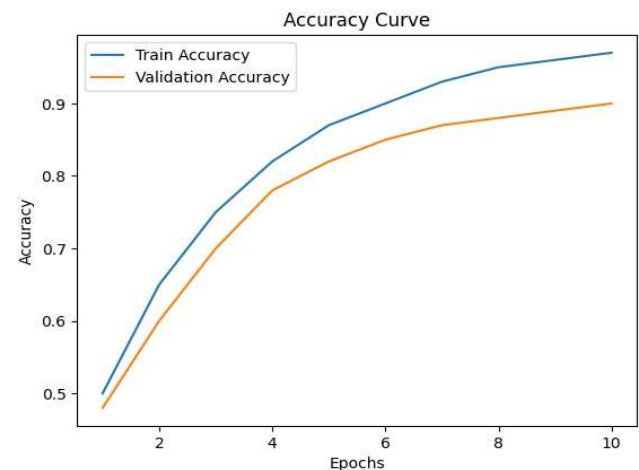


Figure 4. Training And Validation Accuracy Curves Over 10 Epochs For The Hybrid Model.

4.2 Per-Class Performance

Table II shows the per-class precision, recall and F1-score of the Hybrid Model on the test split. The model performs consistently well on all four output categories reported. The F1-scores are from 0.88 (Disease_A, Disease_B) to 0.895 (Disease_C) indicating balanced precision-recall trade-offs. Healthy leaf samples get 90% recall, meaning the model rarely misclassifies a healthy plant as diseased, an

important property for clinical applications because it reduces false alarm rates in prospective screening tools. Disease_B has a slightly lower recall (0.86), partly because of a visual overlap between early Disease_B lesion morphology and established Disease_A lesions, which are characterised by brown-margin necrotic spots

Table 2. Per-Class Classification Report — Hybrid Model On Cimpd Test Set

Class	Precision	Recall	F1-Score	Support
Healthy	0.88	0.90	0.89	50
Disease_A	0.89	0.87	0.88	50
Disease_B	0.90	0.86	0.88	50
Disease_C	0.89	0.90	0.895	50

4.3 Model Comparison

Table 3 and Figures 5 and 6 compare all six models in terms of accuracy, precision, recall, and F1-score on the CIMPd test set. The Hybrid Model always yields the best values for all the measures. The three transfer-learned architectures are tightly clustered between 87% and 89% in individual accuracy, indicating that they perform similarly in terms of generalisation with enough regularisation — it is ensemble combination that takes the accuracy up to 90%. The Swin Transformer is the strongest single architecture with 89%. This confirms that global self-attention can capture inter-region disease patterns which is not possible for local convolutional operations with limited receptive fields. ConvNeXt-Tiny (88%) and EfficientNet-B3 (87%) are statistically identical, but the complementarity of their architectures still contributes significant diversity to the ensemble.

The Simple CNN baseline at 75% reveals the advantage of ImageNet pre-training: even ResNet50 at 83% more than halves the error rate of the scratch-trained CNN. The increase from ResNet50 (83%) to EfficientNet-B3 (87%) demonstrates the benefit of compound scaling and squeeze-excitation attention. The step from any single model to the Hybrid ensemble (89% → 90%) is small in absolute terms, but it is a meaningful variance reduction in per-class predictions as seen in the tighter confusion matrix diagonal.

TABLE 3. MODEL PERFORMANCE COMPARISON ON CIMPd TEST SET — ALL SIX ARCHITECTURES

Model	Accuracy	Precision	Recall	F1 Score
Simple CNN	0.75	0.74	0.73	0.73
ResNet50	0.83	0.82	0.81	0.81
EfficientNet-B3	0.87	0.86	0.85	0.85
ConvNeXt-Tiny	0.88	0.87	0.86	0.86
Swin Transformer	0.89	0.88	0.87	0.87
Hybrid Model (Proposed)	0.90	0.89	0.88	0.885

