Optimizing Deep Learning Model Ensembles for Plant Disease Detection through Ablation and Correlation Analysis

Nawel Ghrieb[1*], Tahar Guerram[2], Othaila Chergui[3]
[1]Laboratory of Mathematics, Informatics and Systems (LAMIS), Echahid Cheikh Larbi Tebessi university, Tebessa, 12000, Algeria
[2] RelaCS2Laboratory, Larbi Ben M'hidi University, Oum El Bouaghi, 04000, Algeria
[3]Laboratory of Signals and Smart Systems, Echahid Cheikh Larbi Tebessi university, Tebessa, 12000, Algeria
E-mail: nawel.ghrieb@univ-tebessa.dz, tahar_guerram@yahoo.fr, othaila.chergui@univ-tebessa.dz
[*]Corresponding author

Keywords: Plant disease detection, deep learning, ensemble learning, ablation study, vision transformer, Swin transformer, PlantDoc

Received: July 19, 2025

Early detection of plant diseases is crucial for global food security. While Deep Learning ensemble techniques are widely adopted to improve performance, the assumption that simply aggregating models is always beneficial should be nuanced. This paper addresses this issue by conducting a rigorous analysis of an ensemble of four state-of-the-art architectures (Swin Transformer, Vision Transformer, EfficientNetV2, ConvNeXt) on the PlantDoc dataset, a benchmark known for its complexity.
Our approach is twofold. First, we conduct a systematic ablation study to assess how each model contributes to the ensemble's performance. This analysis leads to the counter-intuitive finding that an optimized three-model subset (Swin, ViT, and EfficientNetV2) outperforms the full four-model ensemble. Quantitatively, the pruned ensemble achieves a Macro F1-score of 0.7503 and an accuracy of 0.7619, compared to 0.7409 and 0.7500 for the full set, respectively. Second, to explain this phenomenon, we perform a prediction correlation analysis. This reveals significant predictive redundancy, stemming from the architectural similarities between ConvNeXt and the Transformer-based models, as the cause for this sub-optimality. These findings suggest a key principle for ensemble design: the predictive complementarity of the models is a more critical factor than their individual performance or the complexity of the aggregation method-a finding reinforced by our benchmark showing that even advanced strategies like calibrated voting and stacking failed to outperform the pruned ensemble. Our work thus positions the methodological pairing of ablation study and correlation analysis as an essential and pragmatic approach to optimize the performance of ensembles in computer vision.

Povzetek: Prispevek pokaže, da pri zgodnjem odkrivanju bolezni rastlin ni vedno bolje združiti več modelov: z analizo prispevkov in korelacij napovedi ugotovijo, da manjši, premišljeno izbran ansambel doseže boljše rezultate zaradi manjše redundance.

1 Introduction

Safeguarding crop health is a cornerstone of sustainable agriculture and food security. Plant diseases are responsible for significant yield losses worldwide annually. The rapid and accurate identification of these diseases by experts is often costly, time-consuming, and inaccessible in many regions. Facing this challenge, artificial intelligence, and more specifically Deep Learning, has emerged as a promising solution for automating plant disease diagnosis from leaf images.

From Convolutional Neural Network (CNN) architectures to the more recent Vision Transformers (ViT), numerous models have been successfully applied to this task. To further improve model performance, ensemble techniques, which combine the predictions of multiple models, have become a common practice. The underlying assumption is that models with diverse architectures will learn different feature representations and that their errors will be uncorrelated, thereby allowing the ensemble to correct individual weaknesses.

However, most works merely aggregate high-performing models without deeply analyzing the synergy or conflict between them. The question of whether a larger ensemble is always better remains open. Indeed, the existing literature, while rich in ensemble applications, often focuses on demonstrating overall improvement without systematically quantifying the contribution, whether positive or negative, of each constituent model. Ablation studies, when present, are rarely the main focus of the investigation, leaving the question of optimal ensemble composition largely unexplored. In this context, our contribution is threefold:

1. We evaluate and compare four modern architectures (Swin-Base, ViT-Base, EfficientNetV2-B3, ConvNeXt-Tiny) on the complex data of the PlantDoc dataset.

2. We implement a Soft Voting ensemble method and conduct a rigorous ablation study to quantify the contribution of each component.
3. We demonstrate that careful model selection is paramount and that a smaller, optimized ensemble can outperform a larger one, thereby providing guidelines for designing more effective ensembles.

The remainder of this paper is structured as follows. Section 2 reviews the related work on plant disease detection and the use of ensemble techniques. Section 3 details our systematic methodology, including the data preparation, the advanced training protocol for individual models, and our comprehensive protocol for ensemble optimization which covers the ablation study, the benchmark of alternative aggregation strategies, and the correlation analysis. Section 4 presents the quantitative results of these experiments, detailing the performance of individual architectures and all ensemble configurations, and concludes with an analysis of the optimal model's error patterns. Section 5 then provides an in-depth discussion, contextualizing our results by comparing them to the state of the art, analyzing their broader implications for ensemble design, and outlining the limitations of our study. Finally, Section 6 concludes the paper, summarizing our contributions and suggesting directions for future work.

2 Related work

Plant disease detection has undergone a fundamental transition from manual approaches to automated solutions. Traditional methods, though sometimes still used, are widely recognized as being costly, subjective, and labor-intensive. To overcome these limitations, research has shifted towards techniques that leverage computer vision and image processing, analyzing features such as texture, color, or shape to identify pathologies.

In this context, Machine Learning (ML) offered a first generation of effective tools. While these algorithms were applied to a wide range of agricultural tasks, their use in disease identification specifically highlighted a key limitation. Foundational studies in this area successfully used a variety of classical classifiers. This approach typically relied on a multi-stage process where engineered features like color, texture, and shape were manually extracted from a region of interest before classification. Representative studies include using a Support Vector Machine (SVM) to classify grape leaf diseases based on color and texture features [1], or a K-Nearest Neighbor (KNN) classifier for groundnut leaves after a similar feature extraction stage [2]. Although respectable accuracies were achieved, the performance of all these ML models remained fundamentally dependent on this manual feature engineering step. This process, which requires domain expertise to define relevant features, limited their ability to capture the full complexity of visual patterns and their adaptability to new diseases.

To overcome these obstacles, Deep Learning (DL) has established itself, offering the crucial advantage of automatic feature learning. One of the foundational works in this field was conducted by Mohanty et al., who used an AlexNet-type CNN to achieve 99.27% accuracy on the public PlantVillage dataset, which consists of images taken under controlled laboratory conditions [3]. Too et al. later reinforced these findings in a comparative study, reporting 99.75% accuracy with a DenseNet121 architecture on the same data [4].

This initial success with Convolutional Neural Networks (CNNs) was systematically confirmed by a series of literature reviews, from early analyses [5] up to more recent and comprehensive surveys [6], all establishing the consistent superiority of DL over traditional ML methods. The high effectiveness of this approach is now routinely demonstrated on focused tasks, such as the use of a custom lightweight CNN to detect Bacteriosis in peach leaves with 99% accuracy [7]. The drive for peak performance in such idealized environments has continued with the development of even more sophisticated architectures. A prime example is the work by Zhao et al., [8] who engineered a CNN fusing Inception structure with attention mechanisms to further perfect performance on this standard benchmark.

However, this very success on controlled datasets highlighted the next major challenge for the field: generalizing performance to 'in-the-wild' scenarios. This generalization failure is not trivial; for instance, as noted by Chandra et al., a powerful model like InceptionResNetV2 sees its performance collapse from over 99% on PlantVillage to 39.87% accuracy and a 38% F1-score when evaluated on the real-world complexities of the PlantDoc dataset [9]. To address these limitations, researchers such as Iftikhar et al. [10] have explored the use of optimized CNNs for early detection under challenging real-world conditions. Such scenarios, which involve images with complex backgrounds, variable lighting, and different angles, are precisely the conditions represented by the PlantDoc dataset used in our study.

Despite these emerging solutions, relying on single models remains unreliable due to their high sensitivity to the quality and diversity of training data. The robustness of models against background noise in images, symptom variability, and their effective deployment in the field remain active areas of research. It is precisely to address these challenges of robustness and generalization that research has turned to a complementary strategy: ensemble learning.

Ensemble learning [11], is a well-established strategy for enhancing overall model performance [12]. The core idea is to combine predictions from several models to achieve a more robust and stable final decision. Common techniques range from simple majority voting [13], [14] to more sophisticated strategies like weighting [15], soft voting (averaging probabilities) [16] and stacking [12]. A key limitation of these conventional methods, however, is their primary focus on the aggregation mechanism itself, operating under the implicit assumption that simply combining individually strong models is enough.

This assumption is an oversimplification. The fundamental literature on ensemble learning establishes that the effectiveness of an ensemble depends not so much on the individual performance of its members as on their

predictive diversity [11]. A group of models that make the same mistakes offers no advantage. This principle has led to the development of a vast research field dedicated to ensemble selection (or ensemble pruning), for which the state-of-the-art has been summarized in comprehensive literature reviews such as that of Sagi & Rokach [16]. Within this domain, several families of strategies have been successfully developed and applied. These include, for instance, clustering-based approaches, which were proposed as early as the beginning of the 2000s [17]. The effectiveness of these selection methods has been empirically demonstrated in comparative reviews that evaluate different techniques on complex tasks [18]. Other seminal works have analyzed heuristic search techniques like Backward Elimination, a method whose efficacy was originally demonstrated in a high-stakes domain: the medical diagnosis of pneumonia risk [19].

However, despite the proven utility of these selection methods in other complex domains, their systematic application remains surprisingly rare in the specific context of plant disease detection. The majority of recent studies that do employ ensembles tend to focus on simple aggregation mechanisms. For example, Astani et al. used an ensemble classification approach to achieve high accuracy on tomato diseases [20], and Shafik et al. introduced ensemble techniques to enhance model robustness [21]. While these works demonstrate the benefits of combining models, they operate under the implicit assumption that aggregating individually strong models is sufficient. This approach overlooks the crucial question of predictive redundancy and fails to investigate the internal composition of the ensemble itself.

This need for complementarity is particularly relevant with the emergence of state-of-the-art architectures like the Vision Transformer (ViT) [22], the Swin Transformer [23], and modern CNNs like ConvNeXt [24]. These models are rapidly being adopted for disease classification, with much of the research focusing on maximizing their individual performance. This trend is evident across different architectural families. In the domain of modern CNNs, for instance, the study by Chen et al. demonstrated that integrating attention mechanisms into a MobileNetV2 backbone could significantly improve accuracy for rice disease identification [25]. Similarly, for Vision Transformers, research has focused on exhaustive hyperparameter tuning, as illustrated by the study by Ouamane et al., which achieved 99.77% accuracy on the controlled PlantVillage dataset through an optimized ViT configuration [26]. Building on this, some studies have explored combining these architectures into fixed hybrid systems. Notable examples include the "PlantXViT" introduced by Thakur et al. [27] and the "TLMViT" model from Tabbakh et al. [28], both of which created monolithic models by fusing Transformer and CNN-based components.

However, while these studies showcase the power of these new architectures, they concentrate on their performance either in isolation or as fixed hybrid systems. How these models interact within a multi-model ensemble, and how to determine its optimal composition, remains a largely unexplored area. The fact that an optimized ConvNeXt is powerful does not explain how its predictions correlate with those of a Swin Transformer or a ViT within the same team. Understanding whether these architectures are predictively redundant or complementary is a crucial question. To our knowledge, no study has conducted a systematic ablation analysis coupled with a correlation study to empirically investigate this redundancy and complementarity among these specific architectural families for plant disease detection under complex conditions.

Our contribution aims precisely to fill this gap by providing empirical evidence that a carefully pruned, heterogeneous ensemble (Transformer + CNN) can outperform the full set of models by eliminating predictive redundancy. While the methods we employ (ablation and correlation) are conventional, their application to this specific problem yields novel insights into the optimal composition of modern vision model ensembles, providing pragmatic guidelines for practitioners.

3 Proposed methodology

Our methodological approach is designed first to maximize the performance of each individual model, and then to systematically optimize the ensemble's composition. The core pipeline for this process, illustrated in Figure 1, is broken down into several stages: data preparation (3.1, 3.2), a rigorous training protocol for four state-of-the-art architectures (3.3), and the construction of an ensemble optimized via a systematic ablation study (3.4 and 3.5). This empirical optimization is then explained through a final correlation analysis, also detailed in section 3.5.

