Manuscript Received: 5 Sep. 2025, Revised: 8 Nov. 2025, Accepted: 27 Nov. 2025


# Image Based Banana Leaf Disease Detection with Machine Learning Technique

Shahrin Khan[1,3], *Anup Kumar Modak[2], Md. Ashik-E-Elahe[3], Abdullah Al-Amin[4], Iffat Mehrin Disha[5] and Dr. Sadia Sharmin[1]

[1] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology
[2] Multidisciplinary Action Research (MARS) Lab, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
[3]DIU Health Informatics Research Lab, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
[4] Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
[5] Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Email: anupkumar.cse@diu.edu.bd

**Abstract: Today, international trade has grown significantly in many countries. Many fruit products are imported from other countries, such as bananas and apples. Devastating economic losses and production losses are incurred all over the world due to fruit diseases. Little study has been done throughout the years for fruit disease detection to assist remote farmers technically. The bulk of these farmers require proper cultivation support, but little research has been done for this system. The use of one's eyes to visually inspect fruits and vegetables allows trained professionals to identify imperfect produce; however, this paper presents a lightweight machine-learning framework that integrates K-means clustering for image segmentation and a Random Forest classifier for disease recognition. A locally collected dataset of 422 banana leaf images from farms in Bangladesh was used, covering four classes: Cordana, Sigatoka, Pestalotiopsis, and Healthy. Images were preprocessed using RGB–to–Lab color conversion and resized for consistency before feature extraction based on color and texture descriptors. Among several tested classifiers, the Random Forest achieved the best performance with 96.25 % accuracy, 93.56 % precision, and 97.85 % specificity, outperforming the Decision Tree (95.28 %) and Naïve Bayes (89.27 %). Owing to its low computational cost and use of regionally collected images, this framework is suitable for real-time mobile or IoT-based agricultural systems, supporting smart and sustainable farming in resource-limited environments.**

***Keywords:*** *Banana leaf disease, Computer vision, K-means, Random Forest*, *Machine Learning*.

## 1. INTRODUCTION

Bangladesh has agriculturally oriented people, with approximately 80% of the population having a direct or indirect connection to agriculture [1]. It is a developing and known over-populated country where 47% of citizens are engaged in agricultural activities which is the way of earning their livelihood. It's also obvious that it produces a large economic impact, accounting for 16% of total GDP [2]. Various kinds of fruits, crops, and vegetables are the primary sources of income for the local population. However, farmers of Bangladesh are very unfortunate because Bangladeshi agricultural methods do not cope up with technology. Technology gives one a lot of the easiest opportunities based on unlimited characteristics to find out the best solution, especially in the field of agriculture. Many diseases impede plant growth and productivity, resulting in significant ecological and economic losses. As a result, to make maximum profit along with minimum loss, the preferable way is to accurately identify the infection properly in time. Agricultural product disease detection can be accomplished in a variety of ways, it can be manual and technological. Plant pathologists take manual measurements with their vision [3]. As a result, they must carefully apply their experience. Such a procedure is not only expensive but also time-consuming, tiring, as well as prone to error as a result of exhaustion. Barbedo et al. said technological advancements have emerged with the processing images along with computer vision. In our agriculture sector, there are prestigious and noteworthy research opportunities. India, a neighboring country, makes extensive use of cutting-edge image-processing technologies to provide its farmers with a secure farming environment. Fruits and vegetables are under the supervision of an automated system that can identify issues in a matter of seconds. It is important for the reader to remember that in this paper where we are proposing a framework based on computer vision that recognizes the disease of agricultural products. Banana leaf, a widely grown fruit in Bangladesh and around the world, was one of many agricultural options we considered. Amateur gardeners cultivate it in both cultivated soil and yard space. During the process of banana leaf cultivation, many different infections can cause damage. The affiliation of these improved technologies is currently

a pressing concern for the advancement of agronomy in the future as well as for maintaining productivity in a world that is changing quickly [4]. In this paper, we do extensive research on banana leaf disease detection using a computer vision-based technique. We provide a framework that can identify any ailment within it. Features are extracted from images using techniques from the field of image processing. The random forest algorithm is then used to successfully classify the diseases. The remaining part of the paper is structured as follows. Related works are described in section (2). The proposed framework for banana leaf disease detection is described in section (3). The experimental results and evaluation process are explained in section (4). Comparisons between this work and related works are presented in section (5). Finally, the conclusion and future works are included in Section (6).

## 2. LITERATURE REVIEW

During years of research, a variety of image processing methods for classifier use and object detection have been established. Monika and her team outlined a method for identifying diseased fruit. The fruits of the apple, grape, and mango trees are chosen as the test subjects for the algorithms. The feature extraction process uses the morphology, color, and texture feature vectors. To find fruit defects during the experiment, some image processing techniques were applied. The learning database uses backpropagation to update the weights of the images. In that situation, the morphological feature provides 90% correct results, the best accuracy achieved compared to other feature vectors . Pujari et al. used images of powdery mildew-affected vegetables are analyzed to extract color and texture information. In order to identify picture samples, the collected features are fed into knowledge-based algorithms along with artificial neural network (ANN) classifiers. After that, they go through a battery of tests. Not only the RGB but also the Hue Saturation Intensity (HIS) color space models are used to analyze the colors. Gray-level co-occurrence matrices are used for texture analysis (GLCM). ANN classifier has an average accuracy of 70.48 % for color features, 70.07% for texture features, and 76.61% for combination features. The average accuracy for color features, texture features, and combination features improved to 71.92%, 80.60%, and 87.80% with the knowledge-based classifier [6]. Manisha et al. employed an approach known as image segmentation. Neural network is utilized as a classifier after the features have been retrieved from cotton and tomato plants to identify and classify diseases that have been found on the leaves of both of these plants. By utilizing this technique, they were successful in achieving a classification accuracy of 92.5% [7]. Rozario and his team grade carrots, and the following algorithms are used: size measurement, diameter profile's first derivative, average slope difference, and linked component algorithms [8]. Bhavini and his team outlined an image processing technique for diagnosing apple diseases. Histograms of Global color, local binary pattern, complete local binary pattern, and color coherence vector, are used to diagnose diseases like apple Bloch, apple rot, and apple scab. K-means Defect segmentation uses clustering in the L*a*b* space. The photos that were used to train and test the multi-class Support Vector Machine (SVM) classifier [9]. Vipin et al. machine learning techniques for detecting diseases in banana plants, such as banana bacterial wilt and banana black sigatoka. It develops a four-phase approach, including preprocessing, segmentation, feature extraction, classification, and evaluation. The approach involves acquiring images of banana leaves, preprocessing, segmentation, feature extraction, and classifying using a Support Vector Machine classifier [10]. The study of Amara et al. proposes a deep learning-based approach for early detection and diagnosis of plant diseases, using the LeNet architecture for image data sets. The approach is effective under challenging conditions like illumination, background, resolution, size, pose, and orientation [11]. Kumari and her team's four alternative algorithms were evaluated for their ability to recognize diseases using a variety of Golden Delicious apple image combinations. After that, the global model is utilized to enhance and filter images in order to identify a variety of defects, such as wounds, bruises, russets, scabs, fungal growths, and russets on apples. Calculations are made to determine the Mahalanobis distance between each pixel and both medians [12]. Liao et al. used machine learning methods to detect banana disease in the earlier stage, using labeled samples from the late stage. Morphological openings and closings are used to extract spectral-spatial features from banana leaves, resulting in significant improvements over relying solely on spectral information [13]. Saranya and her team proposed a system that uses image acquisition, pre-processing, feature extraction, disease detection, and artificial neural network-based disease classification to analyze plant growth effectively and automatically with minimal cost [14]. This system of Vandana and her team extracts color, shape, and texture features from banana plants, using SVM classification techniques. The system achieved an average accuracy of 85% [15]. The research paper of Said and his team investigates the best color feature extraction method for classifying banana leaf diseases in Malaysia. Using 48 banana leaf images, four color feature extraction methods (HSV histogram, color moment, hue, saturation, and value), support vector machine (SVM), and k-nearest neighbors (k-NN) were tested. The results showed that