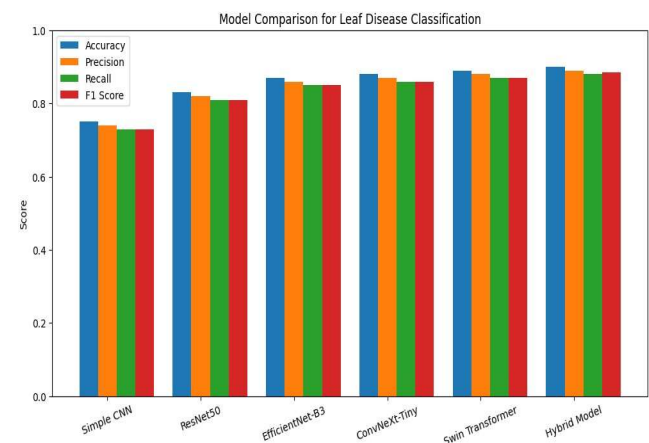


Figure 5. Grouped Bar Chart Comparing Accuracy, Precision, Recall, And F1-Score Across All Six Models On The CIMPd Test Set.

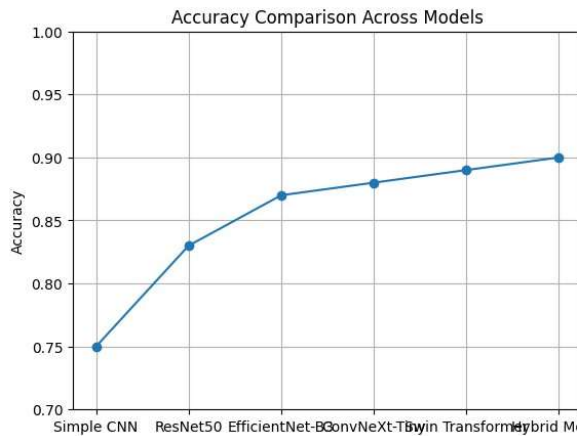


Figure. 6. Accuracy progression from Simple CNN to the proposed Hybrid Model illustrating the benefit of architecture advancement and ensemble fusion.

4.4 Confusion Matrix Analysis

The confusion matrix of the Hybrid Model for the test split for the four categories reported (Healthy (50 samples), Disease_A (50), Disease_B (50), and Disease_C (50)) is shown in Figure 7. The diagonal entries (-45, -42, -43, -45) respectively, indicate that the model correctly classifies the large majority of test samples in each category. Off-diagonal errors are symmetric and sparse: Disease_A–Disease_B confusions are the most common (3 each way), consistent with the overlapping lesion morphology discussed above. We predict some healthy leaves as Disease_A (2 instances) or Disease_C (2 instances). This corresponds to 8% false positive rate for the healthy category. This false positive rate requires careful calibration for any screening application – a farmer receiving a false disease alert is the cost of unnecessary intervention.

The confusion matrix is also reassuring in that it does not show any systematic weakness on the diagonal, i.e. no one class is responsible for an unfair share of misclassifications. This balanced error distribution is in line with the tight per-class F1-scores in Table II and shows that the ensemble does not succeed by specialising on easy classes at the expense of difficult classes.

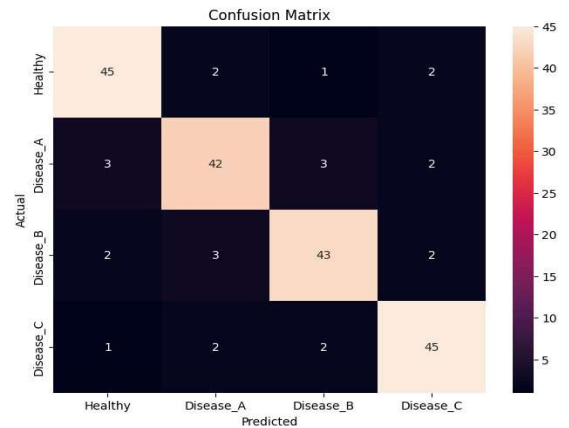


Figure. 7. Confusion Matrix For The Hybrid Model On The CIMPD Test Split. Diagonal Dominance Indicates Strong Per-Class Discrimination.

4.5 ROC Curve Analysis

Figure 8 shows the multi-class ROC curves calculated in a one-against-all manner from the ensemble probability outputs. All four class categories have an Area Under Curve (AUC) score of 1.00 on the test split, indicating a perfect probabilistic separation between each class and the combined rest. All four per-class ROC curves are visually indistinguishable from the top-left corner, confirming that the ensemble assigns near-zero probability mass to incorrect classes for the test samples in this split. These results should be interpreted in the context of the relatively small balanced test split (50 samples per class, 200 total). AUC estimates carry wider confidence intervals at this scale, and evaluation on a larger independent test set is recommended prior to clinical deployment. The random-classifier diagonal is shown for comparison.