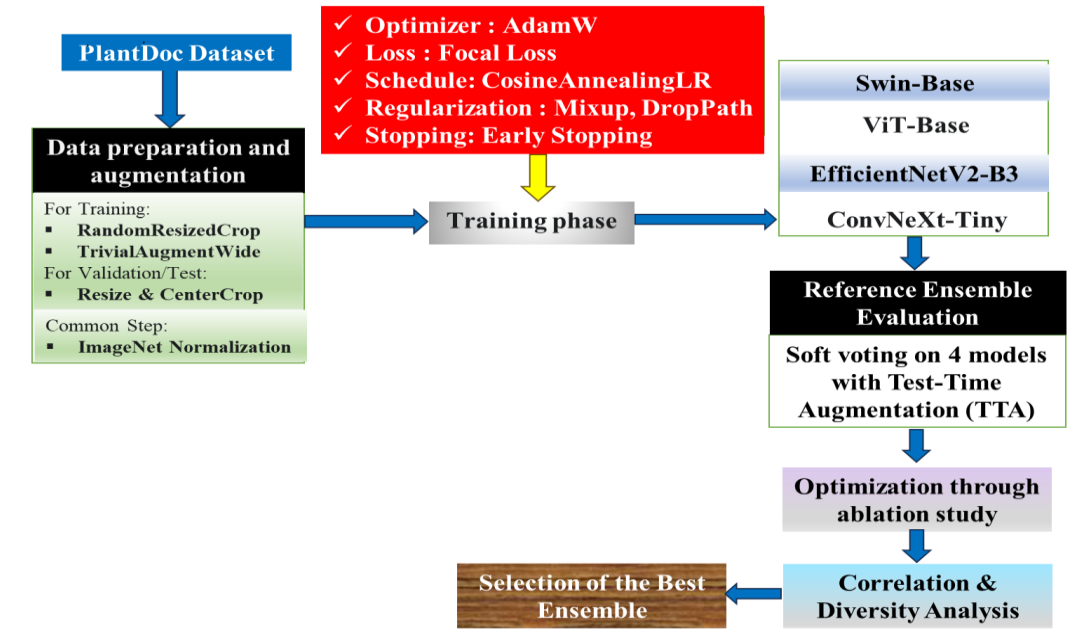


Figure 1: Methodological pipeline for ensemble construction and optimization

3.1 Dataset

Our study is conducted on the public PlantDoc dataset [29], which contains leaf images of 13 different plant species, divided into 27 classes that include both healthy leaves and various disease classifications—see Figure 2 for the distribution and composition of the dataset.

The majority of the images in the PlantDoc dataset were captured in the field (in-the-wild) using smartphones. Unlike other datasets that use simple backgrounds, these images were taken in a natural environment with complex backgrounds, under variable lighting conditions, and from different angles [29], which presents a more realistic challenge for detection models. Image samples illustrate this diversity—see also Figure 3 for image examples.

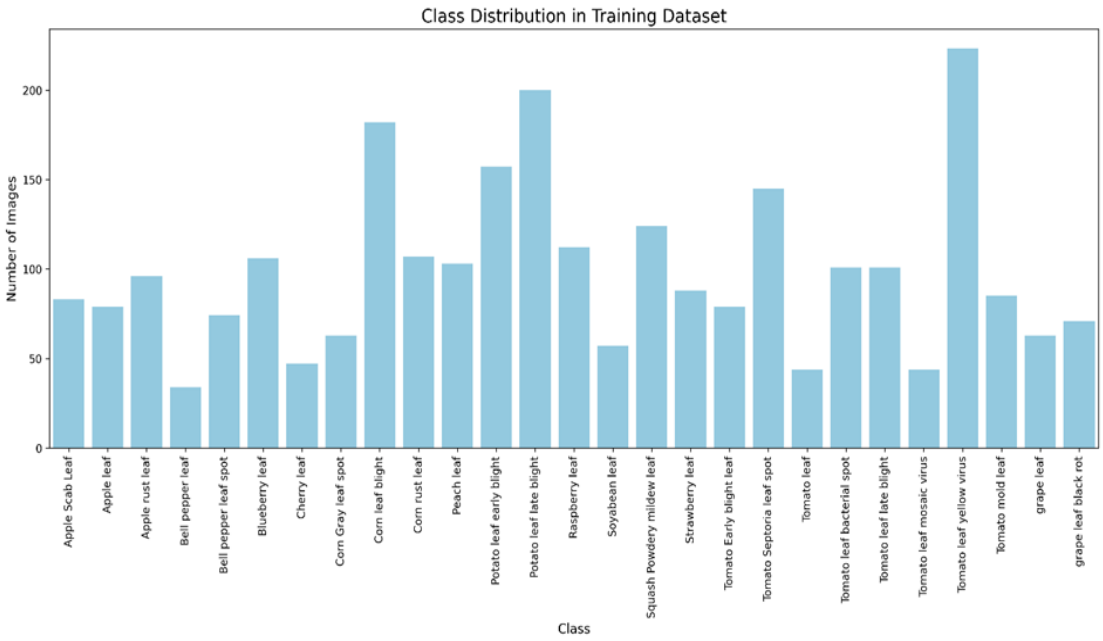
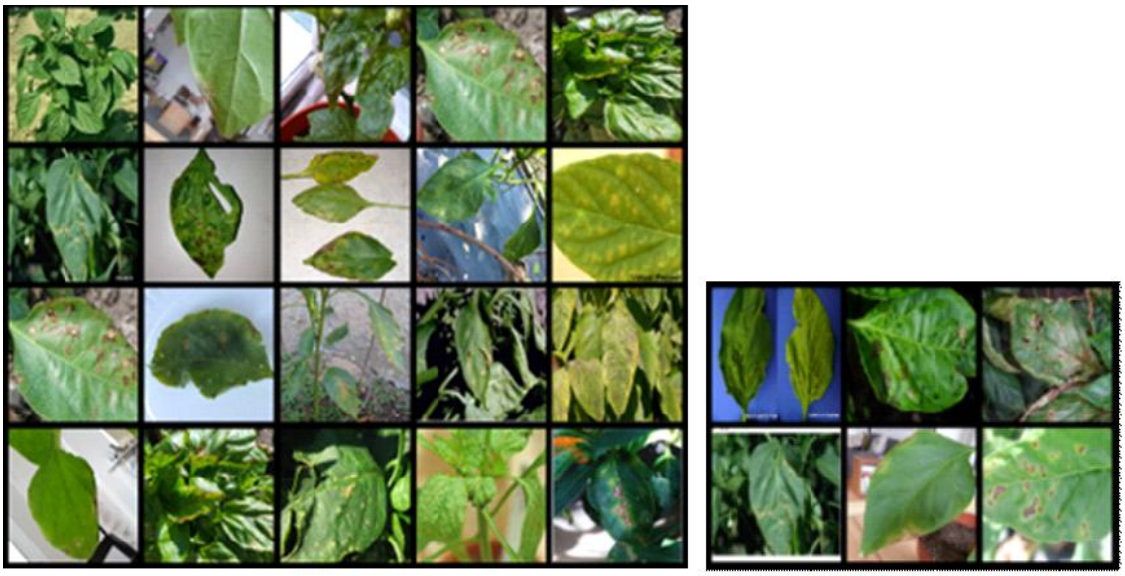


Figure 2: Class distribution in the PlantDoc training set



(a) Train subset of Bell pepper leaf spot class.

(b) Test subset of Bell pepper leaf spot class.

Figure 3: Sample images of PlantDoc Bell pepper leaf spot class.

The dataset is pre-divided into a training set and a test set. In accordance with the official split, we use the test set for our final evaluation. Upon verification of the publicly available data archive, we identified 252 usable image files, which we used for all our experiments to ensure reproducibility. For the training of each individual model, we subdivided the provided training set, allocating 85% of the images to a new training set and the remaining 15% to a validation set. To ensure a balanced representation of classes, this division was performed using stratification. The exact per-class counts for each split are detailed in Appendix A (Table A1). To ensure full auditability, the complete list of filenames used in the test set is provided in Appendix B (Table B1).

3.2 Data preprocessing and augmentation

Our data pipeline distinguishes between the training and evaluation phases to maximize robustness and ensure consistent testing.

For training, an aggressive data augmentation strategy was employed to improve the models' robustness. The transformations applied to the training images include a RandomResizedCrop to 224x224 pixels and the application of TrivialAugmentWide, an automatic augmentation policy that combines multiple operations.

For the validation and test sets, deterministic preprocessing was used to ensure consistent evaluation. The images were first resized so that their shorter side was 224 pixels while maintaining the aspect ratio, and then a 224x224 CenterCrop was applied. All images, after transformation, were converted to PyTorch tensors and normalized using the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225], which are standard for models pre-trained on ImageNet.

Furthermore, to ensure a robust evaluation and mitigate sensitivity to object centering, all final performance metrics on the test set were obtained using Test-Time Augmentation (TTA). For each test image, we averaged the predictions from the original image and its horizontally flipped version.

3.3 Architectures and individual model training

To ensure a rich and informative comparison, we selected four state-of-the-art models, summarized in Table1, chosen specifically to represent diverse architectural families. This selection includes:

- Two prominent Transformer-based architectures: The Swin Transformer (Swin-Base) and the standard Vision Transformer (ViT-Base).
- Two modern CNN-based architectures: A recent version of EfficientNet (EfficientNetV2-B3), known for its improved training efficiency and performance, and a modern CNN (ConvNeXt-Tiny) that incorporates architectural insights from Transformers.

This strategic selection enables a rigorous analysis of predictive synergies and conflicts both within and between architectural families. To establish a fair and equitable comparison, all models were initialized with weights pre-trained on the large-scale ImageNet-21k dataset (or the comparable ImageNet-22k). The exact timm model identifiers used for each architecture are detailed in Table 1 to ensure full reproducibility.

Table 1: Model architectures used in the study

Model	Timm Identifier	Type	Description
Swin-Base	swin_base_patch4_window7_224.ms_in22k	Transformer	Hierarchical Transformer using shifted attention windows.
ViT-Base	vit_base_patch16_224.augreg_in21k	Transformer	Standard Vision Transformer architecture.
EfficientNetV2-B3	tf_efficientnetv2_b3.in21k	CNN	Modern convolutional network with fused-MBConv blocks.
ConvNeXt-Tiny	Convnext tiny.fb_in22k	CNN	Modern convolutional network inspired by Transformers.

Subsequently, each model was fine-tuned independently on the PlantDoc training set following a rigorous and unified training protocol. To ensure the robustness of our findings and to account for the variance inherent in the training process, each individual model training was conducted three times using distinct random seeds (42, 50, and 72). The mean and standard deviation of these runs are reported in Section 4.1 to demonstrate the stability of our models. All subsequent analyses in Section 4 (including the ensemble studies and benchmarks) are then performed on the single, most representative set of models (from seed 42) to ensure clarity and consistency. This multi-seed validation approach ensures that our main conclusions are drawn from a stable and representative baseline, rather than from a single, potentially fortunate outcome.

The training protocol was designed to maximize the performance of each model before their integration into the ensemble, and its key components are as follows:

- **Optimizer:** We used the AdamW optimizer with an initial learning rate of 5e-5 and a weight decay of 0.05.
- **Loss function**: To effectively address the potential class imbalance in the PlantDoc dataset, we opted for the Focal Loss with parameters alpha=0.25 and gamma=2.0
- **Learning rate scheduler:** A CosineAnnealingLR scheduler was implemented, preceded by a linear warmup phase over the first 5 epochs to stabilize initial training.
- **Regularization:** In addition to data augmentation, we employed the Mixup technique (alpha=0.5) during training to improve the model's generalization by creating virtual examples through linear interpolation of images and their labels. DropPath regularization (rate=0.2) was also enabled for architectures that support it.
- **Stopping conditions and model selection**: Training was conducted for a maximum of 60 epochs. An Early Stopping mechanism was implemented to prevent overfitting: if the validation loss did not improve for 10 consecutive epochs, training was stopped. For the final evaluation and inclusion in the ensemble, we kept the model weights from the epoch that achieved the lowest validation loss. This ensures that each model is evaluated at its peak performance.

The chosen hyperparameters are based on established best practices for fine-tuning ImageNet-pretrained Transformer-based and modern CNN models. The learning rate of 5e-5 is a common and effective starting point for the AdamW optimizer, while the Focal Loss parameters (alpha=0.25, gamma=2.0) correspond to the widely adopted default values established by Lin et al. [30], known for their robustness in handling class imbalance.

This rigorous and unified training protocol was systematically applied to each of the four base architectures. The use of an Early Stopping mechanism based on validation performance ensures that every model was trained to its peak performance under identical conditions, guaranteeing a fair comparison under identical optimization constraints. Consequently, the contribution of each model in the subsequent ensemble study is a function of its intrinsic capabilities and complementarity, rather than its training duration.

3.4 Ensemble method: soft voting

To aggregate the predictions from our models, we employ Soft Voting, a classic and effective ensemble method [16]. The process consists of two main steps.

First, for a given input image, each individual model I within the ensemble E produces a vector of raw prediction scores, known as logits (z_i). This logit vector is then converted into a probability vector p_i using the Softmax function, ensuring that the outputs represent a valid probability distribution over the C classes. The probability for the class j from the model i is calculated as follows:

$$p_{i,j} = \frac{e^{z_{i,j}}}{\sum_{k=1}^{C} e^{z_{i,k}}} \quad (1)$$

Where:

$\boldsymbol{p_{i,j}}$ is the probability assigned by model i to class j.

$\boldsymbol{z_{i,j}}$ is the logit (raw score) from model i for class j.

C is the total number of classes.

k is an index used to iterate over all classes for the normalization sum.

Second, the final probability vector for the ensemble, P_{ens}, is obtained by computing the element-wise average of the individual probability vectors from all models in the ensemble:

$$P_{ens} = \frac{1}{|E|} \Sigma_{i \in E} p_i \quad (2)$$

Where:
P_{ens} is the final averaged probability vector from the ensemble.
E is the set of all models in the ensemble.
|E| is the total number of models in the ensemble.
p_i is the probability vector from model i.

Finally, the class predicted by the ensemble, ŷ, is determined by identifying the class with the highest averaged probability. This is achieved using the argmax function:

$$\hat{y} = \text{argmax\_j } (P_{ens,\, j}) \quad (3)$$

Where:
ŷ is the final predicted class label.
argmax_j returns the class index j that maximizes the value of the ensemble probability for that class.