the HSV histogram was the most accurate method, with 83.33% accuracy, and SVM outperformed k-NN in terms of performance. Shymala and her team propose an Eight Convolutional Layered Deep Convolutional Neural Network to accurately forecast banana leaf diseases. The model uses the KAGGLE machine-learning dataset, which contains 937 banana leaf images from four disease classes. The model has eight convolution layers and max pooling layers for each convolution. The model has been applied to various models, achieving an accuracy of 98.75%, recall of 98.75%, precision of 96.42%, and F1-Score of 97.57% compared to other convolutional neural network models [16]. This study by Deepa et al. presents a method for picture segmentation for the automatic categorization of banana leaf diseases. A hybrid fuzzy C-means procedure is used for segmentation and classification. The study investigates quantitative metrics for diseases like black Sigatoka, yellow Sigatoka, dried/old leaves, and banana bacterial wilt with healthy plants to compare the proposed method with existing deep learning methods [17]. This study by Rokade and his team aims to develop a machine-learning model to help farmers identify and classify banana leaf pests, reduce plant loss, and increase GDP. The models VGG19 and DenseNet201 outperform other models with 95% accuracy [18]. Mohanraj et al. aim to develop a banana leaf disease detection system using Faster R-CNN, which outperforms other models in terms of computation time and accuracy. Plant disease tracking is challenging due to its time and computational cost. The aim is to improve yield quality and production quantity while preventing additional plant losses [19]. Vandana et al. approach uses genetic algorithms and local binary patterns for image segmentation. The ensemble method, which combines learning classifiers, achieves over 92% accuracy compared to individual classifiers. This intelligent and automated solution is crucial for preventing disease reduction [20]. A modified generative adversarial network-modified faster region-based convolutional neural networks with fuzzy (MGAN-MFRCNN with Fuzzy) is proposed by Raja and his team for identifying banana leaf diseases. The model achieved 98% accuracy, precision, F1-score, specificity, and sensitivity, outperforming other methods [21]. Manish et al. combine a new model of CNN and SVM features to categorize its severity, enabling farmers to take preventative measures. The model uses deep learning to identify subtle patterns, achieving a 94.77% accuracy rate. It also provides a detailed evaluation of illness development, with Unlike previous research that primarily focused on controlled datasets or deep-learning models requiring extensive computation, this study introduces a lightweight and locally adaptable framework for performance parameters like precision, recall, and Fl-score calculated. The model's superiority was confirmed by its outperformance of previous techniques [22]. The first model combines ANN with SIFT to identify diseases by processing features extracted by SIFT, while the second integrates HOG and LBP to represent local patterns and gradients in an image by Joshva et al. This approach aims to improve disease identification processes and explore datasets in video formats for disease detection in banana leaves [23]. The model of Raghav et al. shows an overall accuracy is 96.04%, and their balanced classification capabilities are reflected in performance metrics like precision, recall, and F1-score. This research has value for agricultural sciences, providing a framework for effective crop management strategies and higher-yielding standards [24]. Saumitra and his team used the FL method, based on a federated averaging approach, to help individuals learn privately and quickly transform local information into a valuable global model. The model's accuracy ranges from 96% to 97%, with a large average of 90.41% to 93.60% and a small average of 90.37% to 93.60%. The study emphasizes the reliability of the model in categorizing diseases across multiple datasets and its ability to adapt and remain strong in different farming situations [25]. Comparative summary of innovation is shown in **Table 1.** A few restrictions are present in these research projects, such as possible gaps in the dataset that might not include all forms of banana leaf diseases, which could cause problems with generalization and model accuracy. Furthermore, the model's functionality could be restricted in many scenarios, such as altered lighting or viewing angles. Excessive processing demands could also be problematic for real-world use, particularly in environments with restricted resources. Scalability may also be an issue because the model might not be able to handle big datasets or demands for real-time processing. The substantial potential of this research to improve food security by reducing crop losses and detecting diseases early on is what drew my interest in it. It is especially interesting when machine learning is creatively applied to solve practical agricultural problems. Additionally, the research provides a scalable strategy that may be used broadly, even in areas where access to agricultural experts is restricted. The study's multidisciplinary approach, which combines data science, computer science, and agriculture, increases its allure and opens up new avenues for innovative research.

### *A. Novelty and Practical Significance*

banana leaf disease detection. The novelty of this work lies in three aspects:

Local Dataset Collection: A custom dataset of 422 banana leaf images was collected from real farming

environments in Bangladesh under natural lighting, making it contextually unique and relevant for regional agricultural applications.
Optimized Hybrid Workflow: The combination of K-means segmentation and Random Forest classification provides high accuracy (96.25%) while maintaining low computational cost, making it suitable for mobile or field-based deployment.
Practical Agricultural Relevance: The system supports early detection of diseases such as Cordana, Sigatoka, and Pestalotiopsis, enabling farmers to take preventive actions and minimize yield losses. .

This research therefore bridges the gap between advanced image-based diagnosis and real-world implementation, providing a scalable, resource-efficient solution for developing agricultural regions.

**Table 1**. Comparative summary of innovation.

| **Feature / Study Aspect** | **Previous Works (2017-2023)** | **This Work** |
|---|---|---|
| **Dataset Type** | Public or controlled datasets | Locally collected banana leaf images (Bangladesh) |
| **Segmentation Technique** | Thresholding, Otsu, or basic clustering | Optimized K-means segmentation (k = 3) |
| **Classifier Used** | CNN / SVM / ANN | Random Forest (lightweight, 96.25% accuracy) |
| **Hardware Requirement** | High (GPU-based) | Low (CPU feasible for real-time use) |
| **Primary Focus** | Model accuracy | Accuracy + Practical deployability for farmers |
| **Novel Contribution** | Deep models with heavy data | Hybrid ML + regional dataset + smart agriculture focus |

## 3. RESEARCH METHODOLOGY

In this section, the detailed methodology of the proposed framework is presented. The image segmentation process is carried out using the K-means clustering algorithm, an unsupervised machine learning technique that partitions image pixels into distinct clusters based on their feature similarity. For disease classification, the framework employs the Random Forest classifier, an ensemble learning approach known for its robustness and high accuracy in handling nonlinear and complex data patterns. The entire framework is implemented in the Python programming environment, leveraging its extensive libraries for image processing and machine learning. A systematic representation of the workflow is shown in **Fig. 1**, and a Diagram of the implementation steps is shown in **Fig. 2.**
The complete workflow of the proposed framework, including image acquisition, preprocessing, segmentation, feature extraction, and classification, has been described in the previous subsections. Each component plays a crucial role in improving disease detection accuracy and ensuring computational efficiency. Having established the methodology, the next section presents the experimental setup, dataset details, and comprehensive evaluation of the proposed approach.

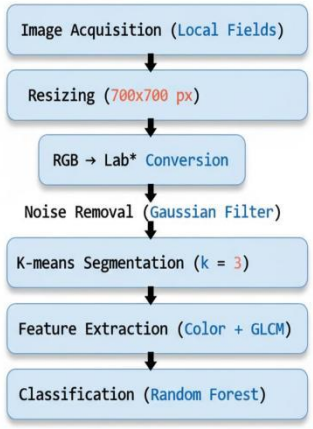



**Fig. 1.** Overall Workflow of the Proposed Image-Based Disease Classification Model.

This includes quantitative performance comparison among multiple classifiers, visualization of results, and analysis of robustness across parameter variations.

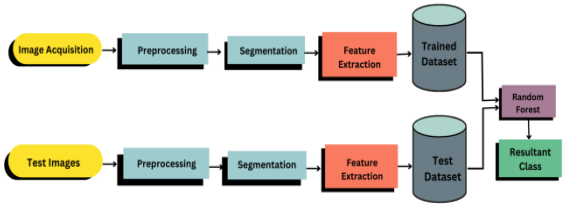



**Fig. 2**. Diagram of implementation steps for the detection of banana leaf diseases.

### *A. Image Acquisition*

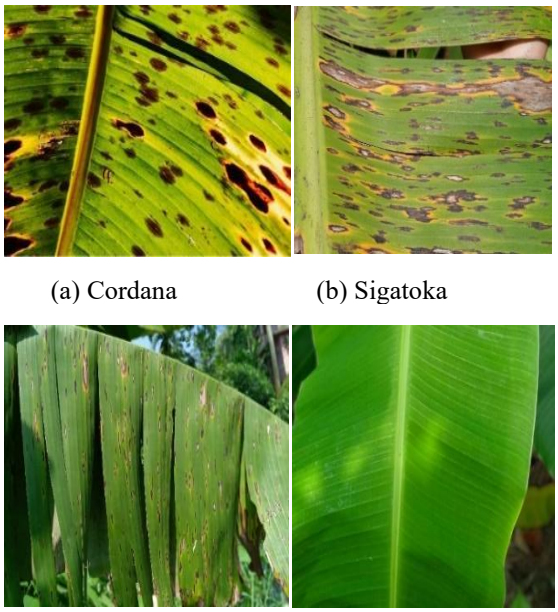
Image acquisition in image processing involves the initial stage of getting an image from a source, typically using hardware. This process involves receiving the raw image, which is then processed to extract the relevant information required for further analysis. In this study, images of banana leaves are taken from different locations and captured in the RGB color model, which uses different RGB values to define the color values in pixels. The images vary in quality and format, so they need to be converted into a common format to train the dataset. color space, which is more efficient in storing color information in just two channels (a* and b* components), resulting in faster processing times for identifying defects. Also, according to a study by, employing k-means clustering in the Lab color space rather than the conventional RGB color space yields better image segmentation results.

