INTERNATIONAL JOURNAL OF APPLIED ARTIFICIAL INTELLIGENCE AND ROBOTICS
E-ISSN: 3124-1212, DOI: 10.67745/ijaic.v2i2.29
Vol. 2 No. 2 2026 pp. 74-85

A Comparative Study of Machine Learning Algorithms for Plant Leaf Disease Classification Using Hybrid Feature

Ulfia Rahmah[1], Dini Rivani[2], Muhammad Ariel[3*], Muh. Subhan[4]

[1,2] Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan komputer, Politeknik Negeri Lhokseumawe, Indonesia
[3] Program Studi Biologi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Syiah Kuala, Indonesia
[4] Program Studi Sistem Informasi, Universitas Jambi, Muaro Jambi, Indonesia

*Corresponding Author: mariel23@mhs.usk.ac.id

Article info: Received 09/09/2026, Revised 19/10/2026, Accepted 07/11/2026

This is an open access article under the CC BY-SA license.

Abstract

Plant leaf diseases are one of the major factors contributing to the decline in agricultural productivity and crop quality. Manual disease identification requires considerable time, experience, and accuracy, making it less efficient when applied on a large scale. Advances in image processing and machine learning technologies offer opportunities to develop automated plant disease classification systems based on leaf images. This study aims to compare the performance of Support Vector Machine (SVM), Random Forest, K-Nearest Neighbor (KNN), and Naive Bayes algorithms in classifying plant leaf diseases using color, texture, and shape feature extraction.The dataset used in this study is the Plant Leaf Dataset, consisting of 3,695 leaf images across four classes: Tomato Healthy, Tomato Leaf Mold, Potato Healthy, and Potato Late Blight. Color feature extraction was performed using Color Moments in RGB and HSV color spaces. Texture features were extracted using the Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP), while shape features were extracted using Contour and Hu Moments. To address data imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed.The experimental results show that SVM achieved the highest accuracy of 97%, followed by Random Forest and KNN, both achieving 95%, while Naive Bayes achieved an accuracy of 57.10%. These findings indicate that SVM is the most effective algorithm for plant leaf disease classification on the dataset used in this study.

Keywords: Plant Leaf Disease Classification, Feature Extraction, Color Moments, Gray Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP), Synthetic Minority Oversampling Technique (SMOTE)

1. Introduction

The agricultural sector serves as the backbone of the economy in many developing countries, including Indonesia. Tomato and potato plants are high-value horticultural commodities whose production is highly vulnerable to disease outbreaks. Globally, plant diseases cause annual crop losses ranging from 20% to 40%, with increasingly severe impacts in countries that still rely on manual detection based on farmers' visual observations [1]. Leaf diseases such as Potato Late Blight, caused by the pathogen *Phytophthora infestans*, and Tomato Leaf Mold, caused by *Passalora fulva*, are among the most economically damaging plant diseases in agricultural history. Potato Late Blight, in particular, was responsible for the devastating Irish Great Famine in the nineteenth century [2]. To this day, both diseases remain major agricultural threats that are difficult to control without accurate early detection [3].

Conventional approaches to diagnosing plant leaf diseases rely heavily on the expertise of agronomists or experienced farmers who can accurately recognize visual symptoms. However, these methods have significant limitations: they are subjective, time-consuming, difficult to reproduce consistently, and insufficiently scalable for large-scale agricultural field monitoring. Therefore, developing automated artificial intelligence-based systems for plant leaf disease detection and classification has become an urgent necessity [4]. Handcrafted feature extraction

approaches offer several competitive advantages over deep learning, particularly in resource-constrained computational environments [5]. Handcrafted features, such as Color Moments for color characterization, Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) for texture analysis, as well as Hu Moments and Contour descriptors for shape representation, provide rich and interpretable visual representations. When combined with appropriate classifiers, such as Support Vector Machine (SVM), Random Forest, or K-Nearest Neighbor (KNN), these approaches can achieve high classification accuracy with substantially lower computational complexity [6].

Several previous studies have investigated plant disease classification using machine learning techniques. The study by Mohanty et al. [7] pioneered the application of deep Convolutional Neural Networks (CNNs) to the PlantVillage dataset, which contains more than 50,000 images, achieving an accuracy of 99.35% in classifying 26 diseases across 14 plant species. This achievement established an early benchmark that stimulated subsequent research in this field. Further research by Tm et al. [3] developed a tomato leaf disease detection system using a modified CNN architecture and achieved an accuracy of 99.13% on the PlantVillage dataset. However, the researchers acknowledged that this high performance was largely attributable to the controlled dataset conditions, characterized by uniform backgrounds and consistent lighting, which differ considerably from real-world field conditions. Shoaib et al. [1] conducted a systematic review of more than 150 studies on plant disease detection using deep learning and found that, despite the high accuracy achieved by CNNs, several significant limitations remain, including the requirement for large volumes of labeled data, dependence on high-performance GPUs, and limited model interpretability. Their review recommended developing hybrid approaches that combine handcrafted features with conventional machine learning as a more practical alternative.