3.5 Ensemble optimization protocol

To systematically identify the optimal ensemble composition, we implemented a multi-stage analysis protocol. Our approach adapts the principle of ablation studies—a standard method for assessing component importance—to evaluate the marginal contribution of each model within the ensemble [31]. The protocol proceeds as follows:

1. **Full ensemble reference:** First, we construct a reference ensemble (E_full) comprising the four fully optimized trained models (Swin-Base, ViT-Base, EfficientNetV2-B3, ConvNeXt-Tiny). Predictions are aggregated using Soft Voting (Section 3.4), and its performance on the test set is established as our benchmark for the ablation study.
2. **Systematic "Leave-One-Out" ablation**: Building on this reference ensemble, we then perform a systematic ablation. For each model m in the full ensemble, we create a subset (E_full \ {m}) and evaluate its performance. This process allows us to empirically identify the best-performing subset and to quantify the marginal contribution of each model.
3. **Correlation analysis for a deeper explanation**: To understand and explain the results of the ablation study, we quantify the predictive redundancy between the optimized base models. We calculate a Pearson correlation matrix based on the final class predictions generated by each model pair. We intentionally perform this analysis on the official test set. Since the performance of the ablated ensembles is evaluated on this same set, this choice creates a direct and interpretable link between the observed performance changes (the effect) and the predictive behavior of the models on the exact data where those changes occurred (the cause). This provides a clear and methodologically sound explanation for the observed synergy or redundancy.

3.6 Evaluation of alternative aggregation strategies

While our primary optimization method focuses on selecting the best architectural composition of the ensemble (as detailed in Section 3.5), a common alternative is to optimize the aggregation mechanism itself. To provide a robust benchmark, we implemented and evaluated two distinct families of aggregation-focused strategies on the full 4-model ensemble: Soft Voting and Stacking.

3.6.1 Soft voting strategies

As modern deep learning models are frequently miscalibrated (i.e., overconfident), our analysis of soft voting goes beyond a simple average. We systematically evaluated four strategies to assess the impact of probability calibration and advanced weighting schemes. This experiment, conducted on the full four-model ensemble (from seed 42), evaluates and compares the following strategies:

- **Uncalibrated soft voting**: This strategy corresponds strictly to the "Full Ensemble Reference" (4 models) established in our ablation study (Section 3.5). It computes the simple average of the raw probability outputs from each individual optimized model. Its performance serves as the reference point for this comparative analysis.
- **Calibrated soft voting:** In this strategy, each model's probabilities are first calibrated using Temperature Scaling on the validation set. The final prediction is the simple average of these calibrated probabilities.
- **Calibrated Weighted (F1):** This approach applies a weighted average to the calibrated probabilities. The weights are derived from each model's macro F1-score on the validation set.
- **Calibrated Weighted (Logloss):** This strategy also uses a weighted average of calibrated probabilities. To ensure numerical stability and to regularize the weights, they are derived from the inverse of each model's Log-Loss on the validation set, with a small regularization constant ε (epsilon) added to the denominator (weight = 1 / (log_loss + ε)).

3.6.2 Stacking

A more complex, two-level learning approach is implemented. A meta-model (XGBoost) is trained to learn how to best combine the predictions generated by the four optimized base models. To prevent data leakage and ensure a robust evaluation, we employed a rigorous protocol defined by four key components:

1. **Feature generation:** The training features for the meta-learner were generated using a 5-fold stratified cross-validation strategy on the full training set. This created "out-of-fold" predictions, ensuring that the features used to train the meta-learner were generated by base models that had not seen that data during their training within that fold. This strict separation of training and validation flows effectively addresses the overfitting issues observed in simpler stacking implementations.
2. **Input features:** The features provided to the meta-learner were the full probability vectors (outputs of the softmax function) generated by the models trained within the cross-validation loop described in the Feature Generation stage. This resulted in a feature vector of size 108 (4 models × 27 classes) for each training sample.
3. **Base models configuration:** The base models integrated into this stacking framework utilize the optimized hyperparameter configurations established in Section 3.3. To prevent data leakage, a strict distinction was applied regarding model training:
   - **For meta-learner training (Step 1):** While the hyperparameters (e.g., learning rate, batch size, loss function) remained fixed to the optimal values determined in Section 3.3, fresh instances of the base models were re-trained from scratch within each fold of the cross-validation process described in Step 1. This ensures that the features fed to the meta-learner were unbiased (Out-Of-Fold).
   - **For final test inference:** Once the meta-learner was fully tuned, we utilized the distinct, optimized models (corresponding to the representative Seed 42 run detailed in Section 4.1) to generate the final predictions on the test set.
4. **Hyperparameter tuning and final training:** Finally, the hyperparameters of the XGBoost model were themselves optimized using a nested, inner-loop 3-fold cross-validation with GridSearchCV. The optimal configuration identified during this search was then fitted on the complete set of "out-of-fold" predictions to create the final meta-learner.

3.7 Evaluation metrics

To provide a comprehensive evaluation of our models, we used a set of standard classification metrics [32] for multi-class problems. We report three key metrics: Accuracy, Balanced Accuracy (BA), and the Macro F1-score. These metrics are derived from the standard components of a confusion matrix: True Positives (TP), False Positives (FP), and False Negatives (FN).

Accuracy: This metric measures the overall fraction of correctly classified instances and is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

In our multi-class context, per-class metrics are calculated by treating each class c individually (the "one-vs-rest" approach). From these per-class scores, we derive the following aggregated metrics:

Balanced Accuracy (BA): This metric is the average of recall obtained on each class. It is particularly suited for imbalanced datasets as it gives equal weight to each class, regardless of its number of samples. It is defined as:

$$BA = \frac{1}{N} \Sigma_{i=1}^{N} \text{Recall}_i \quad (5)$$

Macro F1-Score: This metric is the unweighted average of the F1-scores calculated for each individual class. By treating all classes equally, it provides a robust measure of a model's ability to perform well across both majority and minority classes. The F1-score for a single class is the harmonic mean of Precision and Recall:

$$\text{F1-Score}_i = 2.\frac{\text{Precision}_i \,.\, \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (6)$$

The Macro F1-Score is then calculated as:

$$\text{Macro F1-Score} = \frac{1}{N} \Sigma_{i=1}^{N} \text{F1-Score}_i \quad (7)$$

where for each class i, N is the total number of classes, and:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (8)$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (9)$$

To ensure a thorough and transparent evaluation, we leverage all three metrics. We use Accuracy for a clear measure of overall correctness and for comparison with the state of the art, while the Macro F1-score and Balanced Accuracy provide crucial insights into the model's fairness and performance on rare classes. This comprehensive approach best reflects our goal of developing a robust and effective model.

3.8 Experimental setup

The experiments were conducted on Python 3.10 using the PyTorch framework (v2.5.1) and the timm library (v1.0.16). The evaluation was performed on an NVIDIA RTX 4060Ti GPU with a batch size of 16, utilizing CUDA v12.1. Performance metrics were calculated using the scikit-learn library (v1.7.2) on the official PlantDoc test set. The reproducibility of our results was ensured by setting a fixed random seed for all relevant libraries in each experiment. The specific seeds used are detailed in the corresponding results sections (e.g., Section 4.1).

4 Results

This section presents the results of our experiments, structured to systematically identify and analyze the optimal model ensemble. First, we establish the individual performance of each state-of-the-art architecture (Section 4.1). Second, we identify the best-performing ensemble configuration through a two-part analysis (Section 4.2): a "leave-one-out" ablation study to find the optimal model subset, followed by a benchmark against alternative aggregation strategies. Third, to explain the ablation results, we conduct an analysis of synergy and redundancy using correlation and diversity metrics (Section 4.3). Finally, we perform an in-depth, multi-level error analysis of the best-performing ensemble to understand its strengths and weaknesses (Section 4.4).

Across all stages of this evaluation, performance is measured using a comprehensive set of metrics-Accuracy, Balanced Accuracy, and Macro F1-score-whose complementary roles and definitions are detailed in Section 3.7.

4.1 Individual model performance

The evaluation of individual model performance is the first step in our analysis. To rigorously assess performance and stability, and to account for the variance inherent in the training process, each architecture in our optimized protocol (Section 3.3) was trained and evaluated three times using distinct random seeds (42, 50, and 72). The detailed per-seed results are presented in Table 2. To synthesize these findings and facilitate a clearer comparison of overall performance and stability, the mean and standard deviation for each metric were calculated and are presented in Table 3.

Analysis of the multi-seed results reveals a clear performance hierarchy. The summary in Table 3 shows that the Swin Transformer (Swin-Base) emerges as the top-performing model across all average metrics, achieving a mean accuracy of 73.81% and the highest mean F1-score (macro) of 72.72%. The ConvNeXt-Tiny follows as a very strong competitor. In contrast, while the Vision Transformer (ViT-Base) shows the highest stability with the lowest standard deviations (e.g., ±0.0040 for accuracy), its average performance is lower than the other two main architectures. As detailed in Table 2, the absolute peak F1-score of the study (0.7418) was achieved by ConvNeXt-Tiny with seed 50, while the peak accuracy (0.7460) was achieved by Swin-Base with seed 42.

For the subsequent in-depth analyses, including the ensemble study, it was necessary to select a single, representative set of models from our multi-seed runs. Our choice fell on the models generated with seed 42, as a rigorous analysis showed this seed to be the most representative of the models' typical behavior (i.e., its results had the lowest average distance to the mean performance across all metrics). While this choice was driven by objectivity, it is worth noting that this seed also produced the study's single highest accuracy score.

Table 2: Detailed performance per model and random seed.

Model	Seed	Accuracy	Balanced Accuracy	Macro F1-score
Swin-Base	42	0.7460	0.7434	0.7412
	50	0.7341	0.7350	0.7293
	72	0.7341	0.7242	0.7110
ViT-Base	42	0.7063	0.7073	0.7009
	50	0.6984	0.6959	0.6844
	72	0.7024	0.7036	0.6909
EfficientNetV2-B3	42	0.6032	0.5913	0.5854
	50	0.5635	0.5691	0.5488
	72	0.6151	0.6160	0.6007
ConvNeXt-Tiny	42	0.7183	0.7102	0.7062
	50	0.7302	0.7389	0.7418
	72	0.7103	0.7069	0.7014

Table 3: Mean performance and stability of optimized models (over 3 seeds).

Model	Accuracy (Mean ± Std. Dev.)	Balanced Accuracy (Mean ± Std. Dev.)	Macro F1-score (Mean ± Std. Dev.)
Swin-Base	0.7381 ± 0.0069	0.7342 ± 0.0096	0.7272 ± 0.0152
ViT-Base	0.7024 ± 0.0040	0.7023 ± 0.0058	0.6921 ± 0.0084
EfficientNetV2-B3	0.5939 ± 0.0270	0.5922 ± 0.0235	0.5783 ± 0.0267
ConvNeXt-Tiny	0.7196 ± 0.0100	0.7186 ± 0.0176	0.7165 ± 0.0221

Having established the robustness of our models and selected a representative configuration, we then quantified the effectiveness of our training protocol (described in Section 3.3). To do so, Table 4 compares the performance of the "Optimized" models against a "Baseline" configuration. To ensure a fair and direct comparison, both configurations were run using the same representative random seed (seed 42). The Baseline

configuration uses a minimal setup (e.g., Cross-Entropy loss, fixed learning rate).

Table 4: Performance gain of the optimized protocol (seed 42) vs. baseline.

Model	Accuracy	Balanced Accuracy	Macro F1-score
Swin-Base (Baseline)	0.7222	0.7325	0.7182
Swin-Base (Optimized)	0.7460 (+3.3%)	0.7434 (+1.5%)	0.7412 (+3.2%)
ViT-Base (Baseline)	0.6389	0.6403	0.6342
ViT-Base (Optimized)	0.7063 (+10.5%)	0.7073 (+10.5%)	0.7009 (+10.5%)
ConvNeXt-Tiny (Baseline)	0.7063	0.7053	0.6933
ConvNeXt-Tiny (Optimized)	0.7183 (+1.7%)	0.7102 (+0.7%)	0.7062 (+1.9%)
EfficientNetV2-B3 (Baseline)	0.5714	0.5820	0.5653
EfficientNetV2-B3 (Optimized)	0.6032 (+5.6%)	0.5913 (+1.6%)	0.5854 (+3.6%)

Analysis of Table 4 reveals two major and complementary conclusions.

First, the application of our optimized protocol leads to a systematic and significant performance improvement. The gain on the macro F1-score is particularly notable, with an increase of +10.5% for ViT-Base and +3.2% for the top-performing Swin-Base. This result convincingly validates the effectiveness of our approach, which combines a Focal Loss function to address class imbalance, an advanced learning rate scheduler (CosineAnnealingLR with warmup), and aggressive regularization techniques (Mixup, TrivialAugmentWide, and DropPath) that effectively combat overfitting and improve generalization.

Second, the comparison highlights a clear and consistent performance hierarchy among the architectures. The Swin Transformer establishes itself as the top-performing model under both protocols, setting a strong benchmark. Conversely, EfficientNetV2-B3 consistently displays the most modest performance. This performance hierarchy, observed under identical protocol conditions, suggests that modern architectures capable of modeling global context or incorporating Transformer-like design principles (such as Swin and ConvNeXt) are intrinsically well-suited to the complexities of the PlantDoc dataset, compared to standard CNNs. These optimized individual performances will serve as a solid foundation for the next step: our ensemble construction and ablation analysis.