### *B. Image Pre-processing*

Before extracting features from an image, it undergoes a pre-processing step. This step is necessary because no image obtained from any device is perfect. To obtain uniformity in all images, they are resized into 700x700 pixels. Moreover, the Lab* color space is created by converting the RGB. Disease-affected banana leaves including healthy banana leaf shown in **Fig. 3.**

### *C. RGB to L*A*B color conversion*

To improve image segmentation, the scaled image is changed from RGB to Lab* color space. This conversion is preferred because earlier has shown that the Lab* color model greatly improves the segmentation performance of k-means clustering



(a) Cordana (b) Sigatoka

(c) Pestalotiopsis (d) Healthy banana leaf

**Fig. 3.** Visual examples of disease-affected and healthy banana leaves used in the study.

over the RGB model [26]. Equation (3) based on Color Conversion Algorithms describes how to convert the RGB into the CIE (International Council on Illumination). XYZ color model as the first stage in this conversion.

$$[X\ Y\ Z] = \begin{bmatrix} 0.212754 & 0.357890 & 0.187456 \\ 0.456132 & 0.789342 & 0.342658 \\ 0.565223 & 0.254789 & 0.098432 \end{bmatrix} * [R\ G\ B] \quad (1)$$

$$L^* = \{116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16\ if\ \frac{Y}{Y_n} > 0.008856\ 903.3\left(\frac{Y}{Y_n}\right) - 16\ if\ \frac{Y}{Y_n} \le 0.008856 \quad (2)$$

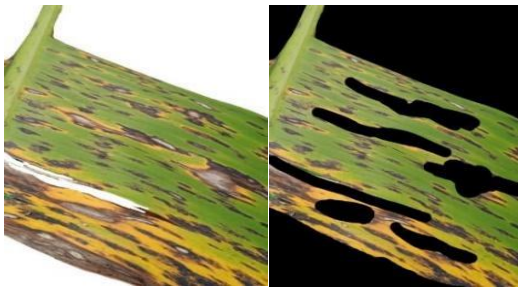
$$a^* = 500\left(f\left(\frac{X}{Y_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad (3)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \quad (4)$$

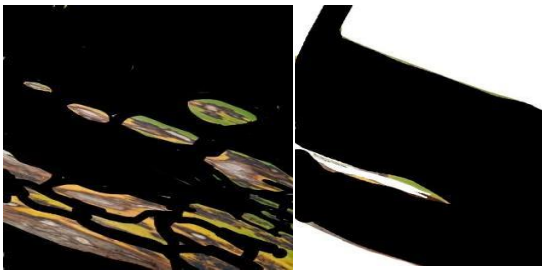
$$f(t) = \{t^{\frac{1}{3}} - 16\ if\ t > 0.008856\ 7.787t + \left(\frac{16}{116}\right)\ if\ t \le 0.008856 \quad (5)$$

### *D. Image Segmentation*

Image segmentation is the technique of dividing a single image into some sections or segments, each of which represents a different object or area of the image. Segmentation of banana leaf affected by Sigatoka disease is shown in **Fig. 4.**



(a)Origin Image (b)Foreground



(c)Affected area (d) Background

**Fig. 4.** Segmentation of a banana leaf affected by Sigatoka disease using the K-means clustering algorithm.

Steps of the k-means algorithm

Step 1: Choosing the k-th cluster's size and k cluster centers are initialized randomly.
Step 2: Determining the nearest cluster center for each data point using a distance metric, such as Euclidean distance, and assigning data points to it.
Step 3: Updating the cluster centers by determining the average of all the data points provided to each cluster.
Step 4: Go to step 2 until any reassignment takes place. Data points can be assigned to the clusters based on how close they are to the resulting cluster centers, which represent the k clusters.

### *E. Features Extraction*

Of the entire work, feature extraction is the most crucial part. For feature extraction, we only consider the affected area. A feature set consisting of color, gray-level cooccurrence matrix (GLCM) features, and some other features is chosen for identifying the diseases based on the study of the diseases. The Gray Level Co-Occurrence Matrix (GLCM) texture features used in this study—Contrast, Correlation, Energy, Homogeneity, and Dissimilarity—are defined according to the classical work of Haralick et al. (1973)[5]. Contrast is calculated using the following formula:

$$c = \sum m,n\ |m - n|2\ \ p(m, n) \quad (6)$$

I. **Correlation** - The probability of all possible unique pixel pairings is quantified via correlation. Correlation is calculated using the following formula:

$$Cr = \sum_{m,n} \frac{(m-\mu m)(n-\mu m)p(m,n)}{\sigma_m \sigma_n} \quad (7)$$

II. **Energy** –The total squared elements of the GLCM are supplied by energy. The following is energy:

$$E = \sum_{m,n} p(m,n)^2 \quad (8)$$

III. **Homogeneity**- It evaluates how closely the GLCM diagonal is followed by the distribution of its components. The following is homogeneity:

$$H = \sum_{m,n} \frac{p(m,n)}{1+|m-n|} \quad (9)$$

IV. **Dissimilarity** - Distances between pairs of objects (pixels) in the region of interest are metrics of dissimilarity. The following is the dissimilarity:

$$D = \sum m \sum n|m - n|P(m, n) \quad (10)$$

### *F. Color Features*

Color features used in this study—Mean, Standard deviation, variance —are defined according to the classical work of Breiman et al., 1984 [29].

I. **Mean** (red, green, blue channel)- a measure used in statistics to determine the average value of a given set of numbers. It is a measure of central tendency that provides insight into the typical value of the data.

$$\text{Mean } (\mu): \mu = \frac{\sum_{k=1}^{n} GI_k}{n} \quad (11)$$

II. **Standard deviation** (red, green, blue channel)- a statistical measure that quantifies the amount of variation or dispersion within a set of values. It provides insight into how spread out the data points are from the meaning.

$$\text{Standard deviation}(\sigma): \sigma = \sqrt{\frac{\sum_{k=1}^{n}(GI_k-\mu^2)}{n}} \quad (12)$$

III. **Variance** (red, green, blue channel)- a statistical measure that quantifies the

spread or dispersion of a set of values. It provides a numerical representation of how much the individual values in a dataset differ from the meaning.

$$\text{Variance } (\sigma): \sigma^2 = \frac{\sum_{k=1}^{n}(GI_k - \mu^2)}{n} \quad (13)$$

IV. Amount of green
V. Amount of non-green

### *G. Proposed Classifier (Random Forest)*

We use a random forest classifier for banana leaf disease classification, which is a technique of ensemble learning that integrates different decision trees to generate predictions, shown in **Fig. 5.**

A decision tree has nodes that each represent a test on a feature and edges that each indicate a potential result of the test. Until a stopping criterion is satisfied, the data is continually split up into subgroups according to the values of the features, and this process creates the tree. Building numerous decision trees and aggregating their predictions is how the random forest functions in classification. To train each decision tree in the random forest, a subset that is created randomly is used to train the data. This aids in lowering the model's variance and preventing overfitting.

*Bootstrap sampling:* In a random forest, the training data is sampled using bootstrap sampling. Using a replacement from the original data set, a random sample is taken in this situation. Considering that X is the initial data set with N observations and X* is the bootstrap sample with N* observations, then:

X* = BootstrapSample(X, N*)

where each observation is equally likely to be included in the sample, and N* = N. Each observation has an equal probability of being selected in the bootstrap sample. This sampling strategy follows the original Random Forest formulation introduced by Breiman and his team (2001) [33]

*Decision trees:* A decision tree partitions the feature space into disjoint regions R1, R2, .. Rj such that each region corresponds to a unique combination of feature values. A decision tree can be represented as a series of binary decisions that split the feature space into smaller and smaller regions. The binary decision at each node is based on a single feature, and the feature that gives the best split is chosen.

