Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2029~2039
e-ISSN: 2548-6861

Exploration of Machine Learning Algorithms and Class Imbalance Handling with Deep Feature Extraction Using ResNet50 for Plant Disease Detection

Ervin Aditya [1]*, Ajie Kusuma Wardhana [2]*
* Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta
ervinaditya24@students.amikom.ac.id[1], ajiekusuma@amikom.ac.id[2]

Article Info

Article history:

Received 2025-07-20
Revised 2025-08-25
Accepted 2025-09-10

Keyword:

Plant Disease Detection,
Machine Learning,
Under-Sampling,
Pre-Trained ResNet50,
CNN

ABSTRACT

Plant leaf diseases pose a significant threat to agricultural productivity, necessitating accurate and efficient identification systems for timely intervention. This study proposes an approach that leverages deep feature extraction using a pretrained ResNet50 model combined with machine learning algorithms to recognize 38 classes of plant leaves, including both healthy and diseased categories. Each image was transformed into a 2048-dimensional feature vector, followed by normalization and dimensionality reduction using Principal Component Analysis (PCA). To mitigate the issue of class imbalance in the dataset, random under-sampling was applied at the feature level to ensure equal representation across all classes. Eleven machine learning models were trained and evaluated using 5-fold cross-validation, with performance assessed through accuracy, precision, recall, F1-score, and ROC AUC score. Among the evaluated models, the Support Vector Machine (SVM) achieved the highest accuracy of 99.63%, followed by Logistic Regression at 97.33%, and LightGBM at 96.25%. These models demonstrated strong generalization capabilities in multiclass settings, while simpler classifiers like AdaBoost and Decision Tree yielded lower performance. A comparative analysis of training and test accuracy further highlighted model robustness and overfitting tendencies. The findings emphasize the potential of combining pretrained convolutional neural networks for feature extraction with conventional classifiers to address complex agricultural classification tasks. Future work may explore the inclusion of larger and more diverse samples, as well as the integration of the proposed method into accessible diagnostic platforms to support precision farming and enhance crop health monitoring.

I. INTRODUCTION

Plant diseases are one of the most significant threats to global food security, as they can cause substantial crop yield losses and substantial economic losses. It is estimated that plant diseases contribute to 20 – 40% of global crop yield losses each year, with an economic impact of around USD 220 billion [1]. In addition to the economic impact, plant diseases also exacerbate food crises, especially in regions that are already vulnerable to supply shortages. With the world's population growing and climate conditions deteriorating, maintaining plant health is an urgent challenge in achieving sustainable agricultural [2]

To respond to the these issues, researchers have increasingly explored methods that combine deep feature extraction with machine learning classifiers [3]. This combination is commonly used to balance feature representation and algorithm simplicity in image classification tasks. Several recent studies have highlighted that class imbalance is a persistent challenge in plant disease image datasets, commonly encountered in field conditions in agriculture where some diseases are more prevalent than

others [4] [5]. In widely used datasets such as PlantVillage, common diseases like early blight are represented by thousands of images, whereas rare diseases may have fewer than a few hundred samples.

This disproportionate representation can significantly affect the learning process, leading to biased classification results that favor the majority classes and reduce the sensitivity toward minority classes [6] [7]. The impact of imbalance is especially critical in agricultural applications, where minority class diseases, though rare, may be more devastating and require rapid detection to prevent outbreaks [8]. Addressing this issue has therefore become an essential step in building robust plant disease detection systems that are both accurate and fair across all categories.

Machine Learning (ML) and Deep Learning (DL) have been widely explored as promising approaches for developing automated systems in plant disease detection through image classification [9] [10]. Several studies have demonstrated high classification accuracy in controlled environments using deep learning models such as MobileNet and Xception, which achieved 95.80% and 94.64% accuracy respectively in maize leaf disease classification [11], as well as machine learning algorithms like K-Nearest Neighbors 97% and Support Vector Machines 88% [12]. However, despite these promising results, such technologies are still rarely implemented in real-world agricultural settings. In many agricultural regions especially rural and resource limited areas the absence of machine learning and deep learning technologies result in delayed or inaccurate disease diagnosis, inaccurate disease diagnosis, ineffective pest control, increased crop losses, and a reliance on manual inspection by non-experts. This limitation directly affects productivity and prevents farmers from making timely, data-driven decisions. Consequently, the lack of such technologies continues to contribute to disparities in agricultural efficiency and food availability [13] [14].

This study explores the use of machine learning algorithms for plant disease classification in the context of imbalanced data. In contrast to previous studies that often apply oversampling techniques such as SMOTE [15], this research adopts under-sampling to achieve balanced class distribution by reducing the number of majority class samples. The goal is to observe whether this technique can still support reliable classification results, particularly in situations where data imbalance may affect model fairness. The research aims to provide insights into the effectiveness of machine learning methods for plant disease detection when combined with under-sampling and deep feature extraction, especially in real-world cases where imbalanced datasets are common.

II. METHODOLOGY

This Research proposes a classification approach for plant disease detection that integrates feature extraction using ResNet50, data balancing, dimensionality reduction, and machine learning algorithms. The methodology consists of several stages: data gathering, feature extraction, data balancing, dimensionality reduction, data preparation, data normalization, model training, and performance evaluation. Figure 1 illustrates the overall workflow of this research.

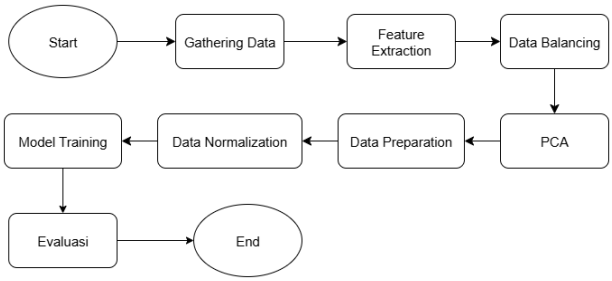

Figure 1. Workflow of this research

A. Gathering Data

This study employed the PlantVillage dataset, which is publicly available on Kaggle. The dataset consists of 54,303 colored images of plant leaves, each categorized into one of 38 distinct plant disease classes spanning various crop species, including tomato, apple, potato, corn, and grape. The images were collected under controlled conditions to minimize background noise and ensure visual consistency. While the dataset includes several classes labelled as healthy, this study does not differentiate between healthy and diseased categories. Instead, all 38 classes are treated equally as part of multiclass classification. The goal is to train a model capable of recognizing and distinguishing among all specific plant conditions, regardless of whether they indicate a healthy or diseased state. This approach enables the model to perform fine grained classification rather than binary health status detection.