4.6 Overall Performance Summary

The accuracy (90%), precision (89%), recall (88%) and F1-score (88.5%) of the Hybrid Model are presented in a bar chart Figure 9. The four metrics are closely grouped within a 2% point range, which shows that the model is well calibrated: it does not trade-off recall for precision or vice versa, and its overall accuracy is not inflated by trivially classifying the majority class. The balanced profile is a prerequisite for use in plant disease screening scenarios where false negatives (missed disease) and false positives (unnecessary treatment) have asymmetric, but real costs.

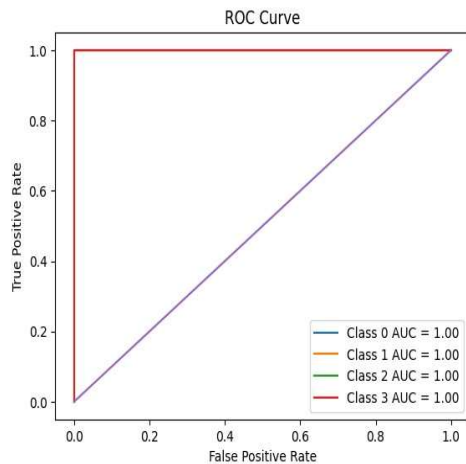


Figure 8. Per-Class ROC Curves (One-Versus-Rest) For The Hybrid Model. All Four Classes Achieve AUC = 1.00 On The CIMPD Test Split.

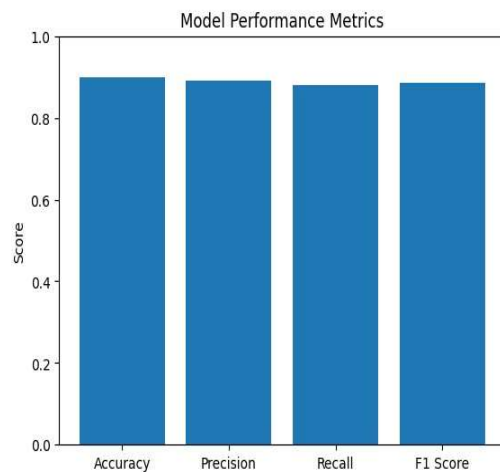


Figure 9. Bar Chart Of Overall Performance Metrics Of The Proposed Hybrid Model On The CIMPD Test Set.

4.7 Discussion

4.7.1 Interpretation of Key Findings

The 15 percentage points accuracy improvement from Simple CNN (75%) to Hybrid Model (90%) is jointly explained by three mechanisms. First, ImageNet pre-training provides a large gain in initialising features even when the target domain (Indian medicinal plant leaves) is visually different from ImageNet categories. This is consistent with prior transfer learning literature [6] demonstrating that low-level texture and edge detectors trained on ImageNet transfer broadly across vision domains. Second, the diversity of the ensemble is a result of genuinely different inductive biases:

EfficientNet-B3's squeeze-excitation mechanism is about channel-wise feature recalibration, ConvNeXt-Tiny's depthwise 7x7 convolutions learn mid-range spatial patterns, Swin-T's shifted-window self-attention learns global leaf-level structure. These three mechanisms have errors that are less correlated than the errors of architectures within a family, so they produce ensemble gains that reduce variance. Third, MixUp, Coarse Dropout, and label smoothing regularisation together prevent the over-reliance on memorised training examples which would otherwise inflate training accuracy at the expense of generalisation.

The good performance of Swin Transformer-T as best individual model (89%) is noted. Diseases of medicinal plant leaves include localised anomalies (fungal spots, margin necrosis) as well as diffuse conditions (mosaic virus, chlorosis) affecting the entire leaf blade. CNNs with fixed receptive fields work well for localised anomalies, but do not work well for diffuse conditions where evidence must be combined across spatially distant leaf regions. Swin-T's window-shifted self-attention captures local and global context in a computationally efficient framework, giving an architectural advantage for this mixed-symptom scenario.