Furthermore, studies employing conventional machine learning techniques have also demonstrated competitive results. Kaur and Kang [8] implemented plant leaf disease classification using SVM and achieved an accuracy of 95.99% on the PlantVillage dataset. This study demonstrated that conventional machine learning approaches remain relevant, although the use of individual feature types has not fully exploited the available representational potential. The research gap addressed in this study lies in the comparative evaluation of several conventional machine learning classifiers, namely SVM, Random Forest, KNN, and Naive Bayes, using a comprehensive combination of color, texture, and shape features on the Plant Leaf Dataset, while explicitly addressing class imbalance through the Synthetic Minority Oversampling Technique (SMOTE). Most previous studies have focused on a single feature type or classifier, or have not adequately addressed imbalanced class distributions, which can significantly affect model reliability, particularly for minority classes.

Accordingly, this study aims to: (1) extract visual features from plant leaf images using Color Moments, GLCM, LBP, Contour descriptors, and Hu Moments; (2) evaluate and compare the classification performance of SVM, Random Forest, KNN, and Naive Bayes; and (3) analyze the impact of SMOTE-based oversampling on the classification performance of all evaluated classifiers.

2. Methods

This study employs a quantitative experimental approach using a classification pipeline consisting of two main stages: training and testing. The overall research workflow is illustrated in Fig 1.

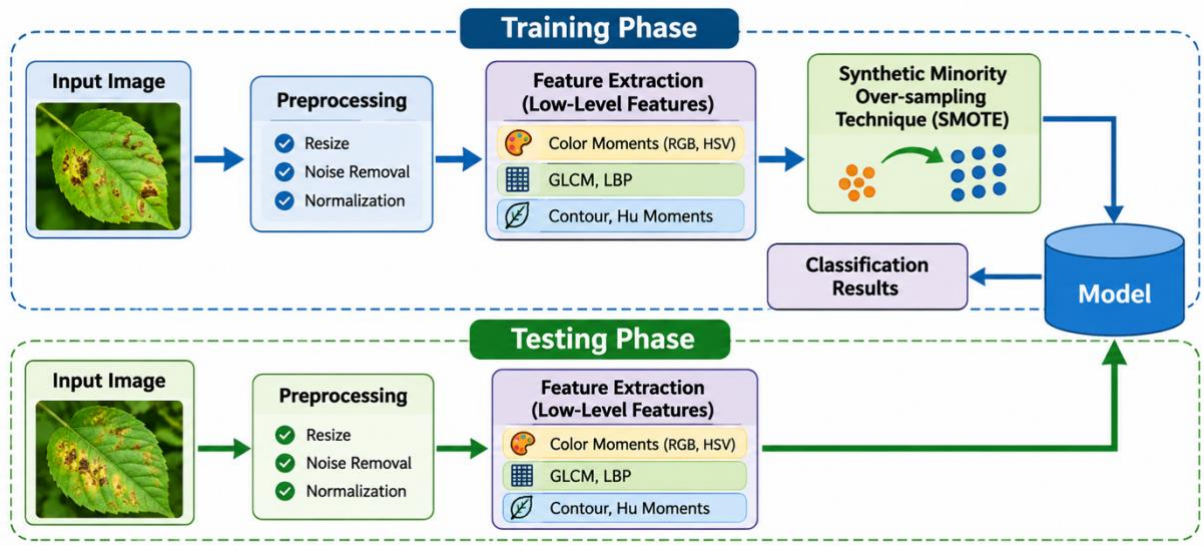


Figure 1. Research Workflow

As illustrated in Figure 1, during the training phase, the images undergo preprocessing, followed by feature extraction to obtain color, texture, and shape characteristics. Since the class distribution is imbalanced, the extracted feature data are balanced using the Synthetic Minority Oversampling Technique (SMOTE) before being used to

train four classification models: Support Vector Machine (SVM), Random Forest, K-Nearest Neighbor (KNN), and Naive Bayes. These classifiers are selected because they employ different learning mechanisms, enabling a comparative evaluation to determine which algorithm is most suitable for plant leaf disease image classification. All four models are trained using the same feature set to ensure a fair comparison, such that differences in classification performance primarily reflect the characteristics and capabilities of each algorithm.

During the testing phase, the test images undergo the same preprocessing and feature extraction procedures as those applied during the training phase, but without applying SMOTE. This ensures that the test data preserve their original class distribution and represent actual data conditions rather than artificially generated samples. The extracted features are then fed into the previously trained models. Each model predicts the class of every test image, determining whether the leaf is healthy or affected by a specific disease. Finally, the predicted labels are compared with the ground-truth labels to assess the classification performance of each model in identifying plant leaf diseases.

2.1 Dataset

The dataset used in this study is the Plant Leaf Dataset obtained from Kaggle and uploaded by Emmarex, comprising 15 classes representing different plant leaf conditions [9]. For this study, a total of 3,695 images from four selected classes were utilized. Representative images from each of the four classes used in this study are presented in Table 1.

Table 1. Representative Images of the Four Plant Leaf Classes Used in This Study

No	Images	Class
1		Tomato Healthy
2		Tomato Leaf Mold
3		Potato healthy
4		Potato Late Blight

The images presented in Table 1 consist of close-up photographs of individual leaves captured against plain gray or purple backgrounds, allowing the leaf objects to be clearly distinguished from their backgrounds. The four classes were selected based on their relevance to two major horticultural crops, namely tomato and potato. These classes include Tomato Healthy, Tomato Leaf Mold, Potato Healthy, and Potato Late Blight, representing both healthy leaves and leaves infected with fungal diseases in each plant species. The distribution of images across the

four classes is imbalanced. Tomato Healthy has the largest number of images (1,591), followed by Potato Late Blight (1,000), Tomato Leaf Mold (952), and Potato Healthy, which has the smallest number (152). This distribution reflects the data availability conditions commonly encountered in plant disease image datasets. Such class imbalance provides the primary motivation for applying the Synthetic Minority Oversampling Technique (SMOTE) during the model training phase, as described in the Methods section.