A notable limitation of the dataset lies in its imbalanced class distribution. Some classes contain more that 4000 images, while others have fewer that 1200. This imbalance can lead to biased model learning, favoring majority classes and reducing the models generalization performance for underrepresented categories. As illustrated in Figure 4, the healthy class along with several dominant disease classes disproportionately represent the dataset, highlighting the need for balancing strategies.

To mitigate this issue, class distribution balancing was applied in the methodology phase, following feature extraction. Specifically, under-sampling was used to reduce the number of samples in overrepresented classes, resulting in a more uniform distribution across classes. Figure 2 presents representative sample images of disease classes, while Figure 3 shows examples of healthy leaf images.

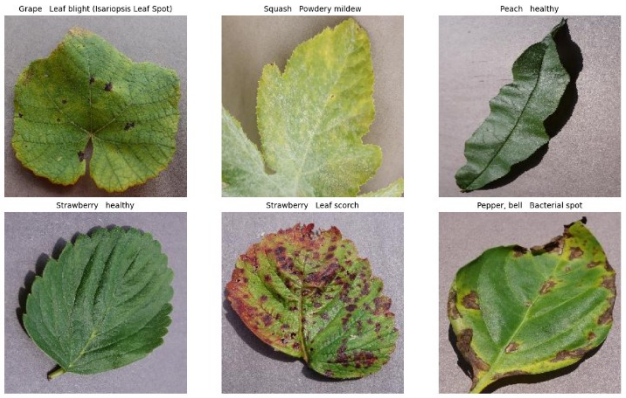

Figure 2. Representative images of diseased leaves

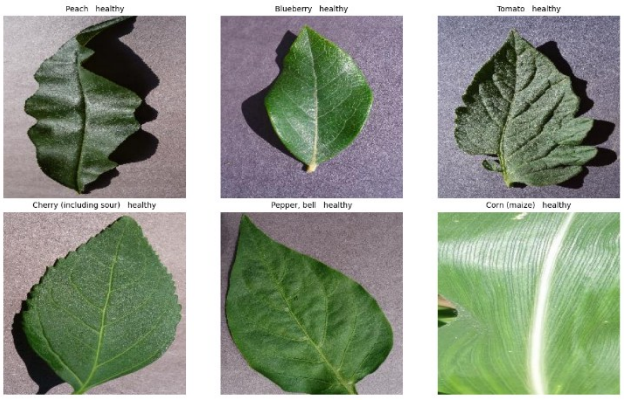

Figure 3. Representative images of healthy leaves

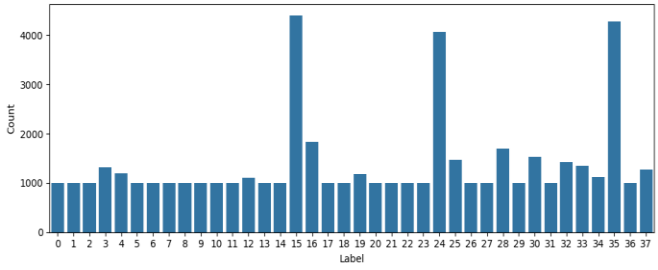

Figure 4. Original class distribution of PlantVillage dataset.

B. Feature Extraction

In this research, a pretrained ResNet50 model was utilized for feature extraction to generate high-level representations from plant leaf images. ResNet50 is a deep convolutional neural network known for its strong performance in visual recognition tasks and its use of residual learning, which helps improve training efficiency in deeper architectures. The model employed weights pretrained on the ImageNet dataset, allowing it to capture transferable visual features without requiring full retraining. All input images were resized to 224 x 224 pixels to match the models input requirements, and standard normalization was applied to ensure compatibility with the pretrained weights. The final classification layer was removed, and features were extracted from the global average pooling layer. Each image produced a 2048 dimensional feature vector containing abstract visual patterns relevant to plant disease characteristics. These feature vectors were then used as input for machine learning algorithms, forming a two-stage pipeline in which the CNN handled feature encoding and the machine learning classifiers performed the final prediction.

To better interpret the structure of the extracted features, a 2D visualization was generated to illustrate the distribution of samples within the feature space. As shown in Figure 3, the projection reveals discernible clusters corresponding to different disease classes, suggesting that the extracted features effectively capture patterns relevant to class separation.

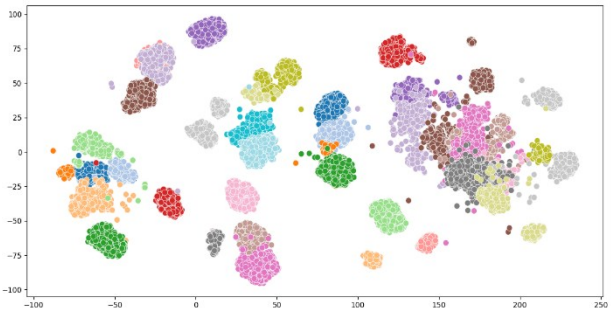

Figure 5. 2D Visualization of Deep Feature Representations

C. Data Balancing

Following the feature extraction stage, the dataset revealed a substantial imbalance in class distribution, wherein certain disease categories contained several thousand samples, while others comprised considerably fewer instances. Such imbalance poses a critical challenge in supervised learning, as classifiers tend to be biased toward majority classes, often resulting in suboptimal generalization and reduced predictive performance on minority classes [16].