4.2 Ensemble optimization: composition vs. aggregation strategy

To identify the optimal ensemble configuration, we conducted a two-part comparative analysis. First, we performed a "leave-one-out" ablation study to find the best subset of models by optimizing the ensemble's composition. Second, we evaluated alternative strategies that optimize the aggregation mechanism itself as a benchmark.

4.2.1 Optimizing ensemble composition via ablation study

The results of our "leave-one-out" ablation study, presented in Table 5, demonstrate a key principle of ensemble learning: adding more models does not always improve performance. On the contrary, our analysis reveals that a three-model subset performs better than the full ensemble.

Table 5: Performance of the ablation study ensembles (soft voting)

Ensemble Configuration	Accuracy	Balanced Accuracy	Macro F1-score
ABLATION STUDY (Soft Voting)			
Ensemble (full, 4 models)	0.7500	0.7433	0.7409
Ensemble (without ViT-Base)	0.7579	0.7543	0.7500
Ensemble (without EfficientNetV2-B3)	0.7540	0.7464	0.7411
Ensemble (without ConvNeXt-Tiny)	**0.7619**	**0.7542**	**0.7503**
Ensemble (without Swin-Base)	0.7222	0.7150	0.7076

The primary finding is that the ensemble without the ConvNeXt-Tiny model achieved the highest scores across all primary metrics, with an accuracy of 0.7619 and a macro F1-score of 0.7503. This configuration outperforms the full four-model ensemble, which peaks at an accuracy of 0.7500 and an F1-score of 0.7409. This improvement suggests that the predictions from the ConvNeXt-Tiny model, while strong individually, introduced a degree of redundancy or predictive noise that limited the collective performance. This redundancy hypothesis will be further investigated in Section 4.3 through the analysis of the prediction correlation matrix.

This ablation study also allowed us to quantify the importance of each architecture. The critical contribution of the Swin-Base is highlighted by the drastic performance drop (F1-score to 0.7076) observed upon its removal, identifying it as the cornerstone of our ensemble.

While the superiority of the three-model ensemble is numerically clear, a formal confirmation of its statistical significance would require a larger test set. Nevertheless, these results provide strong empirical evidence that the careful selection of a diverse and complementary subset of models is a more effective optimization strategy than simply aggregating all available models. For a detailed, per-class performance breakdown of the best-performing ensemble (without ConvNeXt-Tiny) and the full four-model ensemble, refer to the full classification reports in Appendix A (Table A2 and Table A3, respectively).

4.2.2 Benchmarking against alternative aggregation strategies

To further validate our selection-based optimization approach, we benchmarked the performance of our best ablation ensemble (Ensemble without ConvNeXt-Tiny) against advanced aggregation strategies applied to the full four-model ensemble. The goal was to determine whether optimizing the aggregation mechanism could match or exceed the performance of optimizing the ensemble's composition. Two families of strategies were tested: advanced soft voting methods (including calibration and weighting) and a two-level Stacking ensemble with an optimized XGBoost meta-learner. The aggregated results are summarized in Table 6, while the detailed, per-class classification reports for each of these benchmarked strategies can be found in Appendix A (Tables A4-A7).

Table 6: Performance benchmark of alternative aggregation strategies on the full ensemble.

Ensemble Strategy	Accuracy	Balanced Accuracy	Macro F1-score
Ensemble (without ConvNeXt-Tiny)	0.7619	0.7542	0.7503
AGGREGATION STRATEGIES (Full Ensemble)			
Uncalibrated Soft Voting	0.7500	0.7433	0.7409
Calibrated Soft Voting	0.7460	0.7402	0.7387
Calibrated Weighted Voting (F1)	0.7500	0.7433	0.7418
Calibrated Weighted Voting (LogLoss)	0.7540	0.7476	0.7481
Stacking (Optimized XGBoost)	0.7222	0.7148	0.7201

As shown in Table 6, while the probability calibration and weighted voting strategies yielded results comparable to the Full Ensemble Reference (Uncalibrated Soft Voting), the complex Stacking approach (implemented via the rigorous leakage-safe protocol described in Section 3.6.2) actually led to a performance degradation (Accuracy: 0.7222, Macro F1: 0.7201). Our pruned three-model ensemble (Ensemble without ConvNeXt-Tiny) achieves the highest scores across all primary metrics, with an accuracy of 0.7619 and a macro F1-score of 0.7503.

This finding is significant. It suggests that when base models exhibit a degree of predictive redundancy (as will be demonstrated in Section 4.3 with the correlation matrix), the performance gain from physically removing a less synergistic model (like ConvNeXt-Tiny) is greater than the gain from applying a more sophisticated weighting or learning scheme that attempts to mitigate its influence. This limitation aligns with the theoretical understanding of meta-learning: stacking algorithms typically require significantly larger validation sets to effectively model complex inter-classifier correlations without overfitting. In scenarios with moderate dataset sizes (as is common in plant disease classification), simpler voting strategies often prove more resilient and generalize better than high-variance meta-learners.

Ultimately, this benchmark demonstrates that the path to an optimal ensemble in our case lies not in more complex aggregation logic, but in the strategic curation of its components. The fact that a smaller, carefully selected ensemble outperforms both a larger one and more complex, advanced aggregation schemes reinforces our fundamental finding.

4.3 Analysis of ensemble synergy and redundancy

The superior performance of the ensemble after removing the ConvNeXt-Tiny model (as shown in Section 4.2) suggests a phenomenon of predictive redundancy. To investigate this, we analyzed the synergy between the optimized base models. This analysis is twofold: a visual assessment via a prediction correlation matrix and a quantitative evaluation using pairwise diversity metrics.

First, we calculated the Pearson correlation between the final class predictions of each of the four architectures on the test set. The resulting correlation matrix is presented in Figure 4.

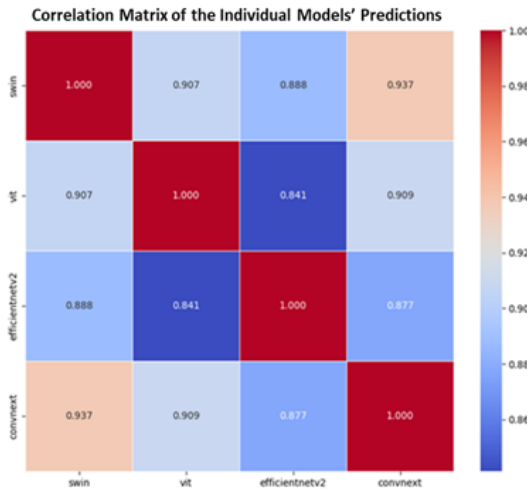


Figure 4: Pearson correlation matrix of model predictions. high values (warm colors) indicate that two models tend to make similar predictions.

The matrix provides crucial insights into the models' predictive behaviors. We observe a very high correlation of 0.937 between Swin-Base and ConvNeXt-Tiny. This strong similarity is likely rooted in their architectural design; ConvNeXt-Tiny was explicitly designed by adapting modern CNNs to incorporate principles from Vision Transformers like Swin-Base, making it plausible that both models learn similar hierarchical feature representations. Conversely, EfficientNetV2-B3 and ViT-Base exhibit the lowest correlation (0.841), highlighting them as the most diverse pair.

To quantify these relationships more precisely, we computed pairwise diversity metrics, including the Disagreement Measure and the Double-Fault Measure. The Disagreement score represents the percentage of test samples on which two models differ, while the Double-Fault score measures the percentage of samples that both models misclassify. The results are summarized in Table7.

Table 7: Pairwise diversity metrics.

Model Pair	Disagreement	Double-Fault
Swin-Base vs Vit-Base	0.1905	0.1786
Swin-Base vs EfficientNetV2-B3	0.1984	0.2262
Swin-Base vs ConvNeXt-Tiny	0.1468	0.1944
Vit-Base vs EfficientNetV2-B3	0.2063	0.2421
Vit-Base vs ConvNeXt-Tiny	0.1786	0.1984
EfficientNetV2-B3 vs ConvNeXt-Tiny	0.2262	0.2262

These metrics, when considered alongside the individual performance results (Table 4), provide a clear justification for our ablation study findings. They reveal a classic ensemble trade-off.

First, the Swin-Base and ConvNeXt-Tiny pair, despite being the two top-performing architectures individually across all evaluated metrics., exhibit the lowest Disagreement score (0.1468). This confirms that despite being individually the strongest models, their predictive similarity limits the marginal benefit of including both in a simple averaging ensemble. Crucially, the second lowest disagreement is observed between ViT-Base and ConvNeXt-Tiny (0.1786). This pattern indicates a systemic redundancy: ConvNeXt-Tiny strongly correlates with both Transformer-based models. Furthermore, it is notable that the disagreement between the two Transformers themselves (Swin vs. ViT, 0.1905) is higher than the disagreement between ConvNeXt-Tiny and either Transformer. This confirms that ConvNeXt-Tiny acts as a redundant architectural intermediary, whereas the Swin-ViT pair preserves greater predictive diversity.

Second, the Double-Fault metric clarifies the synergy within the retained models and further validates the exclusion of ConvNeXt-Tiny. Notably, the Swin-Base vs. ViT-Base pair achieves the lowest Double-Fault score (0.1786) across the entire matrix. This is lower than the Swin-ConvNeXt pair (0.1944), indicating that Swin-Base and ViT-Base are less likely to err simultaneously than Swin-Base and ConvNeXt-Tiny. This makes the Swin-ViT pair the most robust core for the ensemble.

Regarding EfficientNetV2-B3, while its pairs show higher double-fault rates (reflecting its lower individual accuracy on difficult samples), this is counterbalanced by its high disagreement scores with the retained models (e.g., 0.2063 against ViT-Base). This confirms its role as a specialist model that provides diversity, whereas the Swin-ViT pair ensures stability

This detailed analysis explains why removing ConvNeXt-Tiny was the most effective optimization. The optimal ensemble composition (Swin-Base, ViT-Base, EfficientNetV2-B3) is successful because it retains the strongest model (Swin-Base), pairs it with the model least likely to share its errors (ViT-Base), and adds the most diverse model (EfficientNetV2-B3) for a unique corrective perspective. This process eliminated the model most redundant with our core performer, thereby maximizing the overall diversity and error-correction capability of the final ensemble.

Ultimately, this analysis confirms that a principled approach is essential for optimizing ensembles. In our case, the path to the optimal ensemble involved pruning the model (ConvNeXt-Tiny) that offered the least diversity relative to our strongest model (Swin-Base), thereby maximizing the trade-off between high individual performance and collective predictive synergy.

4.4 In-depth analysis of the optimal model's behavior

To understand the performance of our best-performing model, the Ensemble (without Convnext-Tiny), we conducted a multi-level error analysis. This involved a quantitative assessment using the confusion matrix and per-species metrics, as well as a qualitative analysis of failure cases using Grad-CAM to validate our hypotheses.

4.4.1 Quantitative error analysis

The confusion matrix for our best-performing model (Ensemble (without Convnext)) is presented in Figure 5. It enables a finer-grained analysis of the classification errors.

Analysis of the matrix reveals several key points:

1. Well-identified classes: The model demonstrates excellent performance for a significant number of classes. It achieves a perfect recall of 100% for 6 out of 27 classes: Corn_rust_leaf, Raspberry_leaf, Squash_Powdery_mildew_leaf, Strawberry_leaf, grape_leaf and grape_leaf_black_rot. Furthermore, it shows strong discriminative power for other important classes such as Tomato_leaf_yellow_virus (93.3% recall), Tomato_Septoria_leaf_spot (91.7% recall), Apple_Scab_Leaf (90.0% recall), and Apple_rust_leaf (90.0% recall), indicating that their visual features are distinct and well-learned by the ensemble.

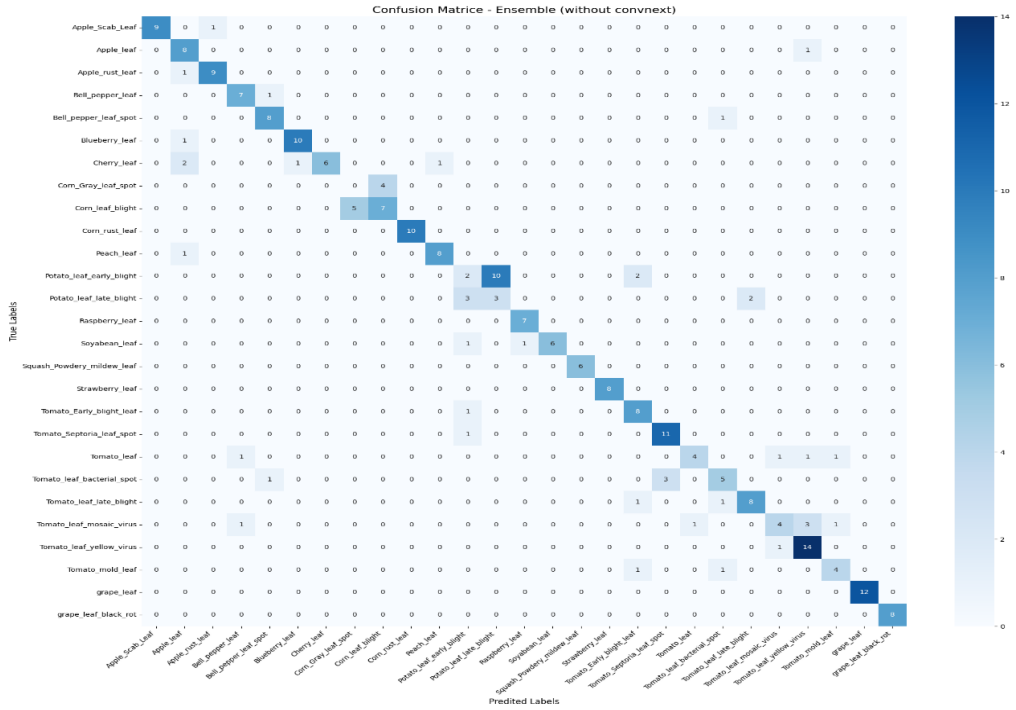


Figure 5: Confusion matrix for the optimal ensemble (without convnext).