Before feature extraction, all images undergo preprocessing, including resizing to a uniform resolution and pixel value normalization, to ensure consistent input characteristics across all classes. As illustrated in Table 1, healthy leaves (Tomato Healthy and Potato Healthy) exhibit relatively uniform green coloration without visible spots or lesions. In contrast, Tomato Leaf Mold images display yellowish-green spots associated with fungal growth on the leaf surface, while Potato Late Blight images exhibit dark brown necrotic lesions with visibly water-soaked margins. These visual differences provide the basis for extracting color, texture, and shape features in the subsequent feature extraction stage.

2.2 Preprocessing

The preprocessing stage plays a crucial role in preparing image data before feature extraction and classification model training. The primary objective of this stage is to produce well-organized and consistent input data, enabling the accurate extraction of leaf visual characteristics through color, texture, and shape feature extraction methods. The Plant Leaf Dataset used in this study comprises four leaf condition categories: Tomato Healthy, Tomato Leaf Mold, Potato Healthy, and Potato Late Blight. To ensure data quality and improve model performance, three main preprocessing steps were implemented:

a. Image Resizing

Image resizing was performed to standardize the dimensions of all images in the dataset, ensuring that color, texture, and shape feature extraction could be consistently conducted at the same scale. Each image loaded using OpenCV was resized to a uniform resolution of 256 × 256 pixels using linear interpolation before further processing. Recent studies have demonstrated that variations in input image dimensions can affect model training time and classification performance, including accuracy, precision, and recall. Therefore, image resizing is an essential step in maintaining consistent visual characteristics across samples [10]. Fig. 2 illustrates a comparison of leaf images before and after the resizing process.



(a) Before resize (b) After Resize

Figure 2. Comparison of Plant Leaf Images Before and After the Resizing Process (256 × 256 Pixels).

b. Pixel Normalization

Normalization is performed to adjust the scale of pixel values to ensure greater stability during feature computation. The pixel intensity values in the R, G, B, and H, S, V channels, which range from 0 to 255, are normalized to a range of 0–1 by dividing each pixel value by 255. This normalization is performed before calculating the mean, standard deviation, and skewness values used in Color Moments feature extraction. Intensity-scale normalization is widely adopted in computer vision research because it maintains numerical consistency across images and prevents one color channel from dominating the others during statistical computations [11].

c. Synthetic Minority Over-sampling Technique (SMOTE)

Unlike deep learning approaches, which generally employ image augmentation to increase the number of samples through visual transformations, this study applies the Synthetic Minority Over-sampling Technique (SMOTE) to the

numerical feature space obtained from feature extraction to address class distribution imbalance in the dataset. SMOTE generates synthetic samples by interpolating feature values between neighboring samples belonging to minority classes in the feature space, thereby achieving a more balanced class distribution in the training data before it is used to train the classification models [12]. This SMOTE-based approach has been widely applied in SVM and KNN algorithms to address class imbalance and has demonstrated its effectiveness in improving model performance in recognizing minority classes without altering the characteristics of the original images [13].

2.3 Features Extraction

a. Color Features: Color Moments

Color Moments is a feature extraction method that characterizes the color distribution of an image using three low-order statistical moments calculated for each color channel (RGB and HSV) [14]. These three statistical moments consist of the mean, which represents the average color intensity of a particular channel, as expressed in Equation (1); the standard deviation, which measures the dispersion of color intensity values relative to their mean, as presented in Equation (2); and skewness, which describes the asymmetry of the color distribution within a particular channel, as calculated in Equation (3).

Mean (μ)

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (1)$$

standar deviation

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (2)$$

skewness(s)

$$s = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^3 \quad (3)$$

Where μ represents the mean color intensity, x_i denotes the intensity value of the i-th pixel, and N is the total number of pixels in the color channel being analyzed.

b. Texture Features: Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP)

The Gray-Level Co-occurrence Matrix (GLCM) calculates the frequency of occurrence of neighboring pixel pairs at specific distances and angles, producing texture features such as Contrast, Energy, Homogeneity, Correlation, Dissimilarity, and Angular Second Moment (ASM) [15]. Contrast measures the intensity variation between neighboring pixels, as expressed in Equation (4). Energy measures the degree of texture uniformity in an image, as presented in Equation (5). Homogeneity measures the closeness of the value distribution within the GLCM to its diagonal, as shown in Equation (6). Correlation measures the linear relationship between neighboring pixels, as presented in Equation (7). Dissimilarity measures the degree of intensity difference between neighboring pixels using linear weighting, as expressed in Equation (8). Finally, Angular Second Moment (ASM) measures the degree of texture regularity in an image, as calculated in Equation (9).