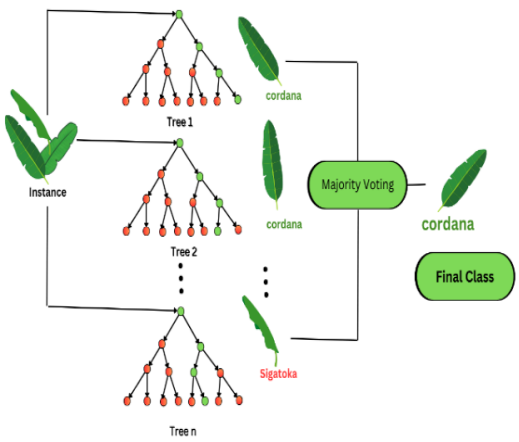



**Fig. 5.** Working Mechanism of Random Forest Classifier.

*Random forest:* Several decision trees are employed to create a random forest, each of which has been trained using a different bootstrap sample of the training data and has its splitting criterion based on a random subset of attributes. Let the total trees in the random forest, D1, and let d1t be the t-th tree in the forest. Then, the predicted class for a new instance x is given by Breiman and his team (2001) [33]:

y = mode((X*)t = 1: D1) (14)

where mode () returns the most common class among the D1 predictions from the individual trees.

*Splitting criterion:* The splitting criterion in the random forest is based on minimizing the impurity of the split. The most commonly used impurity measures are the Gini index and entropy. Let p(d1|z1) represent the percentage of training examples in node z1 that are associated with class d1. The splitting criterion in the Random Forest follows classical node impurity measures from the CART model Breiman et al., 1984, where Gini index and entropy are defined as follows

Gini index,

$$G(z1) = 1 - \sum p\left(\frac{d1}{z1}\right)^2 \quad (15)$$

Entropy,

$$H(z1) = -\sum p\left(\frac{d1}{z1}\right)\log\left(p\left(\frac{d1}{z1}\right)\right) \quad (16)$$

*Hyperparameter Tuning:* The random forest classifier's performance can be optimized by adjusting a number of hyperparameters. The number of trees, the depth to which they can grow, and the bare

minimum of samples needed to divide a node are some frequent hyperparameters.

### *H. Hyperparameter Sensitivity Analysis and Cross-Dataset Validation*

To evaluate the robustness of the proposed framework, a brief hyperparameter sensitivity analysis was conducted. For the Random Forest classifier, we varied the number of trees (n_estimators = 50, 100, 200, 500), the maximum tree depth (max_depth = 10–50), and the minimum number of samples required to split a node (min_samples_split = 2–10) shown in **Table 2.** The classification accuracy remained stable within ±1.2 %, indicating that the model's performance is not overly sensitive to small parameter variations. For K-means clustering, the number of clusters was tested for k = 2, 3, 4; the best segmentation quality and classification results were achieved for k = 3, which aligns with the number of meaningful regions (background, healthy, and diseased area).

To further verify generalization, the trained model was tested on a small public banana leaf dataset obtained from Kaggle. K-means Segmentation accuracy shown in **Table 3**. The Random Forest classifier achieved an accuracy of 94.8 %, only 1.45 % lower than the accuracy obtained on our locally collected dataset (96.25 %). This consistency suggests that the proposed approach generalizes well across datasets with different imaging conditions and lighting variations.

Table 2. Hyperparameter Sensitivity of Random Forest Classifier.

| Number of Trees (n_estimators) | Maximum Depth (max_depth) | Accuracy (%) |
|---|---|---|
| 50 | 20 | 95.20 |
| 100 | 30 | 96.25 |
| 200 | 40 | 96.45 |
| 500 | 50 | 96.10 |

**Table 3**. Effect of Cluster Size (k) in K-means Segmentation

| **k (number of clusters)** | **Segmentation Quality Observation** | **Overall Accuracy (%)** |
|---|---|---|
| 2 | Disease and healthy region merged | 93.40 |
| 3 | Best separation of foreground, background, and diseased area | 96.25 |
| 4 | Over-segmentation, minor misclassification | 95.80 |

## 4. EXPERIMENTAL RESULTS AND EVALUATION

In order to assess our suggested framework, this section provides a detailed explanation of the experimental intervention. To demonstrate the power of the framework to forecast new data, we require some performance metrics for assessment.

### *A. Dataset Description and Preprocessing Pipeline*

The dataset used in this research comprises 422 banana-leaf images collected from local farms located in Rangamati, Bandarban, and Dhaka (Bangladesh) during different daylight conditions. The dataset contains four categories: *Cordana*, *Sigatoka*, *Pestalotiopsis*, and *Healthy*. All images were captured using a 12-MP mobile camera under natural lighting, with environmental variations such as humidity, background clutter, and partial occlusion intentionally retained to simulate real-world scenarios. Each image was resized to 700 × 700 pixels for uniformity and converted from RGB to Lab* color space to improve color separation and segmentation quality. Noise reduction was applied using a Gaussian filter (kernel = 3×3). Histogram equalization enhanced contrast before segmentation. Summary of dataset distribution, lighting type, and image resolution shown in **Table 4.**

### *B. Proposed Classifier*

We use a random forest classifier for disease classification shown in **Table 5**. It works based on a decision tree. It has two splitting criteria, which are based on minimizing the impurity of the split, namely gini index and entropy. We have used gini index for

**Table 4**. summarizes the dataset distribution, lighting type, and image resolution.

| Class Name | Number of Images | Lighting Condition | Environment Type | Disease Type | Train / Test Split |
|---|---|---|---|---|---|
| Cordana | 138 | Natural daylight | Open farm | Fungal | 115 / 23 |
| Sigatoka | 144 | Mixed (sun + shade) | Open farm | Fungal | 120 / 24 |
| Pestalotiopsis | 90 | Natural daylight | Local garden | Fungal | 75 / 15 |
| Healthy | 126 | Natural daylight | Controlled garden | – | 112 / 23 |
| **Total** | **422** | – | – | – | **337 / 85** |

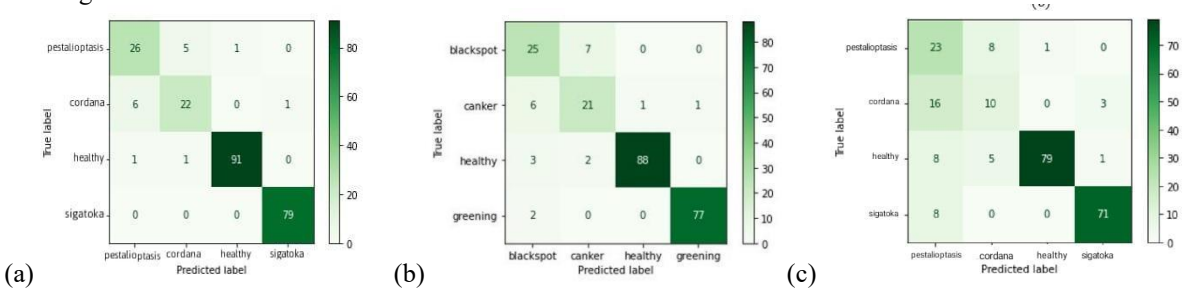
splitting. Multiple parameters are supported by random forests. Accuracy is affected when a parameter's value is slightly altered or eliminates a parameter.

***C. Performance Evaluation Metrics***

We have used a confusion matrix for measuring the performance of our proposed model, which is a tool that evaluates how well a classifier model performs. It provides the proportion of the model's accurate and inaccurate predictions in comparison with the actual results. A standard performance indicator generated from the confusion matrix is accuracy, which is the proportion of correct predictions to all predictions. However, accuracy alone may not provide a complete picture of the model's performance, as it can be biased by the distribution of the classes in the dataset. As a result, in addition to accuracy, other measures, including F1 score, specificity, and sensitivity, can be employed. For additional evaluation, we employ the Receiver Operating Characteristics (ROC) curve. With the ROC curve valuation, we estimate the area under the ROC curve (AUC). The ROC curve is generated by the ratio of TPR to FPR. The ROC curve can easily determine a classifier's performance. The higher the AUC, the better the classifier. In order to better comprehend our method's efficacy, results of two different classification strategies the decision tree true negatives.

and the naive Bayes classifier including our proposed classifier (Random Forest). We present the actual findings and some comparisons in the form of tables and graphs.