2. Main areas of confusion: The matrix highlights significant areas of confusion, which are primarily concentrated within species that have multiple, visually similar diseases.

- **Critical failure on corn_gray_leaf_spot**: The most striking finding is the model's complete failure to identify Corn_Gray_leaf_spot, which has a recall and F1-score of 0.0000. This indicates that all four samples of this class were misclassified, likely as the visually similar Corn_leaf_blight, revealing a critical point of confusion for corn diseases.
- **Severe confusion within potato diseases**: The second major area of weakness lies in distinguishing between potato diseases. The Potato_leaf_early_blight class performs very poorly, with a recall of only 14.3%, and this is compounded by the poor performance on Potato_leaf_late_blight (37.5% recall). The confusion between these two notoriously similar diseases is a primary source of error for the model, leading to a bidirectional misclassification, as they are very difficult to distinguish visually at certain stages.
- **Confusion within tomato diseases**: Several tomato diseases also present a significant challenge. Tomato_leaf_mosaic_virus is frequently misidentified, resulting in a low recall of 40.0%; it is most often mistaken for the visually similar Tomato_leaf_yellow_virus. Similarly, Tomato_leaf_bacterial_spot shows a modest recall score of 55.6%, as it is frequently confused with Tomato_leaf_late_blight. The Tomato_leaf (healthy) class also has a low recall of 50.0%, indicating frequent confusion with other tomato-related classes.

The detailed per-class performance for this optimized ensemble, including precision, recall, and F1-score for each of the 27 classes, is available in the full Classification Report in Appendix (Table A2).

Summary of the analysis: There is a clear pattern where the model excels at identifying unique, visually distinct classes but struggles significantly when faced with multiple diseases on the same plant species that share similar symptoms (e.g., spots, blights, or viral effects). The complete failure on Corn_Gray_leaf_spot and the severe confusion between potato blights are the most critical weaknesses.

4.4.2 Qualitative validation of failure cases

To concretely illustrate these areas of confusion and validate our central hypothesis—that the errors stem from intrinsic visual ambiguity—we performed a qualitative analysis of key failure cases. Figure 6 presents two representative examples of misclassified images, supplemented with their predicted probability distributions and Grad-CAM saliency maps. These maps highlight the image regions the model used to make its prediction.

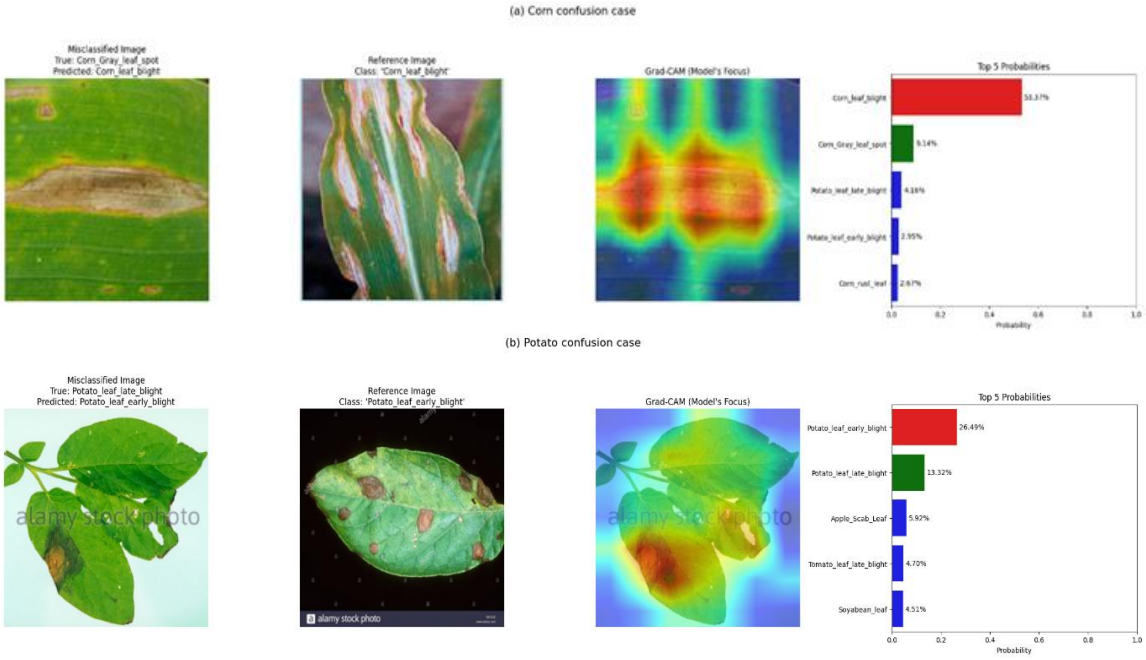


Figure 6: Qualitative analysis of classification errors. For each example, we show (1) the misclassified image, (2) a reference image of the predicted class, (3) the Grad-CAM saliency map showing the model's focus, and (4) the top predicted probabilities.

Examination of the visual examples in Figure 6 confirms and deepens the findings from the confusion matrix:

- **Ambiguity in corn diseases** (Figure 6a): This example shows a Corn_Gray_leaf_spot image misclassified as Corn_leaf_blight. The Grad-CAM map is particularly revealing: it shows that the model correctly focuses its attention on the large, necrotic lesion, confirming it has learned to identify relevant pathological features. The similarity of the necrotic lesions between the two corn diseases makes the model's confusion understandable, even for a human observer. Furthermore, the probability distribution shows only a moderate confidence of 53.37% for the wrong class, with the true class being the second most likely hypothesis. This demonstrates that while the model identifies the correct visual cues, the features of the lesion itself (shape, color, texture) are not sufficiently distinct for unambiguous classification.
- **Challenge of potato blights** (Figure 6b): The confusion between Potato_leaf_late_blight (True) and Potato_leaf_early_blight (Predicted) is notoriously difficult. More significantly, this example shows that distinguishing between early and late potato blight remains highly challenging, even on sharp images. The Grad-CAM map again indicates that the model is correctly analyzing the lesion area on the leaf. The very low prediction confidence (26.49%) and the high probability assigned to the true class (13.32%, the second highest) reveal the model's profound uncertainty.

These cases demonstrate that the model's errors are less due to learning defects than to the intrinsic complexity and ambiguity of the visual diagnosis task

4.4.3 Species-level performance evaluation

While fine-grained disease classification is challenging, another clinically relevant metric is the model's ability to correctly identify the plant species, irrespective of its health status. To assess this, we grouped the classes by species (e.g., all "Tomato" classes into a single "Tomato" category) and re-evaluated the optimized ensemble's performance. The results, presented in Table 8, show a dramatic improvement in performance.

The model achieves a macro-averaged F1-score of 0.9164 at the species level, a significant increase from the 0.7503 achieved for the fine-grained disease classification. This high level of performance is consistent across most species. Notably, the model achieves a perfect F1-score of 1.0000 for Corn, Squash, Strawberry, and Grape, indicating flawless species identification for these categories. Other major species like Apple (0.9180 F1-score) and Tomato (0.9308 F1-score) also demonstrate very high performance. The only notable exception is Cherry, which shows a lower recall (60%). Closer inspection reveals that these errors are primarily due to confusion with the related Peach species, likely owing to their visual similarity.

Overall, the high accuracy of 0.9246 at the species level confirms that the model robustly learns the general visual characteristics of each plant.

Table 8: Species-level classification report

	Precision	Recall	F1-score	Support
Apple	0.8750	0.9655	0.9180	29
Bell	0.8421	0.9412	0.8889	17
Blueberry	0.9091	0.9091	0.9091	11
Cherry	1.0000	0.6000	0.7500	10
Corn	1.0000	1.0000	1.0000	26
Peach	0.8889	0.8889	0.8889	9
Potato	0.8571	0.8182	0.8372	22
Raspberry	0.8750	1.0000	0.9333	7
Soyabean	1.0000	0.7500	0.8571	8
Squash	1.0000	1.0000	1.0000	6
Strawberry	1.0000	1.0000	1.0000	8
Tomato	0.9250	0.9367	0.9308	79
grape	1.0000	1.0000	1.0000	20
accuracy			0.9246	252
macro avg	0.9363	0.9084	0.9164	252
weighted avg	0.9276	0.9246	0.9229	252

Synthesis of the error analysis

Our multi-level error analysis leads to a unified conclusion: the model's errors are not random but are highly systematic and stem from the intrinsic difficulty of the task. The quantitative results show that confusions are concentrated on pairs of classes where visual distinction is intrinsically difficult, even for the human eye—for example, between different types of "blight," "spot," or "virus" on the same plant. The qualitative validation using Grad-CAM confirms this, demonstrating that the model correctly focuses on pathological regions but is hampered by ambiguous visual cues. Finally, the excellent performance at the species level reinforces the conclusion that the model has successfully learned high-level, species-specific features. The primary challenge, therefore, lies not in species identification, but in the fine-grained visual diagnosis of pathologies with similar manifestations. Improving discrimination in these complex cases constitutes a priority avenue for future work, potentially by integrating contextual information or using more targeted attention techniques.

5 Discussion

To contextualize our results within the existing research, we compare them to other studies that have also used the PlantDoc dataset. It is important to note that a direct, one-to-one comparison is challenging, as these studies may employ different data splits, preprocessing techniques, or evaluation protocols. This comparison, summarized in Table 9 (reporting Accuracy, as it is the only metric consistently available across these studies), should therefore be viewed as an indication of relative performance rather than a formal benchmark.

With this caveat in mind, our approach demonstrates highly competitive results. Our best individual model, the Swin-Base, achieves an accuracy of 0.7460. The primary finding, however, is that our optimized ensemble, obtained through a systematic ablation study, reaches a Macro F1-score of 0.7503 and an accuracy of 0.7619. This result compares favorably to previously published ensemble methods on this dataset, such as the stacking approach of R. Ramaprasad & S. Raman (0.7357) and the Snapshot Ensemble approach of Puangsuwan & Surinta (0.6951). Crucially, this strong performance is not achieved by developing a more complex aggregation mechanism; rather, it stems from a systematic and principled selection of a diverse set of base models. Our work demonstrates that the careful curation of an ensemble's composition—particularly by combining complementary CNNs and Transformers and pruning for predictive redundancy—is a more critical factor for success than the aggregation method itself.

Table 9: Comparison of results with other studies on the PlantDoc dataset.

Study	Method	Dataset	Accuracy
This study	Swin-Base	PlantDoc	0.7460
This study (Ensemble)	Ensemble (Swin-Base + ViT-Base + EfficientNetV2-B3)	PlantDoc	0.7619
R. Ramaprasad & S. Raman [33]	Stacking Ensemble (SEMFD-Net)	PlantDoc	0.7357
Menon et al. [34]	MobileNet-V2	PlantDoc	0.6980
Puangsuwan & Surinta [35]	Snapshot Ensemble (DenseNet201)	PlantDoc	0.6951
Chandra et al. [9]	SVM with siamese network	PlantDoc	0.6276
Moupoujou et al. [36]	MobileNet	PlantDoc	0.6014

Beyond a simple comparison of scores, our main contribution lies in the ensemble construction methodology. Existing literature has primarily explored two avenues for improving ensemble performance:

- Simple Voting Aggregation: Works such as those by Chaudhary et al. [15] or Mathew et al. [13] have shown that majority voting improved accuracy. Our study builds upon this by demonstrating the limitation of such approaches: the simple accumulation of models, even when aggregated with a more nuanced soft voting method, can be suboptimal if they are predictively redundant.

- Optimizing the Aggregation Strategy: To address the limitations of simple voting, a common approach is to use more advanced aggregation methods. However, our benchmark of such strategies in Section 4.2.2 provides a compelling counter-narrative. We found that various weighted voting schemes, even after probability calibration, offered only marginal gains or no improvement at all over

simple averaging. More importantly, even a sophisticated, two-level learning method like Stacking actually led to a performance degradation (Macro F1: 0.7201, Accuracy: 0.7222), significantly underperforming our pruned three-model ensemble (Macro F1: 0.7503, Accuracy: 0.7619). This empirically demonstrates a crucial principle: when predictive redundancy is the limiting factor, it is more effective to physically remove a redundant model (via ablation) than to attempt to mitigate its influence with complex aggregation logic or a meta-learner, which often struggle with data scarcity in such tasks. The strategic selection of an ensemble's composition is therefore the dominant optimization step.

Our work thus builds upon the foundational work of Mohanty et al. and Too et al., [3], [4] which established the superiority of Deep Learning on controlled, lab-condition datasets like PlantVillage. We extend their findings in two critical ways: first, by tackling the more challenging 'in-the-wild' conditions of the PlantDoc dataset, and second, by going beyond individual model performance to analyze their complex interactions within an ensemble—a crucial issue with the emergence of modern Transformer architectures.

Practical implications

Beyond the academic performance metrics, our findings offer significant practical implications for the development and deployment of real-world plant disease detection systems. The central conclusion—that a smaller, strategically curated ensemble can outperform a larger, more computationally expensive one—provides a direct guideline for practitioners.