Contrast

$$\text{Contrast} = \sum_{i,j}(i-j)^2\,P(i,j) \quad (4)$$

Energy

$$\text{Energy} = \sqrt{\sum_{i,j}[P(i,j)]^2} \quad (5)$$

Homogeneity

$$\text{Homogeneity} = \sum_{i,j}\frac{P(i,j)}{1+|i-j|} \quad (6)$$

Correlation

$$\text{Correlation} = \sum_{i,j}\frac{(i-\mu_i)(j-\mu_j)\,P(i,j)}{\sigma_i\,\sigma_j} \quad (7)$$

Dissimilarity

$$\text{Dissimilarity} = \sum_{i,j}|i-j|\,P(i,j) \quad (8)$$

ASM

$$\text{ASM} = \sum_{i,j}[P(i,j)]^2 \quad (9)$$

Notation: P(i,j) represents the probability value of a pixel pair with intensities i and j in the co-occurrence matrix; μ_i, and μ_j denote the row and column means of the GLCM, respectively; while σ_i, dan σ_j represent the standard deviations of the GLCM rows and columns, respectively. Meanwhile, Local Binary Patterns (LBP) compares each pixel with its neighboring pixels to generate binary patterns, which are subsequently converted into a frequency histogram [16]. The LBP computation is presented in Equation (10), while the sign function s(x) is defined in Equation (11).

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (10)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (11)$$

Notation: g_c represents the intensity value of the center pixel; g_p denotes the intensity value of the p-th neighboring pixel; P is the number of neighboring sampling points (n_points); and R represents the radius of the neighborhood.

c. Shape Features: Contour and Hu Moments

Contour detection identifies the boundaries of leaf objects to extract geometric features, such as area and perimeter. Hu Moments generate seven moment values that are invariant to translation, rotation, and scale, making them robust descriptors for characterizing leaf shapes under different image acquisition angles [17]. The area is calculated as the total number of pixels enclosed within the closed contour of a leaf object, whereas the perimeter is determined by measuring the total length of the contour boundary, as presented in Equation (12). The computation of Hu Moments begins with the calculation of raw moments, as defined in Equation (13), followed by central moments, as expressed in Equation (14). These values are subsequently used to calculate normalized central moments and the seven Hu invariant moments, denoted as Hu1 through Hu7.

Perimeter

$$\text{Perimeter} = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (12)$$

Moment

$$m_{pq} = \sum_{x,y} x^p y^q f(x, y) \quad (13)$$

Moment central

$$\mu_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (14)$$

Notation: n represents the number of points along the contour; x_i and y_i denote the coordinates of the i-th contour point; p and q represent the moment orders; x and y denote the pixel coordinates; f(x,y) represents the pixel intensity value at coordinates (x,y); and $\bar{x}$ and $\bar{y}$ represent the centroid coordinates (center of mass) of the object.

2.4 Machine Learning Classification Algorithms

a. Support Vector Machine (SVM)

Support Vector Machine (SVM) aims to identify the optimal separating hyperplane between classes. For non-linear data, the kernel trick, specifically the Radial Basis Function (RBF) kernel, is employed to transform the data into a higher-dimensional feature space, where class separation becomes more feasible [18]. The hyperplane is defined in Equation (15), while the RBF kernel function is presented in Equation (16).

Hyperplane

$$f(x) = w \cdot x + b = 0 \quad (15)$$

Kernel RBF

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \quad (16)$$

Where w represents the weight vector; b denotes the bias term; and γ is the kernel parameter that controls the width of influence of each data point.

b. Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees trained using bagging (bootstrap sampling) and random feature selection. The predictions from individual trees are then aggregated through majority voting for classification tasks [19]. The majority voting process is presented in Equation (17), while the calculation of Gini-based feature importance is defined in Equation (18).

Majority Vote

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_n(x)\} \quad (17)$$

Feature importance

$$\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2 \quad (18)$$

Notation: $T_i(x)$ represents the prediction of the i-th decision tree; n denotes the total number of trees; p_i is the proportion of samples belonging to class i at a given node; and c represents the total number of classes.

c. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) classifies new data samples based on the majority class among their K nearest neighbors in the feature space, determined using Euclidean distance [20]. The distance between data samples is calculated using the Euclidean distance metric, as presented in Equation (19).

Euclidean Distance:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (19)$$

Notation: x and y represent two data points in an n-dimensional feature space, while n denotes the total number of features.

d. Naïve Bayes (Gaussian)

Gaussian Naïve Bayes applies Bayes' theorem under the assumption of feature independence and assumes that each continuous feature follows a normal (Gaussian) distribution [21]. The calculation of the posterior probability using Bayes' theorem is presented in Equation (20), while the Gaussian likelihood function is defined in Equation (21).

Teorema Bayes

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)} \quad (20)$$

Likelihood gaussian

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (21)$$

Notation: $P(y \mid x_1,..,x_n)$ represents the posterior probability; $P(y)$ denotes the prior probability; $P(x_i \mid y)$ represents the likelihood of feature x_i given class y; $P(x_1,..,x_n)$ denotes the evidence probability; μy and σy^2 represent the mean and variance of the feature values for class y, respectively; and exp denotes the natural exponential function (e^x).

3. Results and Discussion

3.1. Class Distribution Before and After SMOTE

The total dataset consists of 3,695 images distributed across four classes: Tomato Healthy (1,591 images), Potato Late Blight (1,000 images), Tomato Leaf Mold (952 images), and Potato Healthy (152 images). Before applying SMOTE, the training data exhibited a clear class imbalance: Tomato Healthy contained approximately 1,273 training samples (the largest class), Potato Late Blight had approximately 800 samples, Tomato Leaf Mold had approximately 762 samples, and Potato Healthy had only approximately 122 samples (the smallest class). After applying SMOTE, the number of training samples in all classes was equalized to approximately 1,273 samples per class, corresponding to the size of the majority class.