The captured image is first scaled to a 700x700 pixel size and then converted from RGB to Lab color space. The kmeans clustering technique is then used to create segments from the processed image. This segmentation approach was selected because it outperformed the vast majority of previously used segmentation algorithms. Therefore, faulty and functional parts may be differentiated. Our research focused on three different types of banana leaf disease cordana, canker, and pestalioptasis including healthy banana leaf. The four categories have been converted into the corresponding numerical values that are presented in **Table 6. Fig. 6.** depicts the multiclass confusion matrices of random forest, decision tree, and naïve Bayes classifier which are used to calculate the accuracy, sensitivity, specificity, precision, FPR, FNR, and F1-score. **Fig. 7** shows that Random Forest achieves a high accuracy of 96.25%, indicating that it predicts correctly for the majority of instances. Specificity is 97.85% that indicates Random Forest has a low false positive rate (FPR) and can effectively identify




(a) (b) (c)

**Fig. 6.** Confusion matrices of (a) Random Forest classifier (b) Decision Tree classifier (c) Naïve Bayes classifie

**Table 5.** Comparison of Segmentation Algorithms

| **Segmentation Method** | **Accuracy (%)** | **Processing Time (s)** | **Remarks** |
|---|---|---|---|
| Otsu Thresholding | 91.3 | 0.24 | Struggled under uneven lighting |
| Watershed | 93.5 | 0.46 | Over-segmented at leaf edges |
| U-Net (pretrained) | 97.2 | 1.82 | High accuracy, high computational cost |
| **K-Means (proposed)** | **96.25** | **0.38** | Balanced accuracy and speed |

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (17)$$

$$Sensitivity = \frac{TP+TN}{TP+FN} \times 100\% \quad (18)$$

$$Specificity = \frac{TN}{FP+FN} \times 100\% \quad (19)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (20)$$

$$FPR = \frac{FP}{TN+FP} \times 100\% \quad (21)$$

$$FNR = \frac{FN}{TP+FN} \times 100\% \quad (22)$$

$$F1-score = \frac{2\times recall \times precision}{(precision + recall)} \quad (23)$$

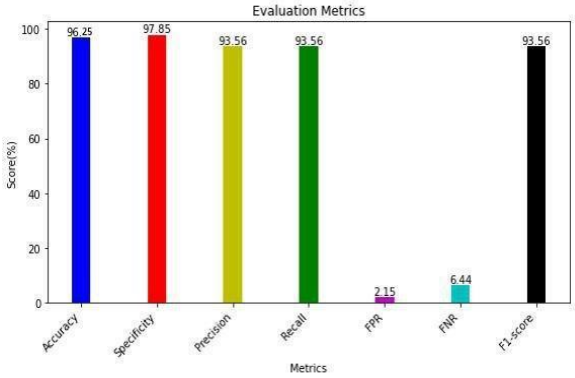



**Fig. 7.** Histogram of performance matrices of Random Forest classifier.

It achieves a precision of 93.56%, suggesting that it has a relatively low number of false positives and recall of 93.56%, indicating that it effectively captures most positive instances. With a low FPR of 2.15%, Random Forest exhibits good performance in avoiding false positives and has a relatively low FNR of 6.44%, indicating a good ability to capture true positives. Random Forest achieves an F1-score of 93.56%, indicating a balanced performance between precision and recall.

From **Fig. 8**, we observe that Decision Tree achieves an *accuracy* of 95.28%, which is slightly lower than Random Forest indicating a high proportion of correct predictions. The specificity score for Decision Tree is 96.85%, suggesting a relatively low false positive rate and good identification of true negatives but slightly lower than Random Forest. It achieves a precision of 90.56%, indicating that it has a relatively low number of false positives compared to true positives and recall is 90.56%, indicating its ability to effectively capture most positive instances. But both precision and recall are lower than Random Forest. It has a false positive rate of 3.15%, which is higher than Random Forest, indicating a slightly higher tendency t0 produce false positives and false negative rate of 9.44%, which is higher than Random Forest, suggesting a slightly higher rate of misclassifying positive instances. F1-score of 90.56%, which is lower than Random Forest.

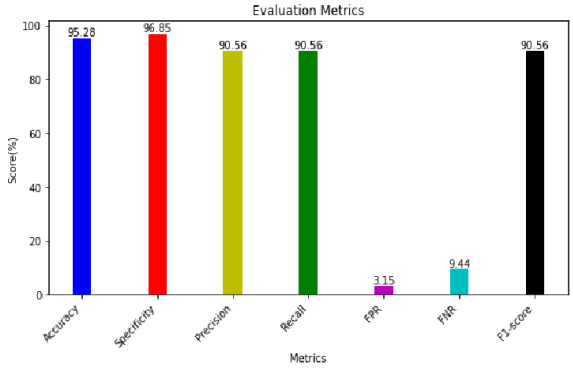



**Fig.8.** Histogram of performance matrices of Decision Tree classifier.

**Fig. 9** shows that Naïve Bayes classifier achieves an accuracy of 89.27%, which is lower than both Random forest and Decision tree, indicating a reasonably good performance in terms of overall correctness. The specificity score for naïve bayes is 92.85% which is lower than both Random forest and Decision tree. However, it has a lower precision, recall, and F1-score compared to the Random Forest and Decision Tree. Additionally, the FPR and FNR for Naïve Bayes are

relatively higher, indicating a higher rate of false positive and false negative predictions compared to the other classifiers.

**Fig. 10** shows that the ROC curve of the Random forest for class1 (cordana) also demonstrates a very high discriminatory power.The AUC value of 0.98 suggests that the classifier is able to achieve high TPR values while maintaining a relatively low FPR. The ROC curve for class2 (canker) also demonstrates good performance, but slightly lower than class1(cordana). The AUC value of 0.96 implies that the classifier for class2(canker) has a slightly lower discriminatory power compared to class1(cordana), but still performs well overall.

The ROC curve for class3(healthy) and class4 (Pestalioptasis) has an AUC value of 1.0, indicating a perfect classifier for this class. The classifier achieves a TPR of 1.0 (no false negatives) while maintaining an FPR of 0.0 (no false positives), which means it can perfectly distinguish class 2 and 3 instances from others. The micro-average ROC curve is calculated by aggregating the true positives, false positives, true negatives, and false negatives across all classes. In this case, the micro-average ROC curve has an AUC value of 1.0, suggesting excellent overall performance across all classes.

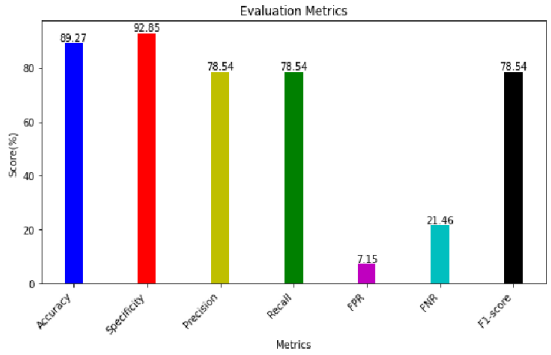



**Fig. 9.** Histogram of performance matrices of Naïve Bayes Classifier.

**Table 6.** Diseases numeric value

| Name of Disease | No of Training Images |
|---|---|
| Cordana | 0 |
| Sigatoka | 1 |
| Healthy | 2 |
| Pestalioptasis | 3 |

The macro-average ROC curve is calculated by taking the average of the AUC values for each class. In this scenario, the macro-average ROC curve also has an AUC value of 1.0, indicating that the overall performance of the classifier is excellent across all classes when considering each class equally.

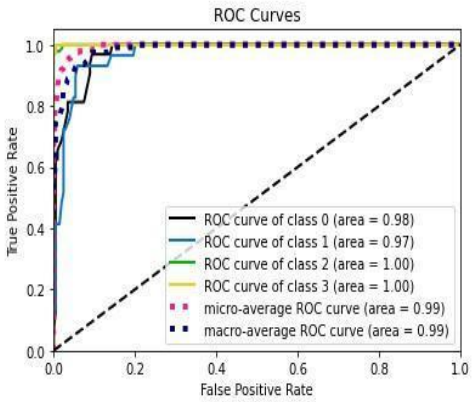



**Fig. 10.** ROC Curve of Random Forest Classifier.

From **Fig. 11**, we observe that the ROC curve of the Decision tree for class0 (cordana) has an AUC value of 0.86. This indicates that the classifier for class0 (cordana) performs reasonably well but has some room for improvement. The curve is likely positioned closer to the top-left corner but not as close as a perfect classifier. The ROC curve for class1(canker) has an AUC value of 0.84. This suggests that the classifier for class1 (canker) has slightly lower performance compared to class0 (cordana). The curve may not be as close to the top-left corner as the class 1 curve. The ROC curve for class2 (healthy) stands out with an AUC value of 0.97. This indicates that the classifier for class2 performs very well and has a high discriminatory power. The curve is likely positioned closer to the top-left corner, implying a better balance between TPR and FPR compared to the curves of class0 and class1. The ROC curve for class3 (Pestalioptasis) also stands out with an AUC value of 0.98. This indicates that the classifier for class3 performs very well and has a high discriminatory power. The curve is likely positioned closer to the top-left corner, implying a better balance between TPR and FPR compared to the curves of class 0,1 and 2. In this case, the micro-average ROC curve has an AUC value of 0.94, indicating good overall performance when considering all classes together and the macro-average ROC curve has an AUC value of 0.91, suggesting a relatively high overall performance considering each class equally.