To address this, class balancing techniques can be broadly categorized into undersampling and oversampling approaches. Undersampling methods reduce the number of majority class samples to achieve proportional representation across classes. This approach offers advantages such as reduced computational complexity and mitigation of overfitting in large datasets, though it carries the inherent risk of discarding informative samples from the majority class. Conversely, oversampling techniques increase the number of minority class instances by duplicating existing samples like random oversampling or generating synthetic samples like the synthetic minority oversampling techniques, thereby enhancing class representation at the potential cost of introducing noisy or redundant data. Some methods combine these approaches by augmenting minority class data while simultaneously controlling the size of the majority class to balance the trade-offs.

To support the selection of an appropriate balancing technique for this study, relevant findings from [17] and [8] are discussed. Table I. Summarizes the methodological characteristics, advantages, limitations, and recommended

scenarios for each approach. Based on these insights and considering characteristics of the dataset, a suitable method was chosen accordingly.

TABEL I
COMPARATIVE ANALYSIS OF BALANCING TECHNIQUES

Study	Techniques	Advantages	Limitations	Best Use case
Wongvarachan et al [17]	RUS	Reduces bias toward majority class.	Potential loss of majority class information.	Large datasets with extreme imbalance, time critical scenarios.
	ROS	Directly increases minority class representation.	Risk of overfitting due to duplicated samples.	Small datasets with low noise in minority class.
	SMOTE + RUS	Balances precision and recall, reduces oversampling noise.	Requires complex tuning.	Moderate to extreme imbalance with noise.
Miftahushudur et al [8]	RUS	Effective for multi class datasets with ≥1,000 minority samples.	Risk of losing majority class variation	Large scale agricultural datasets
	SMOTE	Suitable when classes have clear decision boundaries	Prone to noise if class separation is poor	Datasets with well separated class distributions

Based on these comparative insights and considering the characteristics of the PlantVillage dataset, random under sampling was selected as the balancing strategy for this study. The PlantVillage dataset comprises 38 classes, including both disease categories and healthy samples, with the smallest minority class containing over 1000 samples. Under such conditions, the risk of information loss from the majority class often cited as the primary drawback of undersampling is substantially reduced. Furthermore, given the large scale of the dataset, Random Under Sampling (RUS) effectively reduces the dataset size to a more manageable scale, facilitating efficient model training and experimentation within reasonable resource constraints.

D. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique that transforms high dimensional data into a lower dimensional space while preserving as much variance as possible. By projecting the original feature space onto a set of orthogonal components ranked by their explained variance, PCA facilitates faster computation, reduces the risk of overfitting, and enhances model interpretability particularly when working with high dimensional inputs such as deep learning feature vectors [18].

In this study, PCA was applied to the 2048 dimensional feature vectors extracted from plant leaf images using a pretrained ResNet50 model. The primary goal was to reduce the feature dimensionality while retaining the majority of meaningful variance present in the data. The number of retained components was limited to the top 100 principal components. This configuration preserved approximately 83.40% of the total variance, which was deemed sufficient to maintain the discriminative power of the feature representations while significantly reducing computational complexity. The retained components achieved a practical balance for either preserving meaningful information and reducing dimensionality, enabling more efficient process without excessive loss of variance.

E. Data Preparation

After applying an under-sampling process to address class imbalance, the resulting dataset contained an equal number of samples for each of the 38 plant disease classes. This balanced configuration was essential to ensure that the classification model would not be biased toward overrepresented categories and could treat all classes equally during training. Next, the dataset was divided into training and testing subsets using an 80:20 ratio. To maintain proportional representation of each class in both partitions, a stratified sampling strategy was used. This approach ensured that each class was equally represented in both the training and testing sets, preventing any distributional skew that could impact model evaluation. As a result, 30400 samples were allocated for training and 7600 for testing. The overall structure of the class wise split is summarized in Table II.

TABEL II
TRAIN/TEST SPLIT

Information	Training Data	Testing Data
Proportion	80%	20%
Amount	30400	7600

Based on Table II, this research used hold-out validation where the dataset is split into a portion which 80% for training data, 20% for testing data. The data for training and testing are randomly selected. Since the parameters for each model vary, this research doesn't employ a stratified sampling strategy and directly uses the configuration of the split data instead.

F. Data Normalization

To ensure that all features were on a comparable scale, normalization was applied to the extracted feature vectors

before classification. The high dimensional features obtained from the ResNet50 model exhibit varying magnitudes, which can potentially bias algorithms that depend on distance or gradient based optimization.

In this work, we used standard normalization through the StandardScaler method, which transforms the data to have a mean of zero and a standard deviation of one. Normalization is crucial because it prevents features with larger numeric ranges from dominating the learning process, especially in algorithms like SVM and KNN that are sensitive to the scale of features. The scaler was fitted solely on the training data to maintain the integrity of the evaluation process, and the same transformation was applied to the test data. By ensuring uniform feature scaling, this step contributes to a more stable and balanced model performance across all input dimensions.

G. Models

To perform multiclass classification on the extracted deep features, this study implemented eleven supervised machine learning algorithms representing various learning paradigms. These include linear classifiers, tree-based models, probabilistic methods, instance based approaches, and ensemble techniques, allowing for a comprehensive comparison of performance across different model types.

The classifiers tested were Logistic Regression, Ridge Classifier, and Support Vector Machine (SVM), which are commonly used for high dimensional problems due to their ability to handle linear or margin based decision boundaries. Decision Tree and Random Forest were used to explore the capacity of hierarchical and non liner partitioning, while K-Nearest Neighbors (KNN) provided a distance based alternative. Naive Bayes served as a probabilistic baseline, offering a simple yet interpretable approach. Furthermore, ensemble algorithms such as Gradient Boosting, AdaBoost, XGBoost, and LightGBM were incorporated to assess the benefits of boosting and bagging strategies for improving generalization.