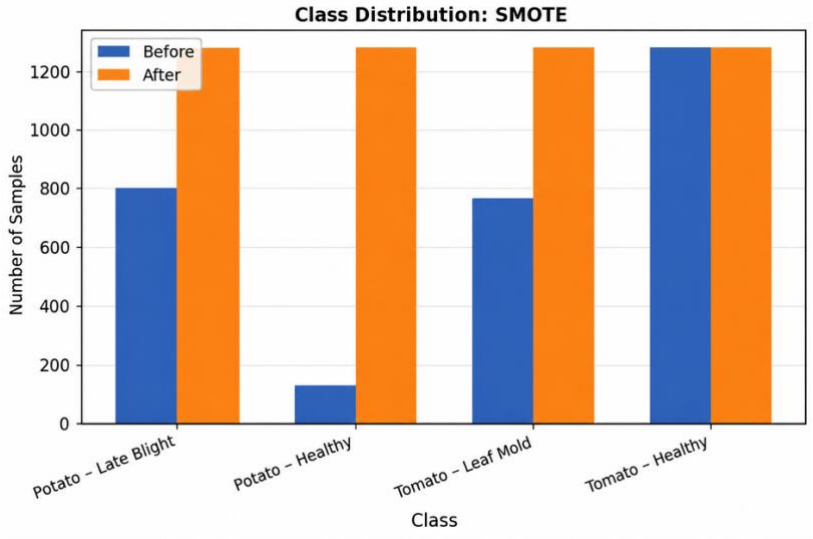


Figure 3. Comparison of Class Distribution in the Training Set Before and After SMOTE Application

These results indicate that SMOTE successfully generated synthetic samples, particularly for minority classes such as Potato Healthy, thereby producing a balanced training dataset. This balancing process enables the classification model to be trained with a more equitable class distribution and reduces the risk of bias toward particular classes. Figure 3 illustrates a comparison of the class-wise data distribution before and after the application of SMOTE.

3.2. SVM Classification Results

The SVM model demonstrates good and consistent classification performance across all four classes. The Tomato Healthy class achieves near-perfect classification, with a precision of 1.00 and a recall of 0.99, while Tomato Leaf Mold and Potato Late Blight achieve precision and recall values ranging from approximately 0.95 to 0.97. Although Potato Healthy was the minority class before resampling, it achieves a recall of 0.97, indicating that SMOTE effectively enables the model to identify nearly all samples belonging to this class. However, its precision of 0.81 indicates that some false positive predictions remain. Table 2 presents a detailed classification report for the SVM model trained.

Table 2. SVM Classification report

Class	Precision	Recall	F1-Score	Support
Tomato Healthy	1.00	0.99	0.99	318
Tomato Leaf Mold	0.97	0.97	0.97	190
Potato Healthy	0.81	0.97	0.88	31
Potato Late Blight	0.95	0.95	0.95	200
Accuracy			0.97	739

The SVM model with SMOTE demonstrates good classification performance across all four classes. The Tomato Healthy class achieves the highest number of correct predictions, with 314 out of 318 samples correctly classified, followed by Potato Late Blight with 190 out of 200 samples and Tomato Leaf Mold with 184 out of 190 samples. The Potato Healthy class is also classified correctly in 30 out of 31 samples, indicating that the model can effectively recognize minority-class samples. The remaining misclassifications occur primarily in the Potato Late Blight class, with 7 samples incorrectly classified as Potato Healthy, and in the Tomato Leaf Mold class, with 6 samples incorrectly classified as Potato Late Blight. Overall, the confusion matrix results confirm that the SVM model with SMOTE can classify the four leaf disease classes with an accuracy of 97%. The confusion matrix of the SVM model is presented in Fig. 4.

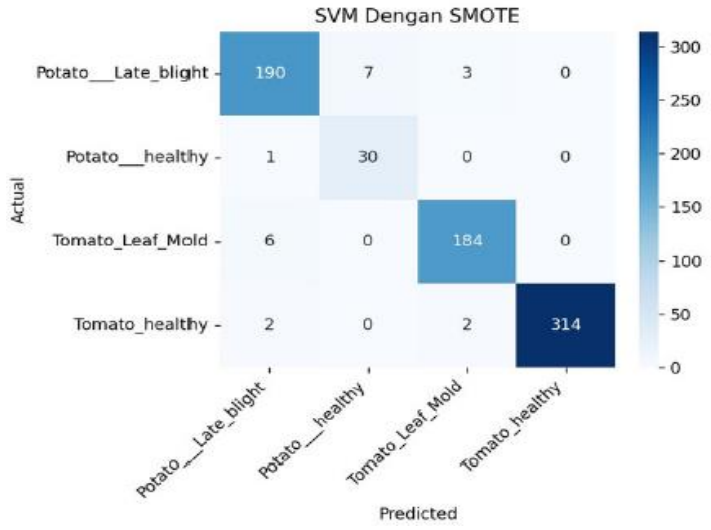


Figure 4. SVM Confusion matrix

3.3. Random Forest Classification Results

As shown in Table 3, the Random Forest model achieves its best performance on the Tomato Healthy class, with a precision of 1.00 and a recall of 0.99. Although Potato Healthy remains the class with the weakest classification performance, it achieves a recall of 0.87, indicating that SMOTE enables the model to correctly identify most samples belonging to the minority class, despite its relatively lower precision of 0.79. The confusion matrix of the Random Forest model is presented below.