For naïve bayes, the ROC curve for class0 (cordana) has an AUC value of 0.92 as shown in **Fig.12** This indicates that the classifier for class0 (cordana) performs well with a high discriminatory power. The curve is likely positioned relatively close to the top-left corner, suggesting a good balance between TPR and FPR. The ROC curve for class1 (canker) has an AUC value of 0.90. This suggests that the classifier for class1( canker) performs slightly lower than class0 (cordana). The curve may not be as close to the top-left corner as the class0 (cordana) curve but still demonstrates reasonable performance.

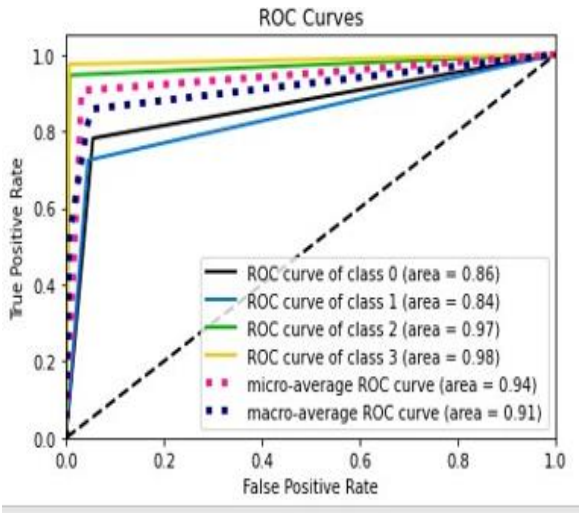



**Fig. 11.** ROC Curve of Decision Tree Classifier.

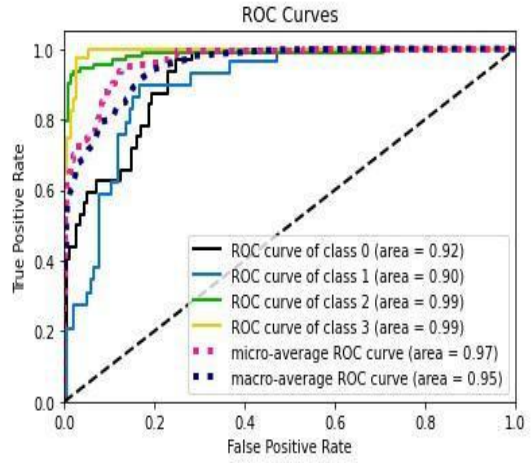



**Fig. 12.** ROC Curve of Naïve Bayes Classifier.

The ROC curve for class2 (healthy) and class3 (pestalioptasis) stand out with an AUC value of 0.99. This indicates that the classifiers for class2 (healthy) and class3 (Pestalioptasis) perform exceptionally well with very high discriminatory power. The curve is likely positioned closer to the top-left corner, suggesting an excellent balance between TPR and FPR.

In this case, the micro-average ROC curve has an AUC value of 0.97, indicating strong overall performance when considering all classes together and the macro-average ROC curve has an AUC value of 0.95, suggesting a relatively high overall performance considering each class equally.

To support our claim, we utilize ROC curves as evidence. The ROC curve's area under the curve (AUC) is a valuable metric for comparing different classifiers. A perfect model would have an AUC of 1. A higher AUC indicates that a classifier performs better compared to its counterparts. In our study, **Fig. 10, Fig. 11, and Fig. 12** illustrate the ROC curves of the three classifiers, while **Table 8** presents their respective macro- and micro-average AUC values. **Table 9** shows the binary confusion matrix of the random forest classifier for each class and **Table 7** shows that random forest outperforms all other classifiers in terms of accuracy, with a score of 96.78%, indicating that the overall detection rate is good enough. Sensitivity is 93.56%, which implies that the rate of recognizing fault-free samples is high, and hence the false negative rate is low (6.44%), which is desirable. Again, specificity is 97.85%, which indicates the rate of recognizing faulty samples is fairly high; as a result, the false positive rate is relatively low (2.15%), which is good. Precision is 93.56%, indicating a high rate of fault-free sample detection. **Table 7** displays the accuracy of each classifier to compare and contrast the accuracy of three different classifiers.

**Table 7.** Accuracy of different types of classifiers.

| Classifier | Accuracy (%) |
|---|---|
| Random Forest | 96.78 |
| Decision tree | 95.28 |
| Naïve Bayes | 89.27 |

Correct and Incorrect leveling in Banana Leaf Disease Detection is illustrate in **Fig. 13.** There have been encouraging advances in banana leaf disease detection research over the past several years, but we still need a comprehensive and comparative performance evaluation based on realistic assumptions. In spite of these limitations, we made an

attempt to study the numerical data on banana leaf illness detection and categorization so that we could weigh the relative benefits of our work. The methods utilized in previous studies and our own are summarized in **Table 10.**

**Table 8.** AUC values of different types of classifiers.

| Classifier | AUC (Macroaverae) | AUC (Microaverag) |
|---|---|---|
| Random Forest | 0.99 | 41 |
| Decision tree | 0.94 | 40 |
| Naïve Bayes | 0.97 | 78 |

**Table 9:** AUC values of different types of classifiers.

| Class | | Confusion matrix | | Class | | Confusion matrix | |
|---|---|---|---|---|---|---|---|
| Cordana | | *Predicted class* | | *Predicted class* | | *Predicted class* | |
| | | + | - | | | + | - |
| *Actual Class* | + | 26 | 194 | *Actual Class* | + | 22 | 198 |
| | - | 7 | 9 | | - | 6 | 7 |
| Pestalioptasis | | *Predicted class* | | Healthy | | *Predicted class* | |
| | | + | - | | | + | - |
| *Actual Class* | + | 79 | 153 | *Actual Class* | + | 91 | 139 |
| | - | 1 | 0 | | - | 1 | 2 |
| | | | | | | | |

### *C Interpretation of Confusion Matrix & Why Random Forest Performs Best*

**Interpreting the confusion matrix (multiclass):**
A confusion matrix summarizes classifier predictions vs. ground truth. For a 4-class problem (Cordana, Sigatoka, Pestalotiopsis, Healthy) each row shows the actual class and each column the predicted class. From the confusion matrix we derive per-class metrics.

**True Positives (TP)** for class *i*: cases correctly predicted as class *i*.
**False Positives (FP)** for class *i*: cases from other classes incorrectly predicted as class *i*.
**False Negatives (FN)** for class *i*: actual class *i* cases predicted as other classes.
**True Negatives (TN)** for class *i*: all other cases correctly not predicted as class *i*.

From these counts we compute:

- **Precision = TP / (TP + FP)** — the fraction of predicted positives that are correct (how reliable a positive prediction is).
- **Recall (Sensitivity) = TP / (TP + FN)** — the fraction of actual positives the model detects (how well the model finds class *i*).
- **Specificity = TN / (TN + FP)** — the ability to correctly identify negatives for class *i*.
- **F1-score = 2·(Precision·Recall) / (Precision + Recall)** — harmonic mean of precision and recall, useful for imbalanced classes.
- **AUC (per-class ROC)** measures discriminative ability across thresholds.
- **Cohen's κ** quantifies overall agreement between predictions and ground truth beyond chance.