All models were trained using standardized and PCA-reduced features derived from the training set. To ensure robust performance evaluation, 5-fold cross-validation was employed across all models. This strategy reduced bias due to random data splits an ensured consistent validation across the classifier set. Default hyperparameters were used unless otherwise noted, and results were reported based on average scores abtained across folds.

$$\mathrm{AUC}_i = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\ d(\mathrm{FPR}) \tag{7}$$

H. Evaluation Metrics

To rigorously assess the performance of the classification models, this study employed a comprehensive set of evaluation metrics encompassing both overall and class assessments. Relying solely on accuracy, which calculates the proportion of correctly classified instances out of the total number of predictions, can be misleading particularly in multiclass scenarios with imbalanced data distributions. In such settings, a model may appear accurate simply by favoring dominant classes, while underperforming on minority classes.

To provide a more detailed and fair evaluation, additional metrics were incorporated, namely precision, recall, and F1-score. Precision measures the proportion of true positive predictions among all instances labeled as positive by the model, indicating how reliable the model's positive predictions are. Recall, on the other hand, captures the model's ability to identify actual positive instances, reflecting its sensitivity to minority classes. The F1-score, defined as the harmonic mean of precision and recall, offers a balanced metric that becomes particularly important when precision and recall exhibit trade-offs. To formally define these evaluation metrics and support their interpretation, their respective expressions are provided in the following Equation 2-5.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5}$$

Where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. To ensure robustness in model evaluation, these metrics were computed using 5-fold cross-validation, providing performance averages across multiple data partitions and minimizing evaluation bias due to arbitrary data splits.

In addition, the ROC AUC score was calculated in a one-vs-rest (OvR) configuration to measure the models' capability in distinguishing each class from the others. For each class i, the ROC AUC is defined as the area under the ROC curve constructed from the class's true positive rate (TPR) and false positive rate (FPR), as shown in Equation 6-8:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

$$\text{AUC}_{\text{macro}} = \frac{1}{C}\sum_{i=1}^{C}\text{AUC}_i \quad (8)$$

By integrating both threshold-based metrics such as accuracy, precision, recall, and F1-score and ranking based metrics like ROC AUC, this evaluation approach delivers a more insightful and multidimensional understanding of model behavior. It captures not only how well the models classify instances, but also how confidently and consistently they distinguish between classes, particularly in the presence of class imbalance where accuracy alone may fall short.

III. Results and Discussions

This section presents the experimental results and offers a comprehensive analysis of the classification performance achieved by the selected machine learning models. The evaluation considers key performance metrics, including accuracy, precision, recall, F1-score, and ROC AUC, to provide both quantitative and qualitative insights into the models' generalization capabilities on the balanced, feature extracted dataset. The impact of preprocessing stages including feature extraction, dimensionality reduction, and class balancing is also examined to better understand their contribution to overall model performance.

A. Data Balancing

After applying random under-sampling at the feature level, the dataset was transformed into a more balanced state. Prior to random under-sampling, the dataset comprised 54,303 samples with highly uneven distribution across 38 classes. The majority of classes contained just over one thousand samples, while a few moderately dominant classes had close to two thousand samples. Three classes were particularly large, with labels 15, 24, and 35 containing approximately 4,303, 4,000, and 4,200 samples respectively. This imbalance posed a risk of bias in the learning process, potentially causing the model to favor majority classes and perform poorly on minority ones. Through random under-sampling, each of the 38 classes was adjusted to contain exactly 1,000 samples, producing a balanced dataset of 38,000 samples in total. The class distributions before and after under-sampling are presented in Figure 6 and Figure 7. As illustrated in Figure 6, several dominant classes initially accounted for a disproportionate share of the data, whereas Figure 7 demonstrates a uniform representation across all categories.

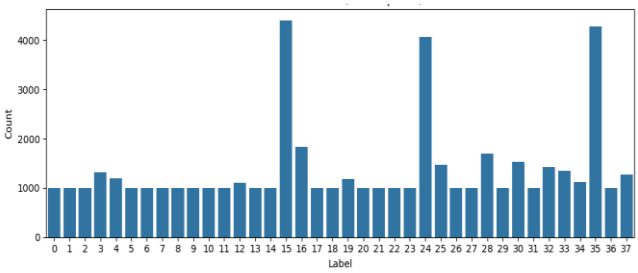

Figure 6. Before Under-Sampling

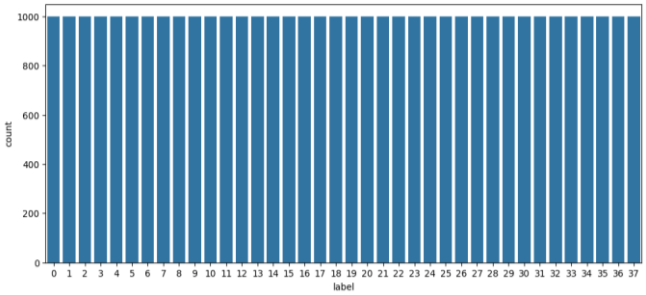

Figure 7. After Under-Sampling

By ensuring equal representation of each class during training, this step helped reduce learning bias and enhanced the classifier's ability to generalize across categories. The balanced dataset also supported fairer evaluation metrics and minimized the influence of class dominance during model training.

B. Model Performance and Evaluation

To comprehensively assess the classification performance, a total of eleven machine learning models were trained and validated using 5-fold cross-validation on the normalized and PCA reduced feature vectors extracted from the ResNet50 model. In addition to cross-validation, all models were also tested on a completely unseen dataset, outside of the cross-validation folds, to evaluate their generalization performance on previously unobserved data. The assessment employed a combination of global and per-class metrics, including accuracy, precision, recall, F1-score, and ROC AUC score, to capture both overall effectiveness and class level behavior across a multiclass setting. The consolidated results are presented in Table III, which serves as the primary basis for interpreting the comparative strengths and limitations of each model.