For developers of agricultural technology, this means that instead of defaulting to the largest possible ensemble of models, a more resource-efficient and effective approach is to conduct a preliminary ablation study. By identifying and removing redundant or underperforming models, practitioners can achieve better overall performance while simultaneously reducing the computational overhead (memory, processing time) required for inference. This is particularly crucial for applications designed to run on edge devices with limited resources, such as smartphones, drones, or in-field sensors.

Consequently, our methodology translates into a tangible pathway for creating diagnostic tools that are not only more accurate but also faster and more accessible, thereby facilitating earlier and more widespread disease detection in agricultural practices.

Limitations and future work

Although our results establish a highly competitive performance and a robust methodology, it is important to acknowledge the limitations of this study to guide future work.

The primary limitation, is that our analysis, while diverse in terms of architectures, remains entirely based on visual features. Consequently, the errors of our best ensemble are concentrated on diseases with nearly identical symptoms, where visual diagnosis alone reaches its intrinsic limits. A second limitation is that, although PlantDoc contains real-world images, validating the robustness of our ensemble on broader data distributions—from different geographical regions or cultivation conditions—remains a necessary step for large-scale application.

These limitations open clear and promising avenues for future research, which can be categorized into three main directions.

The first direction involves exploring more advanced classification and ensemble strategies within the visual domain. Our current optimization was based on a systematic 'leave-one-out' ablation; an interesting alternative would be to compare this with constructive heuristics, such as a greedy forward selection algorithm. Additionally, building upon our finding that species identification is robust, a hierarchical classification approach (first identify the species, then the disease) could allow for smaller, specialized models to better tackle the subtle intra-species confusions.

The second direction, which addresses the intrinsic limits of purely visual diagnosis highlighted by our analysis, lies in the development of multimodal models. This is arguably the most promising path. Integrating structured metadata—such as crop variety, growth stage, or environmental sensor data (e.g., humidity, temperature)—could provide the critical discriminant context that is missing from the images alone. This would shift the paradigm from simple visual recognition to a more holistic, data-driven diagnostic system.

Finally, the third direction directly addresses the challenge of real-world generalization. Future work should focus on Transfer Learning and Domain Adaptation techniques. These methods would ensure that the performance of our optimized ensemble is maintained when faced with data from new field conditions without requiring extensive re-labeling, thereby guaranteeing the model's practical robustness.

6 Conclusion

In this study, we addressed the challenge of accurate plant disease detection by proposing a rigorous methodology for the optimization of Deep Learning ensembles. Through an advanced training protocol, systematically applied across multiple random seeds, we first established the robust individual performances of four state-of-the-art architectures.

Our central finding is that a pruned, three-model ensemble, identified through a systematic ablation study, achieves a highly competitive Macro F1-score of 0.7503 and an accuracy of 0.7619 on the complex PlantDoc dataset. This result surpasses the performance of the full four-model ensemble, a phenomenon we explain through correlation and diversity analysis, which revealed significant predictive redundancy. Our work empirically demonstrates that the naive aggregation of models is a suboptimal strategy. We have shown that carefully selecting an ensemble's composition to maximize predictive complementarity is a more effective

optimization lever than employing more complex aggregation methods like weighted voting or stacking, which proved less effective than our pruning approach.

While our results establish a robust performance benchmark, they also open clear avenues for future research, primarily through the integration of multimodal data to resolve the most ambiguous visual cases. In summary, our work positions the methodological pairing of ablation study and correlation analysis as an essential and pragmatic tool for designing robust and high-performing model ensembles in the field of precision agriculture.

References

[1] Padol, P. B. and Yadav, A. A. (2016). SVM classifier based grape leaf disease detection. Conference on Advances in Signal Processing (CASP), Pune, India, 2016, pp. 175-179. https://doi.org/10.1109/CASP.2016.7746160

[2] Vaishnnave, M. P., Devi, K. S., Srinivasan, P. and Jothi, G. A. P. (2019). Detection and Classification of Groundnut Leaf Diseases using KNN classifier, IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-5. https://doi.org/10.1109/ICSCAN.2019.8878733

[3] Mohanty, S. P., Hughes, D. P. and Salathe, M. (2016). Using deep learning for image-based plant disease detection. Front. Plant Sci. https://doi.org/10.3389/fpls.2016.01419

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

[5] Hasan, R.I.; Yusuf, S.M.; Alzubaidi, L. (2020). Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. Plants, 9, 1302. https://doi.org/10.3390/plants9101302

[6] Shoaib, M., Shah, B., Ei-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., Ali, F. (2023). An advanced deep learning models-based plant disease detection. A review of recent research. Front. Plant Sci., 14, 1158933. https://doi.org/10.3389/fpls.2023.1158933

[7] Akbar, M., Ullah, M., Shah, B., Khan, R.U., Hussain, T., Ali, F., Alenezi, F., Syed, I., Kwak, K.S. (2022). An effective deep learning approach for the classification of Bacteriosis in peach leave. Front. Plant Sci., 13, 1064854. https://doi.org/10.3389/fpls.2022.1064854

[8] Zhao, Y., Sun, C., Xu, X., and Chen, J. (2022). RIC-net: A plant disease classification model based on the fusion of inception and residual structure and embedded attention mechanism, C*omputers and Electronics in Agriculture* 193: 106644. https://doi.org/10.1016/j.compag.2021.106644

[9] Chandra, M., Redkar, S., Roy, S., Patil, P. (2020). Classification of Various Plant Diseases Using Deep Siamese Network. Available online: https://www.researchgate.net/publication/341322315 (accessed on 18 December 2025).

[10] Iftikhar, M., Kandhro, I. A., Kausar, N., Kehar A., Uddin, M. and Dandoush A. (2024). Plant disease management: A fine-tuned enhanced CNN approach with mobile app integration for early detection and classification, *Artif. Intell. Rev.*, vol. 57, no. 7, pp. 1-29. https://doi.org/10.1007/s10462-024-10809-z

[11] Ganaie, M., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. Eng. Appl. Artif. Intelligence, 115, 105151. https://doi.org/10.1016/j.engappai.2022.105151

[12] Li, H., Jin, Y., Zhong, J., and Zhao, R. (2021). A fruit tree disease diagnosis model based on stacking ensemble learning. Complexity, 2021, 6868592. https://doi.org/10.1155/2021/6868592

[13] Mathew, A., Antony, A., Mahadeshwar, Y., Khan, T., and Kulkarni, A. (2022). Plant disease detection using GLCM feature extractor and voting classification approach. Mat. Today, Proc. 58, Part 1, pp.407–415. https://doi.org/10.1016/j.matpr.2022.02.350

[14] Palanisamy, S., and Sanjana, N. (2023). Corn leaf disease detection using genetic algorithm and weighted voting, Proceedings of the 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India. (ICAECA). pp. 1-6. https://doi.org/10.1109/ICAECA56562.2023.10200196

[15] Chaudhary, A., Thakur, R., Kolhe, S., and Kamal, R. (2020). A particle swarm optimization-based ensemble for vegetable crop disease recognition. Comput. Electron. Agricult., 178, 105747. https://doi.org/10.1016/j.compag.2020.105747

[16] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249. https://doi.org/10.1002/widm.1249

[17] Giacinto, G., and Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition, 34(9), pp.1879-1881. https://doi.org/10.1016/S0031-3203(00)00150-3

[18] Tsoumakas, G., Katakis, I. (2007). Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM), 3(3), pp. 1-13. https://doi.org/10.4018/jdwm.2007070101

[19] Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A. (2004). Ensemble selection from libraries of models. Proceedings of the 21st International Conference on Machine Learning (ICML '04). https://doi.org/10.1145/1015330.1015432

[20] Astani, M., Hasheminejad, M. and Vaghefi, M. (2022). A diverse ensemble classifier for tomato disease recognition, *Comput. Electron. Agricult.*, vol. 198, Art. no. 107054. https://doi.org/10.1016/j.compag.2022.107054

[21] Shafik, W., Tufail, A. De Silva Liyanage, C. and R. Apong, A. A. H. M. (2024). Using transfer learning-

based plant disease classification and detection for sustainable agriculture, *BMC Plant Biol.*, vol. 24, no. 1, p. 136. https://doi.org/10.1186/s12870-024-04825-y

[22] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Proceedings of the International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2010.11929

[23] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV48922.2021.00986

[24] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). A ConvNet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). .https://doi.org/10.48550/arXiv.2201.03545

[25] Chen, J., Zhang, D. Zeb, A. and Nanehkaran, Y. A. (2021). Identification of Rice plant diseases using lightweight attention networks, *Expert Syst. Appl.*, vol. 169, Art. no. 114514. https://doi.org/10.1016/j.eswa.2020.114514

[26] Ouamane, A., Chouchane, A., Himeur, Y., Miniaoui, S., Atalla, S., Mansoor, W. (2025). Optimized Vision Transformers for Superior Plant Disease Detection, *IEEE Access*, vol. 13, pp. 48552-48570. https://doi.org/10.1109/ACCESS.2025.3547416

[27] Singh Thakur, P., Khanna, P., Sheorey, T., and Ojha, A. (2022). Explainable vision transformer enabled convolutional neural network for plant disease identification: Plantxvit, *arXiv:2207.07919*. https://doi.org/10.48550/arXiv.2207.07919

[28] Tabbakh, A. and Barpanda, S. S. (2023). A deep features extraction model based on the transfer learning model and vision transformer 'TLMViT' for plant, *IEEE Access*, vol. 11, pp. 45377-45392. https://doi.org/10.1109/ACCESS.2023.3273317

[29] Singh, D., Padgett, E., & T. S. G., A. K. (2020). PlantDoc: A Dataset for Visual Plant Disease Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). https://doi.org/10.48550/arXiv.1911.10317

[30] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2980-2988. https://doi.org/10.48550/arXiv.1708.02002

[31] Kasneci, G. and Kasneci, E. (2024). Enriching Tabular Data with Contextual LLM Embeddings: A Comprehensive Ablation Study for Ensemble Classifiers, arXiv preprint arXiv:2411.01645. https://doi.org/10.48550/arXiv.2411.01645

[32] Grandini M., Bagli E., Visani G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756. https://doi.org/10.48550/arXiv.2008.05756

[33] Ramaprasad, R., Raman, S. (2022). SEMFD-Net: A Stacked Ensemble for Multiple Foliar Disease Classification. Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD) pp. 241-245. https://doi.org/10.1145/3493700.3493719

[34] Menon, V.; Ashwin, V.; Deepa, R.K. (2021). Plant disease detection using CNN and transfer learning. In Proceedings of the International Conference on Communication, Control and Information Sciences (ICCISc), Idukki, India, pp. 16–18; IEEE: Piscataway, NJ, USA, 2021; Volume 1, pp. 1–6. https://doi.org/10.1109/ICCISc52257.2021.9484957

[35] Puangsuwan T.; Surinta, O. (2021). Enhancement of plant leaf disease classification based on snapshot ensemble convolutional neural network. ICIC Exp Lett., 15(6), pp. 669–680. https://doi.org/10.24507/icicel.15.06.669

[36] Moupojou, E., Tagne, A., Retraint, F., Tadonkemwa, A., Dongmo, W., Tapamo, H., Nkenlifack, M. (2023). FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning. IEEE Access, 11, pp. 35398–35410. https://doi.org/10.1109/ACCESS.2023.3263042

Appendix A

Table A1: Class-wise Data Split (Stratification)

	Classe	**Total (Original Train)**	**Entraînement (85%)**	**Validation (15%)**	**Test (Officiel)**
0	Apple_Scab_Leaf	83	71	12	10
1	Apple_leaf	79	67	12	9
2	Apple_rust_leaf	96	82	14	10
3	Bell_pepper_leaf	34	29	5	8
4	Bell_pepper_leaf_spot	74	63	11	9
5	Blueberry_leaf	106	90	16	11
6	Cherry_leaf	47	40	7	10
7	Corn_Gray_leaf_spot	63	54	9	4
8	Corn_leaf_blight	182	155	27	12
9	Corn_rust_leaf	107	91	16	10
10	Peach_leaf	103	88	15	9
11	Potato_leaf_early_blight	157	133	24	14
12	Potato_leaf_late_blight	200	170	30	8
13	Raspberry_leaf	112	95	17	7
14	Soyabean_leaf	57	48	9	8
15	Squash_Powdery_mildew_leaf	124	105	19	6
16	Strawberry_leaf	88	75	13	8
17	Tomato_Early_blight_leaf	79	67	12	9
18	Tomato_Septoria_leaf_spot	145	123	22	12
19	Tomato_leaf	44	37	7	8
20	Tomato_leaf_bacterial_spot	101	86	15	9
21	Tomato_leaf_late_blight	101	86	15	10
22	Tomato_leaf_mosaic_virus	44	37	7	10
23	Tomato_leaf_yellow_virus	223	189	34	15
24	Tomato_mold_leaf	85	72	13	6
25	grape_leaf	63	54	9	12
26	grape_leaf_black_rot	71	60	11	8
	TOTAL	2668	2267	401	252