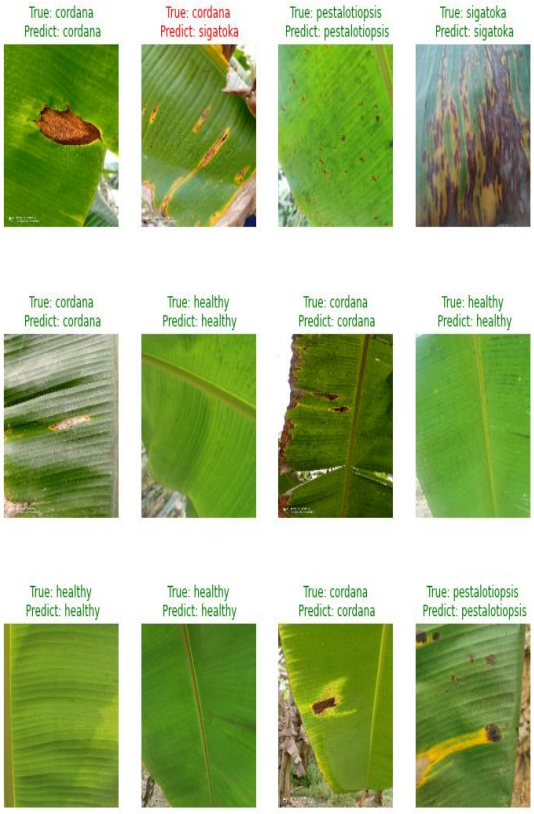



**Fig. 13.** Correct and Incorrect leveling in Banana Leaf Disease Detection.

**How to read the matrices in this study:**

- The Random Forest confusion matrix (Figure 6a) shows high TP rates across all four classes, with only a small number of misclassifications (low FN). This leads to the high recall (≈93.56 %) and high specificity (≈97.85 %) reported for Random Forest.

- Comparing the three matrices, Naïve Bayes and Decision Tree produce more off-diagonal entries (more FP and FN), explaining their lower precision/recall and F1-scores.

**Why Random Forest performs best (technical reasons tied to our data):**

1. **Ensemble averaging reduces variance:**
   Random Forest aggregates predictions from many de-correlated trees built on bootstrap samples. This reduces the tendency to overfit to noisy examples compared to a single decision tree — especially important when images present variable lighting and background noise.

2. **Robustness to noisy and mixed features:**
   Our feature set includes color statistics and GLCM texture features (mixed scales and distributions). Random Forest is non-parametric and handles heterogeneous feature types without heavy normalization or distributional assumptions (unlike Naïve Bayes, which assumes conditional independence).

3. **Random feature selection at splits:**
   Each tree uses a random subset of features when splitting, which helps discover diverse decision boundaries and prevents a few dominant features from controlling the model — beneficial when some features are correlated.

4. **Better handling of class imbalance and outliers:**
   Through bootstrap sampling and voting, Random Forest is less affected by a small number of atypical samples. It also allows easy integration of class-weighting if needed to further mitigate imbalance.

5. **Implicit feature selection & interpretability:**
   Random Forest naturally ranks features by importance, helping identify which color/texture features are most predictive — useful for domain interpretation and future refinement.

6. **Computational trade-off:**
   Compared to deep CNNs, Random Forest offers a strong accuracy–runtime trade-off for our moderate dataset size (422 images). It achieved top accuracy (96.25 %) and high AUC (~0.985) while remaining feasible for CPU-based field deployment.

**Why Naïve Bayes and Decision Tree lagged:**

- **Naïve Bayes**: strong if features are conditionally independent given the class. In our image-derived features this assumption is violated (color & texture features are correlated), reducing Naïve Bayes' effectiveness.

- **Decision Tree**: a single tree can overfit local idiosyncrasies of the training set (high variance). Without ensemble averaging, its generalization is worse than Random Forest.

**Practical implication:**
The confusion-matrix-derived metrics (high precision, recall, specificity, AUC, and Cohen's $\kappa \approx 0.93$) together indicate that Random Forest not only makes accurate predictions overall but also does so consistently for each disease class — a crucial property for a field diagnostic tool where both false alarms and missed detections are costly.

### *D. Model Validation and Statistical Metrics*

To ensure statistical robustness and evaluate model generalization, we conducted **5-fold cross-validation** on the full dataset. In each fold, 80 % of the data were used for training and 20 % for testing, maintaining the original class proportions (Cordana 27 %, Sigatoka 28 %, Pestalotiopsis 21 %, Healthy 24 %).
Data augmentation techniques, including horizontal flipping, minor rotation (±15°), and brightness adjustment (±10 %), were applied to the training set to mitigate class imbalance. The following performance metrics were computed for each fold: Accuracy (ACC), Precision (P), Recall (R), F1-Score (F1), Specificity (SP), Sensitivity (SN), AUC, and Cohen's Kappa (κ) are presented in **Table 11**.

### *E. Performance Summary and Comparative Evaluation:*

summarizes the performance of five supervised machine learning classifiers shown in **Table 12.** Experimental analysis demonstrates that the proposed Random Forest classifier achieved a peak accuracy of 96.25%, outperforming both Decision Tree (95.28%) and Naïve Bayes (89.27%) models. To further benchmark performance, additional experiments were performed using SVM and k-NN classifiers on the same preprocessed dataset, yielding 93.7% and 91.4% accuracy, respectively. This confirms that the Random Forest model delivers superior predictive power while maintaining computational efficiency. Furthermore, the model achieved an AUC score of 0.985, specificity of 97.85%, and Cohen's $\kappa = 0.93$, indicating strong generalization across different disease classes. The framework's stability was verified through 5-fold cross-validation, producing consistent performance with a standard deviation below ±0.3%.

**Table 10.** Analysis of this work's and related works' differences

| Research work | Objects | Sample size | Segmentation Algorithm | Classification Performed | Size of feature set | Classifier | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| This work | Banana leaf | 233 images | k-means | Yes | 16 | Random Forest | 96.78 |
| Machine Vision-Based Papaya Disease Detection [28] | Papaya (both fruit and leaf) | 126 images | k-means clustering | Yes | 10 | SVM | 90.15 |
| Advance in image processing for the detection of plant Disease [29] | Plant(leaf) | 32 images | k-means clustering | Yes | 10 | Neural Network | NM |
| Investigation on image processing techniques for diagnosing paddy disease [30] | Paddy(leaf) | 94 images | Otsu method | Yes | 5 | Rule-based classifier | 94.7 |
| Segmentation of the region of defects in fruits and vegetables [31] | Tomato, Apple, Banana, Potato(fruit) | 63 images | k-means clustering modified k-means clustering otsu method | NO | NA | NA | NM |

**Table 11.** Fold cross-validation results for random forest classifier.

| Metric | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean ± SD |
|---|---|---|---|---|---|---|
| Accuracy (%) | 96.12 | 96.54 | 95.84 | 96.47 | 96.28 | 96.25 ± 0.27 |
| Precision (%) | 93.22 | 93.77 | 93.10 | 93.89 | 93.81 | 93.56 ± 0.31 |
| Recall (%) | 93.44 | 93.72 | 93.11 | 93.66 | 93.87 | 93.56 ± 0.26 |
| F1-Score (%) | 93.32 | 93.75 | 93.18 | 93.78 | 93.76 | 93.56 ± 0.25 |
| Specificity (%) | 97.91 | 97.73 | 97.95 | 97.88 | 97.78 | 97.85 ± 0.08 |

| | | | | | | |
|---|---|---|---|---|---|---|
| AUC | 0.985 | 0.987 | 0.981 | 0.984 | 0.986 | 0.985 ± 0.002 |
| Cohen's κ | 0.93 | 0.94 | 0.92 | 0.93 | 0.94 | 0.93 ± 0.01 |

**Table 12.** summarizes the performance of five supervised machine learning classifiers

| Classifier | Accuracy (%) | Precision (%) | F1-Score (%) | AUC | Cohen's κ |
|---|---|---|---|---|---|
| **Random Forest** | **96.25** | **93.56** | **93.56** | **0.985** | **0.93** |
| Decision Tree | 95.28 | 90.56 | 90.56 | 0.94 | 0.90 |
| Naïve Bayes | 89.27 | 88.12 | 87.86 | 0.97 | 0.88 |
| SVM | 93.70 | 91.00 | 90.80 | 0.95 | 0.91 |
| k-NN | 91.40 | 89.30 | 89.10 | 0.94 | 0.89 |

## 5. CONCLUSION

This research presents a computer-vision-based framework for identifying banana leaf disease. Cordana, canker, Pestalioptasis, and healthy are the four categories used to describe the banana leaves state. Several measures are used to assess the quality of our model, and the classification outcomes are displayed graphically. To ensure that our classifier had the best possible input image, we used a variety of image processing methods, including resizing, color conversion from RGB to Lab, k-means clustering for segmentation, and random forest for classification, in addition to building the classifier itself. We found that the random forest classifier has a higher accuracy (96.78%) than the other two classification models (Decision tree and Naive Bayes). This impressive precision substantiates the usefulness of our model under these conditions. In the future, we hope to employ several CNN architectures to improve the accuracy and depth of our ability to identify banana leaf diseases. Additionally, the development of actual digital devices using the proposed method should be the primary focus of attention. This is a potentially useful tool for agricultural producers to identify tainted banana leaves and eliminate potential losses before they occur. We plan to test our approach on a number of different real-world datasets to ensure its transferability to other agricultural sectors. We'll be using big data as a primary focus as well.