TABEL III
EVALUATION MODEL COMPARISON USING 5-FOLD CROSS-VALIDATION

Model	Accuracy	Precision	Recall	F1-Score	AUC Score
Random Forest	0.9521	0.9522	0.9521	0.9517	0.9609
Gradient Boosting	0.9507	0.9479	0.9465	0.9478	0.9447
Logistic Regression	**0.9733**	**0.9734**	**0.9733**	**0.9732**	**0.9807**
AdaBoost	0.1377	0.0845	0.1377	0.0712	0.8402
Support Vector Machine	**0.9963**	**0.9962**	**0.9960**	**0.9958**	**0.9998**
Decision Tree	0.7838	0.7817	0.7838	0.7816	0.8884
LightGBM	**0.9625**	**0.9631**	**0.9625**	**0.9626**	**0.9895**
K-Nearest Neighbors	0.9437	0.9450	0.9437	0.9427	0.9569
XGBoost	0.9563	0.9565	0.9563	0.9562	0.9776
Ridge Classifier	0.9069	0.9078	0.9069	0.9028	0.9677
Naïve Bayes	0.9026	0.9073	0.9026	0.9040	0.9480

The evaluation results indicate that the Support Vector Machine (SVM) achieved the highest classification performance across all evaluation metrics, with an accuracy of 99.63%, precision of 99.62%, recall of 99.60%, F1-score of 99.58%, and an AUC score of 0.9998. Such exceptional performance reflects the model's strong generalization capability when trained on high-dimensional deep features extracted from ResNet50, particularly in a context where the dataset is clean, balanced through under-sampling, and composed of images that exclusively highlight disease and healthy related visual patterns without background noise. The high accuracy, while seemingly extraordinary, is not the result of dataset bias. Instead, it can be attributed to the intrinsic characteristics of the dataset, in which each sample prominently displays the diseased and healthy leaf with minimal variation in lighting, background, or occlusion, thus enabling the SVM to identify and exploit highly discriminative decision boundaries. The margin based learning principle of SVM, which aims to maximize the separation between classes in a high-dimensional space, further enhances its ability to achieve practically complete separability under these conditions without overfitting.

The remaining models also delivered strong results, albeit with slightly lower scores. Logistic Regression attained an accuracy of 97.33% and LightGBM achieved 96.25%, both with AUC scores exceeding 0.98. This demonstrates that even models with fundamentally different learning paradigms such as linear classifiers and gradient boosting methods can achieve robust generalization when trained on deep features that are both discriminative and representative of the target classes. Random Forest and XGBoost exhibited similarly solid capabilities, with accuracies surpassing 95% and AUC scores approaching 0.98, highlighting their proficiency in modeling non-linear relationships and capturing intricate feature interactions.

In contrast, AdaBoost demonstrated the lowest performance, achieving an accuracy of only 13.77%, precision of 8.45%, recall of 13.77%, F1-score of 7.12%, and an AUC score of 0.8402. Its dependence on weak base learners, combined with the reduced feature diversity resulting from under-sampling, appears to have substantially impaired its learning capability. These findings indicate that certain ensemble methods particularly those relying heavily on weak learners may be less effective for high dimensional deep features, especially when the dataset has been constrained through balancing techniques.

In the next tier of performance, K-Nearest Neighbors, Ridge Classifier, and Naïve Bayes produced competitive yet comparatively lower results, with accuracies between 90% and 94% and slightly reduced AUC scores. These outcomes suggest that while these methods can still generalize effectively from high-quality deep features, their inherent assumptions and model structures may limit their adaptability to the more complex feature distributions produced by ResNet50. The Decision Tree algorithm, despite benefiting from the structured nature of the extracted features, achieved an accuracy of 78.38% and an AUC score of 0.8884, indicating limited capacity to fully capture the complex decision boundaries required for optimal classification in this domain.

While overall performance metrics such as accuracy and F1-score provide a general indication of model effectiveness, they may obscure variations in predictive ability across individual classes. To obtain a more comprehensive understanding of model behavior, it is therefore essential to examine the per-class accuracy and error distribution. This analysis allows for the identification of categories with consistently high classification performance as well as those that exhibit greater misclassification tendencies, thereby providing a more balanced perspective on model performance beyond overall summary metrics.

Figure 8. Per-Class Accuracy Logistic Regression

Figure 8 shows the per-class accuracy of the Logistic Regression model. Overall, the model demonstrates consistently high accuracy across nearly all classes, indicating reliable overall classification. However, classes such as Maize Common Rust (7), Maize Northern Leaf Blight (9), Tomato Healthy (30), and Tomato Spider Mites (34) exhibit slightly lower accuracy, likely due to visually similar symptoms, reflecting occasional misclassification.

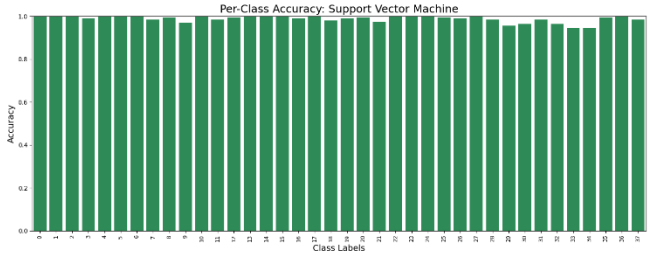

Figure 9. Per-Class Accuracy Support Vector Machine

Figure 9 presents the per-class accuracy of the Support Vector Machine (SVM) model. The evaluation shows consistently high accuracy across nearly all classes. However, slightly lower accuracy was observed in Tomato Septoria Leaf Spot (33), Tomato Spider Mites (34), and Tomato Early Blight (29), likely due to visually similar symptoms, reflecting occasional misclassification. Overall, the SVM model demonstrates robust performance.

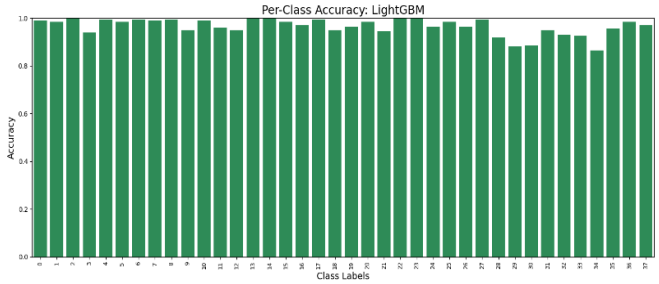

Figure 10. Per-Class Accuracy LightGBM

Figure 10 illustrates the per-class accuracy of the LightGBM model. The model demonstrated high accuracy across almost all classes. However, slightly lower accuracy was noted for the Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34) classes, indicating some instances of misclassification. Overall, the LightGBM model shows robust performance.