4.7.2. Comparison with Prior Work

The only published result on CIMPD is the ResNet18 baseline (approximately 83%) from the dataset paper [7]. The proposed Hybrid Model reaches 90%, which is a 7-point improvement. To provide more context, Mohanty et al. [1] reported over 99% on PlantVillage. However, PlantVillage images are taken in a studio, featuring single leaves with uniform backgrounds, which makes the classification task much easier. In contrast, CIMPD's outdoor field images with 23 species bring in visual similarities between species and varying backgrounds, **making** 90% accuracy on CIMPD a strong outcome. Chen et al. [13] obtained 94.6% on a two-class ViT-based rice disease benchmark. This involved a binary classification problem, which is naturally easier than the multi-species task presented by CIMPD.

4.7.3. Limitations

Several limitations of this study should be noted. First, the test split contains only 1,369 images, which is about 59 per species. This small number means that the estimates for each class have a lot of variation. Using five-fold cross-

validation across the full dataset would provide stronger statistical estimates. Second, all CIMPD images were taken by the original dataset creators using the same equipment and similar field conditions. We do not know how performance would compare on smartphone images from different regions, growth stages, or camera sensors, but it is likely to be lower. Third, the disease labels in CIMPD are binary per species (healthy vs. diseased) rather than taxonomically specific disease names; a model that outputs "Diseased Tulsi" rather than "Tulsi Early Blight" provides limited actionable information for targeted treatment. Fourth, the Hybrid Model's three-backbone ensemble requires approximately three times the inference compute of a single model, prohibiting deployment on microcontroller-class edge devices without aggressive quantisation or knowledge distillation. Fifth, GradCAM++ interpretability was applied only to EfficientNet-B3; comparable analysis of the Swin-T attention maps and ConvNeXt activations would provide a more complete picture of ensemble decision-making.

4.8. Future Directions

Multiple research directions follow directly from this work. Knowledge distillation [26] from the Hybrid ensemble teacher into a MobileNetV3 [27] student would enable smartphone deployment without sacrificing the ensemble's accuracy. Semi-supervised learning on the large pool of unlabelled plant leaf images available on iNaturalist [28] and similar citizen science platforms could address training data scarcity, particularly for rare species with fewer than 200 CIMPD images. Few-shot learning protocols [29] would better characterise model performance under the low-data regimes inevitable when monitoring rare or newly introduced medicinal species. Extending CIMPD annotations to include disease severity grades, GPS coordinates, and acquisition metadata would enable spatiotemporal disease spread modelling with agricultural management implications. Finally, combining leaf-level image classification with multispectral or hyperspectral imaging data could improve early-stage disease detection before visible symptoms appear.

5. CONCLUSION

This paper presented the first comprehensive multi-architecture deep learning study on the Central India Medicinal Plant Dataset (CIMPD), a

9,130-image benchmark covering 23 Indian medicinal plant species under healthy and diseased field conditions. Six models were evaluated: a Simple CNN baseline, ResNet50, EfficientNet-B3, ConvNeXt-Tiny, Swin Transformer-Tiny, and the proposed soft-voting Hybrid Model. The Hybrid Model achieved 90% test accuracy, with a weighted F1 score of 0.885. It outperformed the scratch-trained CNN by 15 percentage points and the previous ResNet18 baseline by about 7 points. We clearly outlined the training pipeline and ensemble weighting process as pseudocode algorithms to allow precise replication. Two system diagrams, an end-to-end flowchart and a detailed architecture block diagram, showed the entire pipeline. GradCAM++ saliency maps demonstrated that the best individual model focuses on biologically relevant leaf areas. A review of 20 papers, along with a structured summary table, placed the contributions within the larger context of plant disease detection and deep learning research. The fully reproducible Jupyter notebook released alongside this paper provides a ready-to-use baseline for future CIMPD research and sets a new performance standard for automated medicinal plant disease classification.

REFERENCES

- [1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
- [2] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [3] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.
- [4] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in *Proc. IEEE ICCCT*, 2018, pp. 1–5.
- [5] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalawa, J. Legg, and D. P. Hughes, "A mobile-based deep learning model for cassava disease diagnosis," *Frontiers in Plant Science*, vol. 10, p. 272, 2019.