Table A2: Classification Report for the Ensemble without Convnext

Class	Precision	Recall	F1-Score	Support
Apple_Scab_Leaf	1.0000	0.9000	0.9474	10
Apple_leaf	0.6154	0.8889	0.7273	9
Apple_rust_leaf	0.9000	0.9000	0.9000	10
Bell_pepper_leaf	0.7778	0.8750	0.8235	8
Bell_pepper_leaf_spot	0.8000	0.8889	0.8421	9
Blueberry_leaf	0.9091	0.9091	0.9091	11
Cherry_leaf	1.0000	0.6000	0.7500	10
Corn_Gray_leaf_spot	0.0000	0.0000	0.0000	4
Corn_leaf_blight	0.6364	0.5833	0.6087	12
Corn_rust_leaf	1.0000	1.0000	1.0000	10
Peach_leaf	0.8889	0.8889	0.8889	9
Potato_leaf_early_blight	0.2500	0.1429	0.1818	14
Potato_leaf_late_blight	0.2308	0.3750	0.2857	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.6667	0.8889	0.7619	9
Tomato_Septoria_leaf_spot	0.7857	0.9167	0.8462	12
Tomato_leaf	0.8000	0.5000	0.6154	8
Tomato_leaf_bacterial_spot	0.6250	0.5556	0.5882	9
Tomato_leaf_late_blight	0.8000	0.8000	0.8000	10
Tomato_leaf_mosaic_virus	0.6667	0.4000	0.5000	10
Tomato_leaf_yellow_virus	0.7368	0.9333	0.8235	15
Tomato_mold_leaf	0.6667	0.6667	0.6667	6
grape_leaf	1.0000	1.0000	1.0000	12
grape_leaf_black_rot	1.0000	1.0000	1.0000	8
accuracy			0.7619	252
macro avg	0.7641	0.7542	0.7503	252
weighted avg	0.7708	0.7619	0.7571	252

Table A3: Classification Report for the Full Ensemble (Uncalibrated Soft Voting)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple_Scab_Leaf	0.8889	0.8000	0.8421	10
Apple_leaf	0.6667	0.8889	0.7619	9
Apple_rust_leaf	0.8182	0.9000	0.8571	10
Bell_pepper_leaf	0.7778	0.8750	0.8235	8
Bell_pepper_leaf_spot	0.7273	0.8889	0.8000	9
Blueberry_leaf	0.9091	0.9091	0.9091	11
Cherry_leaf	1.0000	0.7000	0.8235	10
Corn_Gray_leaf_spot	0.0000	0.0000	0.0000	4
Corn_leaf_blight	0.5455	0.5000	0.5217	12
Corn_rust_leaf	1.0000	0.9000	0.9474	10
Peach_leaf	1.0000	0.8889	0.9412	9
Potato_leaf_early_blight	0.2500	0.1429	0.1818	14
Potato_leaf_late_blight	0.2308	0.3750	0.2857	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.6154	0.8889	0.7273	9
Tomato_Septoria_leaf_spot	0.7333	0.9167	0.8148	12
Tomato_leaf	0.8000	0.5000	0.6154	8
Tomato_leaf_bacterial_spot	0.8000	0.4444	0.5714	9
Tomato_leaf_late_blight	0.8000	0.8000	0.8000	10
Tomato_leaf_mosaic_virus	0.6667	0.4000	0.5000	10
Tomato_leaf_yellow_virus	0.7368	0.9333	0.8235	15
Tomato_mold_leaf	0.6667	0.6667	0.6667	6
grape_leaf	1.0000	1.0000	1.0000	12
grape_leaf_black_rot	1.0000	1.0000	1.0000	8
accuracy			0.7500	252
macro avg	0.7596	0.7433	0.7409	252
weighted avg	0.7639	0.7500	0.7462	252

Table A4: Classification Report for the Full Ensemble (Calibrated Soft Voting)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple_Scab_Leaf	0.8889	0.8000	0.8421	10
Apple_leaf	0.6667	0.8889	0.7619	9
Apple_rust_leaf	0.8182	0.9000	0.8571	10
Bell_pepper_leaf	0.7778	0.8750	0.8235	8
Bell_pepper_leaf_spot	0.7273	0.8889	0.8000	9
Blueberry_leaf	0.9091	0.9091	0.9091	11
Cherry_leaf	1.0000	0.6000	0.7500	10
Corn_Gray_leaf_spot	0.0000	0.0000	0.0000	4
Corn_leaf_blight	0.5455	0.5000	0.5217	12
Corn_rust_leaf	1.0000	0.9000	0.9474	10
Peach_leaf	0.8889	0.8889	0.8889	9
Potato_leaf_early_blight	0.2500	0.1429	0.1818	14
Potato_leaf_late_blight	0.2308	0.3750	0.2857	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.6154	0.8889	0.7273	9
Tomato_Septoria_leaf_spot	0.7143	0.8333	0.7692	12
Tomato_leaf	0.8000	0.5000	0.6154	8
Tomato_leaf_bacterial_spot	0.6667	0.4444	0.5333	9
Tomato_leaf_late_blight	0.8000	0.8000	0.8000	10
Tomato_leaf_mosaic_virus	0.7143	0.5000	0.5882	10
Tomato_leaf_yellow_virus	0.7368	0.9333	0.8235	15
Tomato_mold_leaf	0.8000	0.6667	0.7273	6
grape_leaf	1.0000	1.0000	1.0000	12
grape_leaf_black_rot	1.0000	1.0000	1.0000	8
accuracy			0.7460	252
macro avg	0.7565	0.7402	0.7387	252
weighted avg	0.7594	0.7460	0.7428	252

Table A5: Classification Report for the full Ensemble (Calibrated Weighted (F1))

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple_Scab_Leaf	0.8889	0.8000	0.8421	10
Apple_leaf	0.6667	0.8889	0.7619	9
Apple_rust_leaf	0.8182	0.9000	0.8571	10
Bell_pepper_leaf	0.7778	0.8750	0.8235	8
Bell_pepper_leaf_spot	0.7273	0.8889	0.8000	9
Blueberry_leaf	0.9091	0.9091	0.9091	11
Cherry_leaf	1.0000	0.6000	0.7500	10
Corn_Gray_leaf_spot	0.0000	0.0000	0.0000	4
Corn_leaf_blight	0.5455	0.5000	0.5217	12
Corn_rust_leaf	1.0000	0.9000	0.9474	10
Peach_leaf	0.8889	0.8889	0.8889	9
Potato_leaf_early_blight	0.2500	0.1429	0.1818	14
Potato_leaf_late_blight	0.2308	0.3750	0.2857	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.6154	0.8889	0.7273	9
Tomato_Septoria_leaf_spot	0.7333	0.9167	0.8148	12
Tomato_leaf	0.8000	0.5000	0.6154	8
Tomato_leaf_bacterial_spot	0.8000	0.4444	0.5714	9
Tomato_leaf_late_blight	0.8000	0.8000	0.8000	10
Tomato_leaf_mosaic_virus	0.7143	0.5000	0.5882	10
Tomato_leaf_yellow_virus	0.7368	0.9333	0.8235	15
Tomato_mold_leaf	0.8000	0.6667	0.7273	6
grape_leaf	1.0000	1.0000	1.0000	12
grape_leaf_black_rot	1.0000	1.0000	1.0000	8
accuracy			0.7500	252
macro avg	0.7621	0.7433	0.7418	252
weighted avg	0.7650	0.7500	0.7464	252

Table A6: Classification Report for the full Ensemble (Calibrated Weighted (LogLoss))

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple_Scab_Leaf	0.8889	0.8000	0.8421	10
Apple_leaf	0.6667	0.8889	0.7619	9
Apple_rust_leaf	0.8182	0.9000	0.8571	10
Bell_pepper_leaf	0.7778	0.8750	0.8235	8
Bell_pepper_leaf_spot	0.7273	0.8889	0.8000	9
Blueberry_leaf	0.9091	0.9091	0.9091	11
Cherry_leaf	1.0000	0.7000	0.8235	10
Corn_Gray_leaf_spot	0.0000	0.0000	0.0000	4
Corn_leaf_blight	0.5455	0.5000	0.5217	12
Corn_rust_leaf	1.0000	0.9000	0.9474	10
Peach_leaf	1.0000	0.8889	0.9412	9
Potato_leaf_early_blight	0.2500	0.1429	0.1818	14
Potato_leaf_late_blight	0.2308	0.3750	0.2857	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.6154	0.8889	0.7273	9
Tomato_Septoria_leaf_spot	0.7143	0.8333	0.7692	12
Tomato_leaf	1.0000	0.5000	0.6667	8
Tomato_leaf_bacterial_spot	0.6667	0.4444	0.5333	9
Tomato_leaf_late_blight	0.8000	0.8000	0.8000	10
Tomato_leaf_mosaic_virus	0.7500	0.6000	0.6667	10
Tomato_leaf_yellow_virus	0.7368	0.9333	0.8235	15
Tomato_mold_leaf	0.8000	0.6667	0.7273	6
grape_leaf	1.0000	1.0000	1.0000	12
grape_leaf_black_rot	1.0000	1.0000	1.0000	8
accuracy			0.7540	252
macro avg	0.7693	0.7476	0.7481	252
weighted avg	0.7711	0.7540	0.7524	252

Table A7: Classification Report for the Full Ensemble Stacking (Optimized XGBoost)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple_Scab_Leaf	0.9000	0.9000	0.9000	10
Apple_leaf	0.6364	0.7778	0.7000	9
Apple_rust_leaf	0.9091	1.0000	0.9524	10
Bell_pepper_leaf	0.7000	0.8750	0.7778	8
Bell_pepper_leaf_spot	0.8571	0.6667	0.7500	9
Blueberry_leaf	0.9000	0.8182	0.8571	11
Cherry_leaf	1.0000	0.7000	0.8235	10
Corn_Gray_leaf_spot	0.2500	0.2500	0.2500	4
Corn_leaf_blight	0.6667	0.6667	0.6667	12
Corn_rust_leaf	0.9000	0.9000	0.9000	10
Peach_leaf	1.0000	0.8889	0.9412	9
Potato_leaf_early_blight	0.5000	0.2857	0.3636	14
Potato_leaf_late_blight	0.2727	0.3750	0.3158	8
Raspberry_leaf	0.8750	1.0000	0.9333	7
Soyabean_leaf	1.0000	0.7500	0.8571	8
Squash_Powdery_mildew_leaf	1.0000	1.0000	1.0000	6
Strawberry_leaf	1.0000	1.0000	1.0000	8
Tomato_Early_blight_leaf	0.3636	0.4444	0.4000	9
Tomato_Septoria_leaf_spot	0.3889	0.5833	0.4667	12
Tomato_leaf	1.0000	0.3750	0.5455	8
Tomato_leaf_bacterial_spot	0.2727	0.3333	0.3000	9
Tomato_leaf_late_blight	0.5833	0.7000	0.6364	10
Tomato_leaf_mosaic_virus	0.7778	0.7000	0.7368	10
Tomato_leaf_yellow_virus	0.8235	0.9333	0.8750	15
Tomato_mold_leaf	0.7500	0.5000	0.6000	6
grape_leaf	0.9231	1.0000	0.9600	12
grape_leaf_black_rot	1.0000	0.8750	0.9333	8
accuracy			0.7222	252
macro avg	0.7500	0.7148	0.7201	252
weighted avg	0.7511	0.7222	0.7248	252

Appendix B: Test Set Filenames

To ensure full auditability and reproducibility of our results, this appendix lists the complete set of the 252 image filenames that constitute the test set used in all our experiments. The file paths are relative to the main dataset directory.

Table B1: Complete list of test set filenames with index

#	File Path	#	File Path
1	Apple_Scab_Leaf/test_Apple Scab Leaf_1.jpg	127	Raspberry_leaf/test_Raspberry leaf_1.jpg
2	Apple_Scab_Leaf/test_Apple Scab Leaf_10.jpg	128	Raspberry_leaf/test_Raspberry leaf_2.jpg
3	Apple_Scab_Leaf/test_Apple Scab Leaf_2.jpg	129	Raspberry_leaf/test_Raspberry leaf_3.jpg
4	Apple_Scab_Leaf/test_Apple Scab Leaf_3.jpg	130	Raspberry_leaf/test_Raspberry leaf_4.jpg
5	Apple_Scab_Leaf/test_Apple Scab Leaf_4.jpg	131	Raspberry_leaf/test_Raspberry leaf_5.jpg
6	Apple_Scab_Leaf/test_Apple Scab Leaf_5.jpg	132	Raspberry_leaf/test_Raspberry leaf_6.jpg
7	Apple_Scab_Leaf/test_Apple Scab Leaf_6.jpg	133	Raspberry_leaf/test_Raspberry leaf_7.jpg
8	Apple_Scab_Leaf/test_Apple Scab Leaf_7.jpg	134	Soyabean_leaf/test_Soyabean leaf_1.jpg
9	Apple_Scab_Leaf/test_Apple Scab Leaf_8.jpg	135	Soyabean_leaf/test_Soyabean leaf_2.jpg
10	Apple_Scab_Leaf/test_Apple Scab Leaf_9.jpg	136	Soyabean_leaf/test_Soyabean leaf_3.jpg
11	Apple_leaf/test_Apple leaf_1.jpg	137	Soyabean_leaf/test_Soyabean leaf_4.jpg
12	Apple_leaf/test_Apple leaf_2.jpg	138	Soyabean_leaf/test_Soyabean leaf_5.jpg
13	Apple_leaf/test_Apple leaf_3.jpg	139	Soyabean_leaf/test_Soyabean leaf_6.jpg
14	Apple_leaf/test_Apple leaf_4.jpg	140	Soyabean_leaf/test_Soyabean leaf_7.jpg
15	Apple_leaf/test_Apple leaf_5.jpg	141	Soyabean_leaf/test_Soyabean leaf_8.jpg
16	Apple_leaf/test_Apple leaf_6.jpg	142	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_1.jpg
17	Apple_leaf/test_Apple leaf_7.jpg	143	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_2.jpg
18	Apple_leaf/test_Apple leaf_8.jpg	144	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_3.jpg
19	Apple_leaf/test_Apple leaf_9.jpg	145	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_4.jpg
20	Apple_rust_leaf/test_Apple rust leaf_1.jpg	146	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_5.jpg
21	Apple_rust_leaf/test_Apple rust leaf_10.jpg	147	Squash_Powdery_mildew_leaf/test_Squash Powdery mildew leaf_6.jpg
22	Apple_rust_leaf/test_Apple rust leaf_2.jpg	148	Strawberry_leaf/test_Strawberry leaf_1.jpg
23	Apple_rust_leaf/test_Apple rust leaf_3.jpg	149	Strawberry_leaf/test_Strawberry leaf_2.jpg
24	Apple_rust_leaf/test_Apple rust leaf_4.jpg	150	Strawberry_leaf/test_Strawberry leaf_3.jpg
25	Apple_rust_leaf/test_Apple rust leaf_5.jpg	151	Strawberry_leaf/test_Strawberry leaf_4.jpg
26	Apple_rust_leaf/test_Apple rust leaf_6.jpg	152	Strawberry_leaf/test_Strawberry leaf_5.jpg
27	Apple_rust_leaf/test_Apple rust leaf_7.jpg	153	Strawberry_leaf/test_Strawberry leaf_6.jpg
28	Apple_rust_leaf/test_Apple rust leaf_8.jpg	154	Strawberry_leaf/test_Strawberry leaf_7.jpg
29	Apple_rust_leaf/test_Apple rust leaf_9.jpg	155	Strawberry_leaf/test_Strawberry leaf_8.jpg
30	Bell_pepper_leaf/test_Bell_pepper leaf_1.jpg	156	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_1.jpg
31	Bell_pepper_leaf/test_Bell_pepper leaf_2.jpg	157	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_2.jpg
32	Bell_pepper_leaf/test_Bell_pepper leaf_3.jpg	158	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_3.jpg