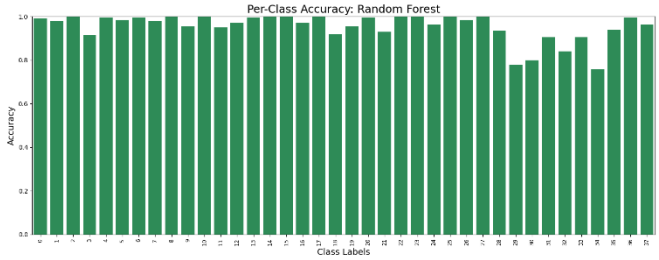

Figure 11. Per-Class Accuracy Random Forest

Figure 11 shows the per-class accuracy of the Random Forest model. Overall, the model shows generally high accuracy, with several classes reaching near-perfect scores, while a few others record comparatively lower performance. However, slightly lower performance was observed in Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34), reflecting occasional misclassification. Overall, the Random Forest model maintains robust performance.

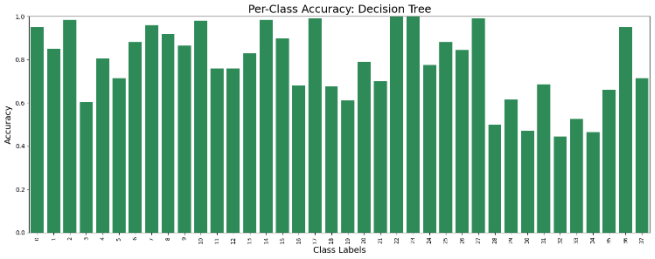

Figure 12. Per-Class Accuracy Decision Tree

Figure 12 shows the per-class accuracy of the Decision Tree model. The results indicate that while the model achieves high accuracy in several classes, it also exhibits noticeably lower accuracy in others, particularly Apple Healthy (3), Tomato Bacterial Spot (28), Tomato Healthy (30), Tomato Leaf Mold (32), Tomato Septoria leaf Spot (33), and Tomato Spider Mites (34), reflecting occasional misclassification. Overall, the Decision Tree model maintains robust performance.

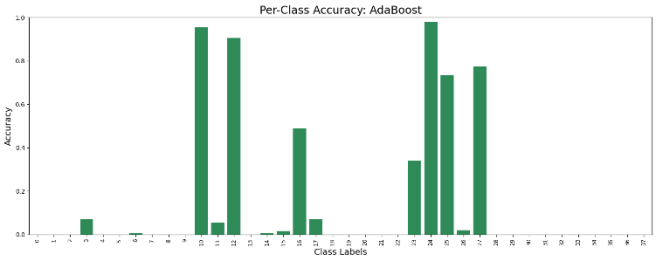

Figure 13. Per-Class Accuracy AdaBoost

Figure 13 shows the per-class accuracy of the AdaBoost model. The results indicate that most classes, including Apple Scab (0), Apple Black Rot (1), Apple Cedar Rust (2), Blueberry Healthy (4), Cherry Healthy (5), Cherry Powdery Mildew (6), Maize Common Rust (7), Grape Healthy (13), Grape Leaf Blight (14), Grape Healthy (13), Orange Haunglongbing (15), Peach Bacterial Spot (17), Peach Healthy (18), PepperBell Bacterial Spot (18), PepperBell Healthy (19), Potato Early Blight (20), Potato Healthy (21), Potato Late Blight (22), Tomato Bacterial Spot (28), Tomato Early Blight (29), Tomato Healthy (30), Tomato Late Blight (31), Tomato Leaf Mold (32), Tomato Septoria Leaf Spot (33), Tomato Spider Mites (34), Tomato Target Spot (35), Tomato Mosaic Virus (36), and Tomato Yellow Leaf (37), demonstrate very low performance, with accuracy values close to zero. This widespread underperformance highlights

AdaBoost's inability to generalize across the dataset, leading to highly inconsistent results and limited reliability for multi-class classification tasks.

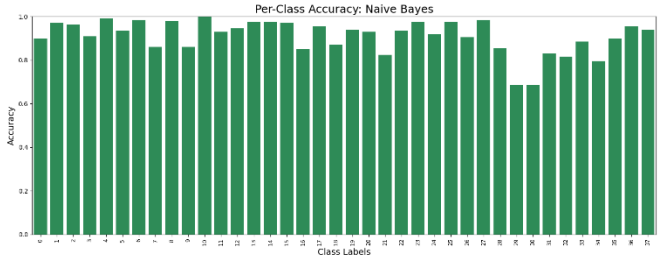

Figure 14. Per-Class Accuracy Naive Bayes

Figure 14 shows the per-class accuracy of the Naive Bayes model. The results demonstrate generally high accuracy across many classes, but with noticeable variation among classes. In particular, classes Potato Healthy (21), Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34) record lower accuracy compared to the rest, indicating that the model does not maintain uniform reliability across all classes.

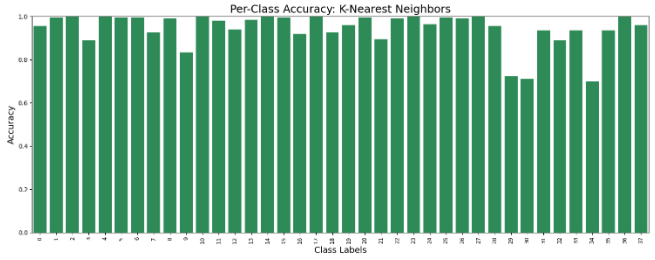

Figure 15. Per-Class Accuracy K-Nearest Neighbors

Figure 15 shows the per-class accuracy of the K-Nearest Neighbors (KNN) model. Overall, the model achieves high accuracy across most classes, demonstrating its effectiveness in distinguishing between them. However, certain classes, specifically Maize Northern Leaf Blight (9), Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34), show relatively lower accuracy, which suggests there may be challenges in maintaining consistent performance across all classes. Despite these discrepancies, the KNN model generally displays reliable classification performance.