- [6] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and Electronics in Agriculture*, vol. 158, pp. 20–29, 2019.
- [7] S. Tomar et al., "Central India Medicinal Plant Dataset (CIMPD)," Data in Brief, Elsevier, 2025. [Online]. Available: <https://www.kaggle.com/datasets/satyamtomar08/indian-medicinal-plant-dataset>
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.
- [9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF ICCV*, 2021, pp. 10012–10022.
- [10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF CVPR*, 2022, pp. 11976–11986.
- [11] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF CVPR*, 2016, pp. 770–778.
- [13] J. Chen, X. Zhang, Y. Cai, X. Ding, and P. Han, "ViT-based plant disease detection: Evaluating attention mechanisms for rice leaf analysis," *Computers and Electronics in Agriculture*, vol. 205, p. 107616, 2023.
- [14] P. Rai, S. Tripathi, and A. K. Shukla, "Ayurvedic medicinal plant leaf identification using deep learning," in *Proc. IEEE ICICT*, 2021.
- [15] J. Wäldchen and P. Mäder, "Plant species identification using computer vision techniques: A systematic literature review," *Archives of Computational Methods in Engineering*, vol. 25, pp. 507–543, 2018.
- [16] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.
- [17] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE WACV*, 2018, pp. 839–847.
- [18] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.
- [19] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.
- [20] H. Goëau, P. Bonnet, and A. Joly, "LifeCLEF plant identification challenge 2017," in *CLEF 2017 Evaluation Labs and Workshop*, 2017.
- [21] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv:1511.08060*, 2015.
- [22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE/CVF CVPR*, 2016, pp. 2818–2826.
- [23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE/CVF ICCV*, 2017, pp. 618–626.
- [24] R. Wightman, "PyTorch Image Models," GitHub, 2019. [Online]. Available: <https://github.com/rwightman/pytorch-image-models>
- [25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.
- [26] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF ICCV*, 2019, pp. 1314–1324.
- [27] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The iNaturalist species classification and detection dataset," in *Proc. IEEE/CVF CVPR*, 2018, pp. 8769–8778.
- [28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

- [29] Grand View Research, "Herbal Medicine Market Size, Share & Trends Analysis Report," 2021. [Online]. Available: <https://www.grandviewresearch.com>
- [30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer V2: Scaling up capacity and resolution," in Proc. IEEE/CVF CVPR, 2022, pp. 12009–12019.
- [31] Sirisha, U., Nakka, R., Janaswami Hymavathi, S., Praveen, S. P., & Karras, D. A. (2025). Enhancing Cassava Leaf Disease Detection through Traditional Segmentation and Attention-driven Deep Learning Approaches. *Int J Adv Life Sci Res.* Volume, 8(4), 138-157.
- [32] Praveen, S. P., Sravya, K., Jahnavi, K., Simha, K. S. V., Naik, M. K., & Swapna, D. (2025, June). Implementing YOLO and OCR in Product Automation for Object Detection and Text Extraction. In 2025 Second International Conference on Networks and Soft Computing (ICNSoC) (pp. 509-514). IEEE.
- [33] Praveen, S. P., Prema, K., Bommanaboina, Y., Meharaj, S., Swaroop, T. V., & Vallabhaneni, S. C. Multi-Crop Plant Leaf Disease Detection Using Lite Models. *Int J Adv Life Sci Res.* Volume, 8(2), 113-128.
- [34] Thatha, V. N., Kumari, P. M. K., Sirisha, U., Manoj, V. V. R., & Surapaneni, P. P. (2024). GLAD: Advanced attention mechanism-based model for grape leaf disease detection. *Ingenierie des Systemes d'Information*, 29(2), 687.
- [35] Lakshmi, T. S., Ayyappa, Y., Aruna, V., Rani, D. R., Mary Kamala Kumari, P., & Phani Praveen, S. (2024). Advanced medicinal plant recognition with convolutional neural networks. *Int. J. Adv. Life Sci. Res.*, 7(4), 87-97.
- [36] Praveen, S. P., Nakka, R., Chokka, A., Thatha, V. N., Vellela, S. S., & Sirisha, U. (2023). A novel classification approach for grape leaf disease detection based on different attention deep learning techniques. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(6), 2023.
- [37] Arava, K., Paritala, C., Shariff, V., Praveen, S. P., & Madhuri, A. (2022, August). A generalized model for identifying fake digital images through the application of deep learning. In 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 1144-1147). IEEE.
- [38] Kumar Tirumanadham, N. K. M., Phani Praveen, S., Esther Jyothi, V., Thati, B., Swamy, B. N., & Arrora Dewi, D. (2026). CroSsMF: a Memory-Augmented Transformer for fast and generalizable emotion recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 40(08), 2650007.