Table 3 Random Forest Classification report

Kelas	Precision	Recall	F1-Score	Support
Tomato Healthy	1.00	0.99	0.99	318
Tomato Leaf Mold	0.94	0.93	0.93	190
Potato Healthy	0.79	0.87	0.83	31
Potato Late Blight	0.92	0.93	0.92	200
Accuracy			0.95	739

Based on Figure 5, the Random Forest model with SMOTE demonstrates good classification performance across all four classes. The Tomato Healthy class achieves the highest number of correct predictions, with 314 out of 318 samples correctly classified, followed by Potato Late Blight with 185 out of 200 samples and Tomato Leaf Mold with 177 out of 190 samples. The Potato Healthy class is correctly classified in 27 out of 31 samples, indicating that the model can recognize minority-class samples reasonably well. The remaining misclassifications occur primarily in the Potato Late Blight class, with 7 samples incorrectly classified as Potato Healthy and 8 as Tomato Leaf Mold, while 13 Tomato Leaf Mold samples are incorrectly classified as Potato Late Blight. Overall, the confusion matrix results indicate that the Random Forest model with SMOTE achieves an accuracy of 95% in classifying the four leaf disease classes.

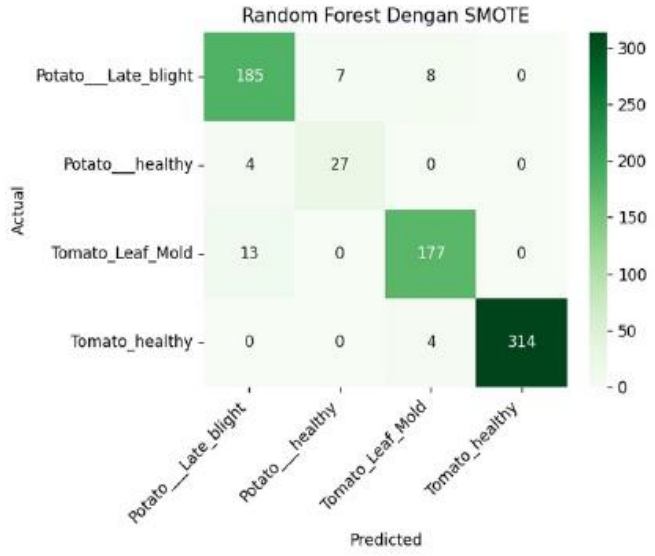


Figure 5. Confusion matrix random forest

3.4. K-Nearest Neighbors (KNN) Classification Results

The KNN model achieves a recall of 0.94 for the Potato Healthy class, representing the highest recall among the four algorithms for this minority class. However, its precision is relatively lower at 0.71, indicating a higher rate of false positive predictions. The confusion matrix of the KNN model is presented below. Table 4 presents the classification report for the KNN model (K = 5).

Table 4. KNN Classification Report

Kelas	Precision	Recall	F1-Score	Support
Tomato Healthy	1.00	0.98	0.99	318
Tomato Leaf Mold	0.96	0.95	0.95	190
Potato Healthy	0.71	0.94	0.81	31
Potato Late Blight	0.92	0.92	0.92	200
Accuracy			0.95	739

Based on Figure 6, the KNN model with SMOTE demonstrates relatively good classification performance across most classes. The Tomato Healthy class achieves 311 out of 318 correct predictions, followed by Potato Late Blight with 184 out of 200 samples and Tomato Leaf Mold with 180 out of 190 samples. The Potato Healthy class is correctly classified in 29 out of 31 samples. The remaining misclassifications occur primarily in the Potato Late Blight class, with 12 samples incorrectly classified as Potato Healthy and 4 as Tomato Leaf Mold, while 10 Tomato Leaf Mold samples are incorrectly classified as Potato Late Blight. Overall, the confusion matrix results indicate that the

KNN model with SMOTE achieves an accuracy of 95% in classifying the four leaf disease classes.

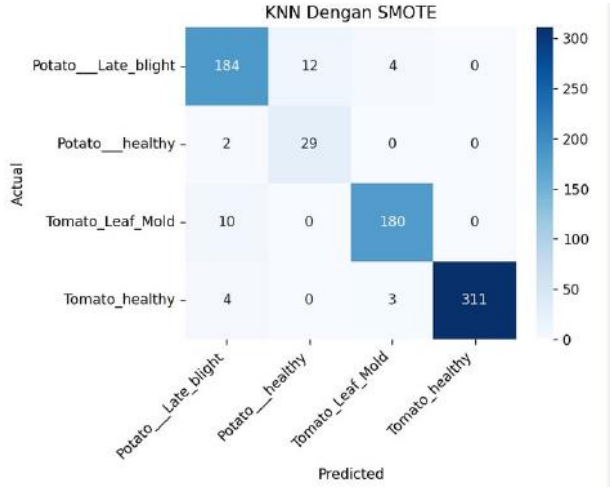


Figure 6. KNN Confusion matrix

3.5. Gausisan Naïve Bayes Classification Results

The Gaussian Naïve Bayes model demonstrates considerably weaker performance compared to the other three algorithms. The Potato Healthy class achieves a very low precision of 0.12, despite having a relatively high recall of 0.81, indicating that the model frequently misclassifies samples from other classes as Potato Healthy. Similarly, the Tomato Healthy class achieves a high precision of 0.99 but a low recall of 0.46, indicating that a substantial proportion of actual Tomato Healthy samples are incorrectly classified into other categories. Table 5 presents the classification report for the Gaussian Naïve Bayes model.