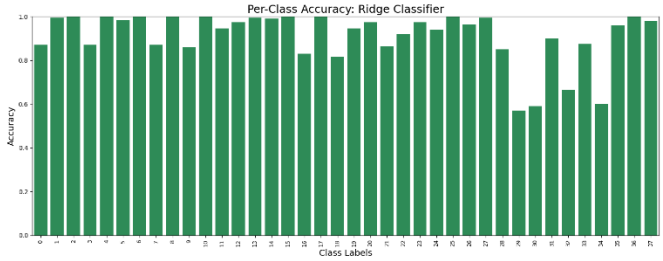

Figure 16. Per-Class Accuracy Ridge Classifier

Figure 16 shows the per-class accuracy of the Ridge Classifier model. The results reveal considerable variation across classes, with a few achieving relatively higher accuracy, while others, such as Tomato Early Blight (29), Tomato Healthy (30), Tomato Leaf Mold (32), and Tomato Spider Mites (34), show noticeably lower accuracy. This indicates that the Ridge Classifier does not perform consistently across all classes.

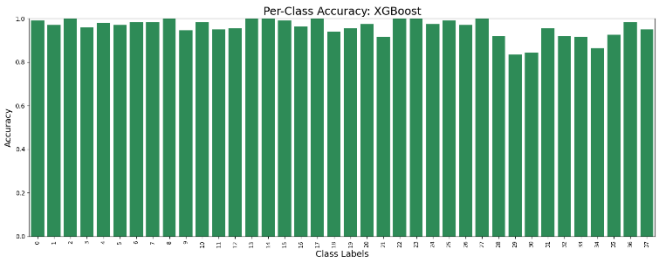

Figure 17. Per-Class Accuracy XGBoost

Figure 17 illustrates the per-class accuracy of the XGBoost model. The model achieves reasonable accuracy across several classes. however, it shows lower accuracy in the Tomato Late Blight (29) and Tomato Healthy (30) classes, indicating some classification errors in these classes. Overall, the XGBoost model demonstrates fairly consistent performance.

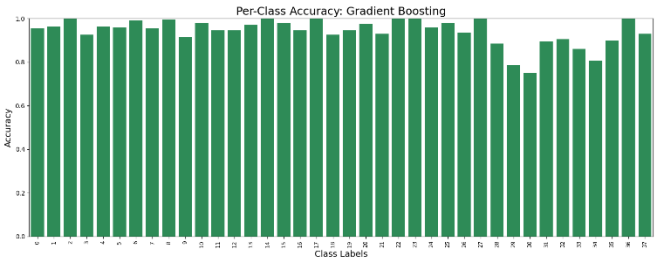

Figure 18. Per-Class Accuracy Gradient Boosting

Figure 18 shows the per-class accuracy of the Gradient Boosting model. The model achieved generally high performance across most classes, although only a few reached perfect accuracy. Lower performance was observed in Tomato Bacterial Spot (28), Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34), reflecting occasional misclassification. Overall, the Gradient Boosting model still provides reliable classification results.

The per-class accuracy analysis shows that misclassification most frequently occurs in classes such as Tomato Early Blight (29), Tomato Healthy (30), and Tomato Spider Mites (34), which consistently record lower performance across nearly all models. Other classes, including Tomato Leaf Mold (32), Tomato Septoria Leaf Spot (33), and Tomato Bacterial Spot (28), also tend to be more challenging to classify correctly.

This study presents a comparative visualization of training and test accuracy across all evaluated models. The resulting bar chart serves as a clear and informative tool for assessing the consistency and generalization capability of each classifier. Models with closely aligned training and test performance indicate robust generalization, while significant

discrepancies may point to overfitting or underfitting. This visual representation provides valuable insight into how each model responds to the dataset and the preprocessing pipeline, complementing the quantitative metrics discussed earlier.

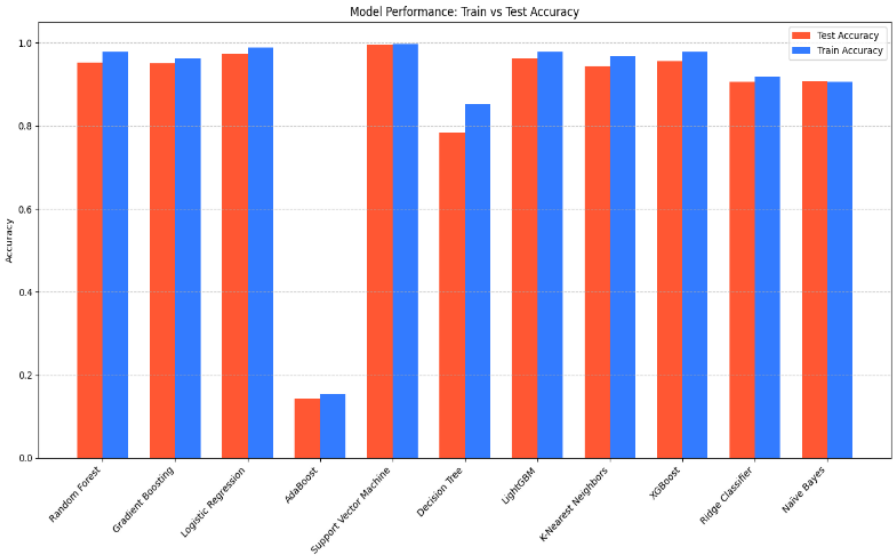

Figure 14. Model Performance Train & Test Acuracy

Figure 14 illustrates a comparative analysis of training and test accuracy across eleven machine learning models evaluated in this study. The primary purpose of this visualization is to assess each model's generalization capability its ability to perform consistently on unseen data after being trained. Overall, most models demonstrate a relatively small gap between training and test accuracy, indicating stable performance, while a few models show a more pronounced disparity, suggesting potential overfitting or underfitting behavior.