#	**File Path**	#	**File Path**
33	Bell_pepper_leaf/test_Bell_pepper leaf_4.jpg	159	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_4.jpg
34	Bell_pepper_leaf/test_Bell_pepper leaf_5.jpg	160	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_5.jpg
35	Bell_pepper_leaf/test_Bell_pepper leaf_6.jpg	161	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_6.jpg
36	Bell_pepper_leaf/test_Bell_pepper leaf_7.jpg	162	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_7.jpg
37	Bell_pepper_leaf/test_Bell_pepper leaf_8.jpg	163	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_8.jpg
38	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_1.jpg	164	Tomato_Early_blight_leaf/test_Tomato Early blight leaf_9.jpg
39	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_2.jpg	165	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_10.jpg
40	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_3.jpg	166	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_11.jpg
41	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_4.jpg	167	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_1_1.jpg
42	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_5.jpg	168	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_1_2.jpg
43	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_6.jpg	169	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_2.jpg
44	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_7.jpg	170	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_3.jpg
45	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_8.jpg	171	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_4.jpg
46	Bell_pepper_leaf_spot/test_Bell_pepper leaf spot_9.jpg	172	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_5.jpg
47	Blueberry_leaf/test_Blueberry leaf_1.jpg	173	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_6.jpg
48	Blueberry_leaf/test_Blueberry leaf_10.jpg	174	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_7.jpg
49	Blueberry_leaf/test_Blueberry leaf_11.jpg	175	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_8.jpg
50	Blueberry_leaf/test_Blueberry leaf_2.jpg	176	Tomato_Septoria_leaf_spot/test_Tomato Septoria leaf spot_9.jpg
51	Blueberry_leaf/test_Blueberry leaf_3.jpg	177	Tomato_leaf/test_Tomato leaf_1.jpg
52	Blueberry_leaf/test_Blueberry leaf_4.jpg	178	Tomato_leaf/test_Tomato leaf_2.jpg
53	Blueberry_leaf/test_Blueberry leaf_5.jpg	179	Tomato_leaf/test_Tomato leaf_3.jpg
54	Blueberry_leaf/test_Blueberry leaf_6.jpg	180	Tomato_leaf/test_Tomato leaf_4.jpg
55	Blueberry_leaf/test_Blueberry leaf_7.jpg	181	Tomato_leaf/test_Tomato leaf_5.jpg
56	Blueberry_leaf/test_Blueberry leaf_8.jpg	182	Tomato_leaf/test_Tomato leaf_6.jpg
57	Blueberry_leaf/test_Blueberry leaf_9.jpg	183	Tomato_leaf/test_Tomato leaf_7.jpg
58	Cherry_leaf/test_Cherry leaf_1.jpg	184	Tomato_leaf/test_Tomato leaf_8.jpg
59	Cherry_leaf/test_Cherry leaf_10.jpg	185	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_1.jpg
60	Cherry_leaf/test_Cherry leaf_2.jpg	186	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_2.jpg
61	Cherry_leaf/test_Cherry leaf_3.jpg	187	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_3.jpg
62	Cherry_leaf/test_Cherry leaf_4.jpg	188	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_4.jpg
63	Cherry_leaf/test_Cherry leaf_5.jpg	189	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_5.jpg
64	Cherry_leaf/test_Cherry leaf_6.jpg	190	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_6.jpg

65	Cherry_leaf/test_Cherry leaf_7.jpg	191	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_7.jpg
#	**File Path**	#	**File Path**
66	Cherry_leaf/test_Cherry leaf_8.jpg	192	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_8.jpg
67	Cherry_leaf/test_Cherry leaf_9.jpg	193	Tomato_leaf_bacterial_spot/test_Tomato leaf bacterial spot_9.jpg
68	Corn_Gray_leaf_spot/test_Corn Gray leaf spot_1.jpg	194	Tomato_leaf_late_blight/test_Tomato leaf late blight_1.jpg
69	Corn_Gray_leaf_spot/test_Corn Gray leaf spot_2.jpg	195	Tomato_leaf_late_blight/test_Tomato leaf late blight_10.jpg
70	Corn_Gray_leaf_spot/test_Corn Gray leaf spot_3.jpg	196	Tomato_leaf_late_blight/test_Tomato leaf late blight_2.jpg
71	Corn_Gray_leaf_spot/test_Corn Gray leaf spot_4.jpg	197	Tomato_leaf_late_blight/test_Tomato leaf late blight_3.jpg
72	Corn_leaf_blight/test_Corn leaf blight_1.jpg	198	Tomato_leaf_late_blight/test_Tomato leaf late blight_4.jpg
73	Corn_leaf_blight/test_Corn leaf blight_10.jpg	199	Tomato_leaf_late_blight/test_Tomato leaf late blight_5.jpg
74	Corn_leaf_blight/test_Corn leaf blight_11.jpg	200	Tomato_leaf_late_blight/test_Tomato leaf late blight_6.jpg
75	Corn_leaf_blight/test_Corn leaf blight_12.jpg	201	Tomato_leaf_late_blight/test_Tomato leaf late blight_7.jpg
76	Corn_leaf_blight/test_Corn leaf blight_2.jpg	202	Tomato_leaf_late_blight/test_Tomato leaf late blight_8.jpg
77	Corn_leaf_blight/test_Corn leaf blight_3.jpg	203	Tomato_leaf_late_blight/test_Tomato leaf late blight_9.jpg
78	Corn_leaf_blight/test_Corn leaf blight_4.jpg	204	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_1.jpg
79	Corn_leaf_blight/test_Corn leaf blight_5.jpg	205	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_10.jpg
80	Corn_leaf_blight/test_Corn leaf blight_6.jpg	206	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_2.jpg
81	Corn_leaf_blight/test_Corn leaf blight_7.jpg	207	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_3.jpg
82	Corn_leaf_blight/test_Corn leaf blight_8.jpg	208	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_4.jpg
83	Corn_leaf_blight/test_Corn leaf blight_9.jpg	209	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_5.jpg
84	Corn_rust_leaf/test_Corn rust leaf_1.jpg	210	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_6.jpg
85	Corn_rust_leaf/test_Corn rust leaf_10.jpg	211	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_7.jpg
86	Corn_rust_leaf/test_Corn rust leaf_2.jpg	212	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_8.jpg
87	Corn_rust_leaf/test_Corn rust leaf_3.jpg	213	Tomato_leaf_mosaic_virus/test_Tomato leaf mosaic virus_9.jpg
88	Corn_rust_leaf/test_Corn rust leaf_4.jpg	214	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_1.jpg
89	Corn_rust_leaf/test_Corn rust leaf_5.jpg	215	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_1.jpg
90	Corn_rust_leaf/test_Corn rust leaf_6.jpg	216	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_2.jpg
91	Corn_rust_leaf/test_Corn rust leaf_7.jpg	217	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_3.jpg
92	Corn_rust_leaf/test_Corn rust leaf_8.jpg	218	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_4.jpg
93	Corn_rust_leaf/test_Corn rust leaf_9.jpg	219	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_5.jpg
94	Peach_leaf/test_Peach leaf_1.jpg	220	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_2_6.jpg
95	Peach_leaf/test_Peach leaf_2.jpg	121	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_3_1.jpg
#	**File Path**	#	**File Path**

96	Peach_leaf/test_Peach leaf_3.jpg	222	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_3_1.jpg
97	Peach_leaf/test_Peach leaf_4.jpg	223	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_4_1.jpg
98	Peach_leaf/test_Peach leaf_5.jpg	224	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_4_2.jpg
99	Peach_leaf/test_Peach leaf_6.jpg	225	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_5.jpg
100	Peach_leaf/test_Peach leaf_7.jpg	226	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_6_1.jpg
101	Peach_leaf/test_Peach leaf_8.jpg	227	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_6_2.jpg
102	Peach_leaf/test_Peach leaf_9.jpg	228	Tomato_leaf_yellow_virus/test_Tomato leaf yellow virus_6_3.jpg
103	Potato_leaf_early_blight/test_Potato leaf early blight_1.jpg	229	Tomato_mold_leaf/test_Tomato mold leaf_1.jpg
104	Potato_leaf_early_blight/test_Potato leaf early blight_2.jpg	230	Tomato_mold_leaf/test_Tomato mold leaf_2.jpg
105	Potato_leaf_early_blight/test_Potato leaf early blight_3.jpg	231	Tomato_mold_leaf/test_Tomato mold leaf_3.jpg
106	Potato_leaf_early_blight/test_Potato leaf early blight_4_1.jpg	232	Tomato_mold_leaf/test_Tomato mold leaf_4.jpg
107	Potato_leaf_early_blight/test_Potato leaf early blight_4_2.jpg	233	Tomato_mold_leaf/test_Tomato mold leaf_5.jpg
108	Potato_leaf_early_blight/test_Potato leaf early blight_4_3.jpg	234	Tomato_mold_leaf/test_Tomato mold leaf_6.jpg
109	Potato_leaf_early_blight/test_Potato leaf early blight_4_4.jpg	235	grape_leaf/test_grape leaf_1.jpg
110	Potato_leaf_early_blight/test_Potato leaf early blight_5.jpg	236	grape_leaf/test_grape leaf_10.jpg
111	Potato_leaf_early_blight/test_Potato leaf early blight_6_1.jpg	237	grape_leaf/test_grape leaf_11.jpg
112	Potato_leaf_early_blight/test_Potato leaf early blight_6_2.jpg	238	grape_leaf/test_grape leaf_12.jpg
113	Potato_leaf_early_blight/test_Potato leaf early blight_6_3.jpg	239	grape_leaf/test_grape leaf_2.jpg
114	Potato_leaf_early_blight/test_Potato leaf early blight_6_4.jpg	240	grape_leaf/test_grape leaf_3.jpg
115	Potato_leaf_early_blight/test_Potato leaf early blight_7.jpg	241	grape_leaf/test_grape leaf_4.jpg
116	Potato_leaf_early_blight/test_Potato leaf early blight_8.jpg	242	grape_leaf/test_grape leaf_5.jpg
117	Potato_leaf_late_blight/test_Potato leaf late blight_1.jpg	243	grape_leaf/test_grape leaf_6.jpg
118	Potato_leaf_late_blight/test_Potato leaf late blight_2.jpg	244	grape_leaf/test_grape leaf_7.jpg
119	Potato_leaf_late_blight/test_Potato leaf late blight_3.jpg	245	grape_leaf/test_grape leaf_8.jpg
120	Potato_leaf_late_blight/test_Potato leaf late blight_4.jpg	246	grape_leaf/test_grape leaf_9.jpg
121	Potato_leaf_late_blight/test_Potato leaf late blight_5.jpg	247	grape_leaf_black_rot/test_grape leaf black rot_1.jpg
122	Potato_leaf_late_blight/test_Potato leaf late blight_6.jpg	248	grape_leaf_black_rot/test_grape leaf black rot_2.jpg
123	Potato_leaf_late_blight/test_Potato leaf late blight_7.jpg	249	grape_leaf_black_rot/test_grape leaf black rot_3.jpg
124	Potato_leaf_late_blight/test_Potato leaf late blight_8.jpg	250	grape_leaf_black_rot/test_grape leaf black rot_4.jpg
125	Raspberry_leaf/test_Raspberry leaf_1.jpg	251	grape_leaf_black_rot/test_grape leaf black rot_5.jpg
126	Raspberry_leaf/test_Raspberry leaf_2.jpg	252	grape_leaf_black_rot/test_grape leaf black rot_6.jpg