Table 5 Classification report Gaussian Naive Bayes

Class	Precision	Recall	F1-Score	Support
Tomato Healthy	0.99	0.46	0.63	318
Tomato Leaf Mold	0.59	0.71	0.64	190
Potato Healthy	0.12	0.81	0.21	31
Potato Late Blight	0.74	0.58	0.65	200
Accuracy			0.57	739

The Gaussian Naïve Bayes model with SMOTE demonstrates suboptimal classification performance compared to the other three algorithms.

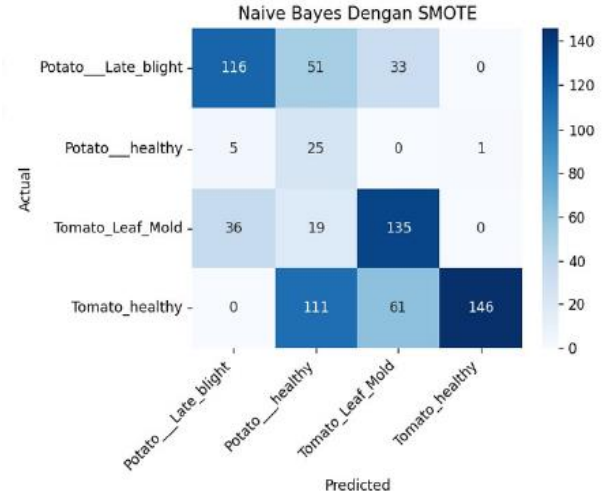


Figure 7. Naïve Bayes Confusion matrix

The Potato Late Blight class achieves only 116 correct predictions out of 200 samples, with the most significant misclassifications occurring when 51 samples are incorrectly classified as Potato Healthy and 33 samples as Tomato Leaf Mold. The Potato Healthy class achieves 25 correct predictions out of 31 samples, while 5 samples are incorrectly classified as Potato Late Blight and 1 sample as Tomato Healthy. The Tomato Leaf Mold class achieves 135 correct predictions out of 190 samples, with 36 samples incorrectly classified as Potato Late Blight and 19 samples as Potato Healthy. Meanwhile, the Tomato Healthy class achieves 146 correct predictions out of 318 samples, with substantial misclassifications, including 111 samples incorrectly classified as Potato Healthy and 61 samples as Tomato Leaf Mold. This high misclassification rate may be attributed to the feature independence assumption underlying the Naïve Bayes algorithm, which may not adequately represent the strong correlations among color, texture, and shape features. Consequently, the model achieves an overall accuracy of only 57%. The confusion matrix of the Gaussian Naïve Bayes model is shown in Figure 7.

3.6. Performance Comparison of All Classifiers

Based on Figure 8, SVM achieves the highest classification accuracy of 97%, followed by Random Forest and KNN, both of which attain an accuracy of 95%. In contrast, Gaussian Naïve Bayes records the lowest accuracy at 57%. These results indicate that kernel-based algorithms, such as SVM, and ensemble learning methods, such as Random Forest, are more suitable for plant leaf disease classification tasks using a combination of color, texture, and shape features than probabilistic algorithms such as Naïve Bayes.

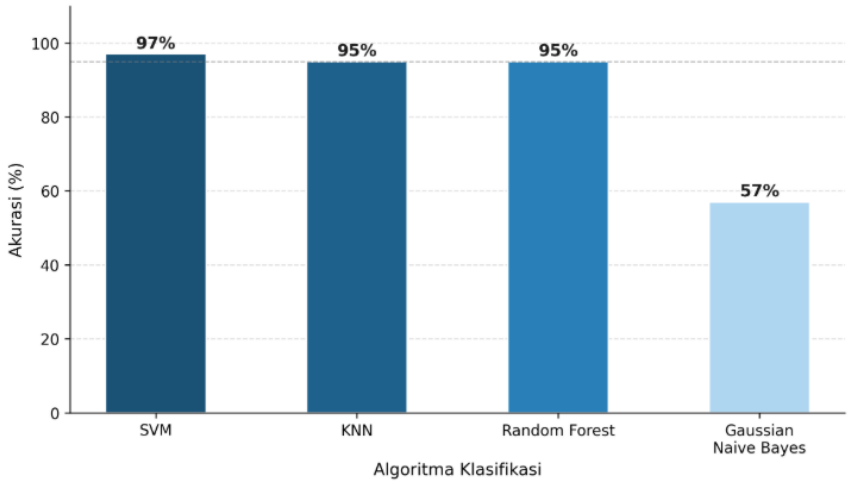


Figure 8. Performance Comparison All Classifiers

4. Conclusion

Based on the experimental results, SVM with SMOTE achieves the best overall performance, attaining the highest accuracy of 97%. This indicates that SVM is the most effective algorithm for distinguishing leaf disease classes based on color, texture, and shape features. Furthermore, its precision, recall, and F1-score values demonstrate greater consistency across all classes. Random Forest and KNN with SMOTE achieve an accuracy of 95%, indicating that both algorithms also provide good and relatively stable classification performance. Meanwhile, Gaussian Naïve Bayes records the lowest accuracy of 57%, suggesting that it is less suitable for this dataset because its feature independence assumption does not adequately reflect the characteristics of leaf image data. Overall, the application of SMOTE helps improve classification performance, particularly for minority classes. Therefore, SVM combined with SMOTE is identified as the best-performing algorithm in this study. Future research could be extended by incorporating more diverse datasets, exploring alternative feature combinations, and implementing other resampling techniques to further improve model generalization, particularly in classifying minority classes.