Among the best-performing models, Support Vector Machine (SVM) and Logistic Regression exhibit near-identical accuracy scores on both training and test sets, reflecting excellent generalization and robustness. Similarly, Random Forest, LightGBM, XGBoost, and K-Nearest Neighbors achieve high accuracy on both datasets, with only marginal differences between training and testing phases, highlighting their capacity to model the data effectively without significant overfitting. Ridge Classifier and Naïve Bayes, though slightly lower in accuracy, still maintain balanced performance, demonstrating consistent behavior under the experimental setting.

In contrast, Decision Tree and particularly AdaBoost reveal significant discrepancies between training and test accuracy. The Decision Tree model shows signs of moderate overfitting, achieving a relatively high training accuracy but dropping notably in the test phase, which suggests the model may have memorized training patterns without capturing generalizable structures. On the other hand, AdaBoost performs poorly on both training and test data, with accuracy scores significantly below those of other models, indicating limited learning capacity or poor adaptation to the characteristics of the dataset. These observations underscore the importance of model selection and the role of preprocessing steps in influencing predictive performance.

IV. CONCLUSION

This research explored the integration of deep feature extraction with machine learning methods for the multiclass classification of plant leaf diseases. Utilizing a pretrained ResNet50 model, high-level visual features were extracted from 38 plant leaf classes, encompassing both healthy and diseased samples. These features were then standardized, reduced in dimensionality using Principal Component Analysis (PCA), and used as input for various machine learning classifiers. The performance evaluation, based on 5-fold cross-validation, showed that Support Vector Machine (SVM), Logistic Regression, and LightGBM consistently achieved superior results in terms of accuracy, F1-score, and ROC AUC score. These models demonstrated strong generalization capabilities across all classes, while methods like AdaBoost and Decision Tree performed less effectively, suggesting shortcoming in handling the high-dimensional feature space derived from CNN representations.

To address class imbalance in the original dataset, a random under-sampling strategy was employed following feature extraction. This preprocessing step ensured a uniform class distribution and contributed to more equitable model training. The relative performance of each model was further examined through a comparative analysis of training and test accuracy. The results highlighted the ability of the top-performing models to generalize well to unseen data, as evidenced by the minimal gap between their training and test accuracy scores, while models with wider performance gaps exhibited signs of overfitting or underfitting.

In summary, this study confirms the effectiveness of combining pretrained convolutional feature extractors, dimensionality reduction, and robust classification algorithms for plant disease recognition. Future work may involve

exploring data augmentation methods, optimizing hyperparameters for further performance gains, and expanding the dataset to include more diverse plant species and environmental conditions to improve model robustness.

REFERENCES

[1] H. H. E. van Zanten *et al.*, "Circularity in Europe strengthens the sustainability of the global food system," *Nat. Food*, 2023, doi: 10.1038/s43016-023-00734-9.

[2] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nat. Ecol. Evol.*, 2019, doi: 10.1038/s41559-018-0793-y.

[3] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," 2023. doi: 10.1016/j.atech.2022.100083.

[4] S. U. Khan, A. Alsuhaibani, A. Alabduljabbar, F. Almarshad, Y. N. Altherwy, and T. Akram, *A review on automated plant disease detection: motivation, limitations, challenges, and recent advancements for future research*, vol. 37, no. 3. Springer International Publishing, 2025. doi: 10.1007/s44443-025-00040-3.

[5] W. Ding, M. Abdel-Basset, I. Alrashdi, and H. Hawash, "Next generation of computer vision for plant disease monitoring in precision agriculture: A contemporary survey, taxonomy, experiments, and future direction," *Inf. Sci. (Ny).*, 2024, doi: 10.1016/j.ins.2024.120338.

[6] H. N. Ngugi, A. E. Ezugwu, A. A. Akinyelu, and L. Abualigah, "Revolutionizing crop disease detection with computational deep learning: a comprehensive review," 2024. doi: 10.1007/s10661-024-12454-z.

[7] P. H. Kyaw, J. Gutierrez, and A. Ghobakhlou, "A Systematic Review of Deep Learning Techniques for Phishing Email Detection," *Electron.*, vol. 13, no. 19, 2024, doi: 10.3390/electronics13193823.

[8] T. Miftahushudur, H. M. Sahin, B. Grieve, and H. Yin, "A Survey of Methods for Addressing Imbalance Data Problems in Agriculture Applications," *Remote Sens.*, vol. 17, no. 3, pp. 1–31, 2025, doi: 10.3390/rs17030454.

[9] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Comput. Intell. Neurosci.*, 2016, doi: 10.1155/2016/3289801.

[10] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, 2016, doi: 10.3389/fpls.2016.01419.

[11] D. S. Joseph, P. M. Pawar, and K. Chakradeo, "Real-Time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning," *IEEE Access*, vol. 12, no. January, pp. 16310–16333, 2024, doi: 10.1109/ACCESS.2024.3358333.

[12] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Glob. Transitions Proc.*, 2022, doi: 10.1016/j.gltp.2022.03.016.

[13] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," 2018. doi: 10.1016/j.compag.2018.02.016.

[14] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning - A Review," 2021. doi: 10.1109/ACCESS.2021.3069646.

[15] A. Bhatia, A. Chug, and A. Prakash Singh, "Application of extreme learning machine in plant disease prediction for highly imbalanced dataset," *J. Stat. Manag. Syst.*, 2020, doi: 10.1080/09720510.2020.1799504.

[16] H. Ghazouani, W. Barhoumi, E. Chakroun, and A. Chehri, "Dealing with Unbalanced Data in Leaf Disease Detection: A Comparative Study of Hierarchical Classification, Clustering-based Undersampling and Reweighting-based Approaches," in *Procedia Computer Science*, 2023. doi: 10.1016/j.procs.2023.10.489.

[17] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," *Inf.*, 2023, doi: 10.3390/info14010054.

[18] S. Ali, M. Hassan, J. Y. Kim, M. I. Farid, M. Sanaullah, and H. Mufti, "FF-PCA-LDA: Intelligent Feature Fusion Based PCA-LDA Classification System for Plant Leaf Diseases," *Appl. Sci.*, vol. 12, no. 7, 2022, doi: 10.3390/app12073514.