REFERENCES

[1] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," 2023, *Frontiers Media SA*. doi: 10.3389/fpls.2023.1158933.

[2] I. Pacal *et al.*, "A systematic review of deep learning techniques for plant diseases," *Artif. Intell. Rev.*, vol. 57, no. 11, Nov. 2024, doi: 10.1007/s10462-024-10944-7.

[3] P. Tm, A. Pranathi, K. Saiashritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," in *2018 11th International Conference on Contemporary Computing, IC3*

2018, Institute of Electrical and Electronics Engineers Inc., Nov. 2018. doi: 10.1109/IC3.2018.8530532.
[4] U. B. Gewali, S. T. Monteiro, and E. Saber, "Machine learning based hyperspectral image analysis: A survey," 2018.
[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, 2021.
[6] V. Gautam, H. Maheshwari, R. G. Tiwari, A. K. Agarwal, N. K. Trivedi, and N. Garg, "Feature Fusion-Based Deep Learning Model to Ensure Worker Safety at Construction Sites," in *2024 1st International Conference on Advanced Computing and Emerging Technologies, ACET 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ACET61898.2024.10730698.
[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, no. September, Sep. 2016, doi: 10.3389/fpls.2016.01419.
[8] A. Singh and H. Kaur, "Potato plant leaves disease detection and classification using machine learning methodologies," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1757-899X/1022/1/012121.
[9] T. O. Emmanuel, "PlantVillage Dataset," https://www.kaggle.com/datasets/emmarex/plantdisease.
[10] S. Saponara and A. Elhanashi, "Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance," in *Lecture Notes in Electrical Engineering*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 10–17. doi: 10.1007/978-3-030-95498-7_2.
[11] I. Corley, C. Robinson, R. Dodhia, J. M. L. Ferres, and P. Najafirad, "Revisiting Pre-trained Remote Sensing Model Benchmarks: Resizing and Normalization Matters," https://openaccess.thecvf.com/content/CVPR2024W/PBVS/html/Corley_Revisiting_Pre-trained_Remote_Sensing_Model_Benchmarks_Resizing_and_Normalization_Matters_CVPRW_2024_paper.html.
[12] H. Mansourifar and W. Shi, "Deep Synthetic Minority Over-Sampling Technique," Mar. 2020.
[13] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," *IEEE Access*, vol. 9, pp. 64606–64628, 2021, doi: 10.1109/ACCESS.2021.3074243.
[14] N. Bhosle and M. Kokare, "Random forest-based active learning for content-based image retrieval," *International Journal of Intelligent Information and Database Systems*, vol. 13, pp. 72–88, 2020, doi: 10.1504/IJIIDS.2020.108223.
[15] S. H. S. Al-Kilidar and L. E. George, "Texture Classification Using Gradient Features with Artificial Neural Network," *Journal of Southwest Jiaotong University*, vol. 55, no. 1, 2020, doi: 10.35741/issn.0258-2724.55.1.13.
[16] N. Alpaslan and K. Hanbay, "Multi-resolution intrinsic texture geometry-based local binary pattern for texture classification," *IEEE Access*, vol. 8, pp. 54415–54430, 2020, doi: 10.1109/ACCESS.2020.2981720.
[17] M. W. Nasrudin, N. S. Yaakob, N. A. Abdul Rahim, M. Z. Zahir Ahmad, N. Ramli, and M. S. Aziz Rashid, "Moment Invariants Technique for Image Analysis and Its Applications: A Review," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1962/1/012028.
[18] J. A. Pruneski and A. Pareek, "Support Vector Machines," *Hand Clin.*, vol. 42, pp. 19–25, Feb. 2026, doi: 10.1016/j.hcl.2025.08.003.
[19] J. Klusowski, "Sharp Analysis of a Simple Model for Random Forests," https://proceedings.mlr.press/v130/klusowski21b.html.
[20] W. Xu, Z. Yuan, and Z. Liu, "Feature Selection for Unbalanced Distribution Hybrid Data Based on k-Nearest Neighborhood Rough Set," *IEEE Transactions on Artificial Intelligence*, vol. 5, pp. 229–243, Jan. 2024, doi: 10.1109/TAI.2023.3237203.
[21] S. Pathak, S. J. Quraishi, A. Singh, M. Singh, K. Arora, and D. Ather, "A Comparative Analysis of Machine Learning Models: SVM, Naïve Bayes, Random Forest, and LSTM in Predictive Analytics," in *Proceedings - International Conference on Technological Advancements in Computational Sciences, ICTACS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 790–795. doi: 10.1109/ICTACS59847.2023.10390255.