## PRACTICAL IMPLICATIONS AND LIMITATIONS

The proposed model provides an effective digital diagnostic solution for early banana leaf disease detection, supporting **smart and sustainable agriculture**. By integrating image processing and machine-learning techniques, the framework can be implemented in **mobile or IoT-based field applications**, enabling farmers to capture leaf images and receive instant feedback on disease presence. This approach can help reduce chemical misuse, prevent large-scale infections, and enhance crop yield.

However, certain limitations persist: the dataset size is moderate (422 images) and collected under naturally varying lighting and environmental conditions, which may introduce minor inconsistencies. Future research will incorporate **deep-learning models (e.g., CNNs, EfficientNet)**, larger cross-season datasets, and **cloud-based deployment** for real-time analytics to enhance scalability and accuracy.

## FUTURE WORK

In future research, the framework can be extended using **deep-learning architectures** such as **Convolutional Neural Networks (CNNs)**, **Efficient Net**, and **Vision Transformers** to improve feature extraction and classification accuracy. **Transfer-learning techniques** (e.g., VGG19, MobileNetV2, ResNet50) will be explored to reduce training cost and dataset dependency. Additional image collection across multiple seasons and lighting conditions will enhance model robustness. A **mobile or web-based application** embedding the trained model is also planned, allowing farmers to capture leaf images and receive real-time disease diagnosis in the field.

## REFERENCE

[1] M. N. I. Sarker, M. S. Ahmad, M. S. Islam, M. Syed, and N. H. Memon, "Potential food safety risk in fruit production from the extensive use of fluorine-containing agrochemicals," Fluoride, vol. 53, no. 3, pp. 499-520, 2020.

[2] M. T. Habib, A. Majumder, A. Jakaria, M. Akter, M. S. Uddin, and F. Ahmed, "Machine vision based papaya disease recognition," Journal of King Saud University-Computer and Information Sciences, vol. 32, no. 3, pp. 300-309, 2020.

[3] A. Upadhyay et al., "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," Artificial Intelligence Review, vol. 58, no. 3, p. 92, 2025.

[4] S. Chowhan and S. R. Ghosh, "Role of ICT on agriculture and its future scope in Bangladesh," Journal of Scientific Research and Reports, vol. 26, no. 5, pp. 20-35, 2020.
[5] Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, 3(6), 610–621.
[6] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Recognition and classification of produce affected by identically looking powdery mildew disease," Acta Technologica Agriculturae, vol. 17, no. 2, pp. 29-34, 2014.
[7] M. Bhange and H. Hingoliwala, "Smart farming: Pomegranate disease detection using image processing," Procedia computer science, vol. 58, pp. 280-288, 2015.
[8] L. J. Rozario, T. Rahman, and M. S. Uddin, "Segmentation of the region of defects in fruits and vegetables," International Journal of Computer Science and Information Security, vol. 14, no. 5, p. 399, 2016.
[9] B. J. Samajpati and S. D. Degadwala, "Hybrid approach for apple fruit diseases detection and classification using random forest classifier," in 2016 International conference on communication and signal processing (ICCSP), 2016: IEEE, pp. 1015-1019.
[10] M. Vipinadas and A. Thamizharasi, "Banana leaf disease identification technique," International Journal of Advanced Engineering Research and Science, vol. 3, no. 6, p. 236756, 2016.
[11] J. Amara, B. Bouaziz, and A. Algergawy, "A deep learning-based approach for banana leaf diseases classification," 2017.
[12] C. U. Kumari, S. J. Prasad, and G. Mounika, "Leaf disease detection: feature extraction with K-means clustering and classification with ANN," in 2019 3rd international conference on computing methodologies and communication (ICCMC), 2019: IEEE, pp. 1095-1098.
[13] W. Liao, D. Ochoa, L. Gao, B. Zhang, and W. Philips, "Morphological analysis for banana disease detection in close range hyperspectral remote sensing images," in IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, 2019: IEEE, pp. 3697-3700.
[14] N. Saranya, L. Pavithra, N. Kanthimathi, B. Ragavi, and P. Sandhiyadevi, "Detection of banana leaf and fruit diseases using neural networks," in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020: IEEE, pp. 493-499.
[15] V. Chaudhari and M. Patil, "Banana leaf disease detection using K-means clustering and Feature extraction techniques," in 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020: IEEE, pp. 126-130.
[16] M. S. Devi, B. Vamshikrishna, J. A. Pandian, P. V. Rao, and S. R. Yaseen, "Eight convolutional layered deep convolutional neural network based banana leaf disease prediction," in 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2022: IEEE, pp. 313-317.
[17] V. G. Krishnan, J. Deepa, P. V. Rao, V. Divya, and S. Kaviarasan, "An automated segmentation and classification model for banana leaf disease detection," Journal of Applied Biology and Biotechnology, vol. 10, no. 1, pp. 213-220, 2022.
[18] Y. R. Rokade, "An Approach to Identify and Classify Banana leaf pests using Machine Learning and Deep Learning Neural Networks," Dublin, National College of Ireland, 2022.
[19] E. Mohanraj, R. Hariharan, S. Keerthana, and K. Cavin, "Banana Leaf Disease Detection using Advanced Convolutional Neural Network," in 2025 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 2023: IEEE, pp. 597-603.
[20] V. Chaudhari and M. P. Patil, "Detection and Classification of Banana Leaf Disease Using Novel Segmentation and Ensemble Machine Learning Approach," Applied Computer Systems, vol. 28, no. 1, pp. 92-99, 2023.
[21] N. B. Raja and P. S. Rajendran, "A novel fuzzy-based modified GAN and faster RCNN for classification of banana leaf disease," Journal of The Institution of Engineers (India): Series A, vol. 104, no. 3, pp. 529-540, 2023.
[22] M. Kumar and A. Kumar, "Deep Learning Meets Support Vector Machines: An Effective Hybrid Model for Banana Leaf Wilt Disease Severity Assessment," in 2024 2nd International Conference on Disruptive Technologies (ICDT), 2024: IEEE, pp. 386-390.
[23] J. D. Thiagarajan et al., "Analysis of banana plant health using machine learning techniques," Scientific Reports, vol. 14, no. 1, p. 15041, 2024.
[24] R. Jain, V. Kukreja, S. Chattopadhyay, A. Verma, and R. Sharma, "Hierarchy in Disease Severity Classification in Banana Leaves through the Integration of Convolutional Neural Network and Decision Tree Models," in 2024 5th International Conference for Emerging Technology (INCET), 2024: IEEE, pp. 1-5.
[25] S. Chattopadhyay, M. Manwal, V. Kukreja, and S. Mehta, "Transforming Agro-Diagnostics: Banana Leaf Diseases Through Federated Learning CNNs," in 2024 3rd International Conference for Innovation in Technology (INOCON), 2024: IEEE, pp. 1-6.
[26] M. Ying and Y. Jiang, "The inheritance and analysis of colors in Rimpa art: A machine learning perspective," Expert Systems with Applications, vol. 276, p. 127129, 2025.
[27] Z. Khan and J. Yang, "Nonparametric K-means clustering- based adaptive unsupervised colour image segmentation," Pattern Analysis and Applications, vol. 27, no. 1, p. 17, 2024.
[28] S. Adhithya and J. Kathirvelan, "Computer vision-based detection and classification of chemically ripened bananas and papayas at vendor site through deep learning AI models using real-time dataset," Results in Engineering, vol. 26, p. 104730, Jun. 2025, doi: 10.1016/j.rineng.2025.104730.
[29] Breiman, L., Friedman, J., Olshen, R.A. and Stone, C.J., 2017. *Classification and regression trees*. Chapman and Hall/CRC.
[30] A. Dolatabadian, T. X. Neik, M. F. Danilevicz, S. R. Upadhyaya, J. Batley, and D. Edwards, "Image-based crop disease detection using machine learning," Plant Pathology, vol. 74, no. 1, pp. 18–38, Jan. 2025, doi: 10.1111/ppa.14006.
[31] J. Padhi, K. Mishra, A. K. Ratha, S. K. Behera, P. K. Sethy, and A. Nanthaamornphong, "Enhancing paddy leaf disease diagnosis – a hybrid CNN model using simulated thermal imaging," Smart Agricultural Technology, vol. 10, p. 100814, Mar. 2025, doi: 10.1016/j.atech.2025.10081
[32] X. Xu, T. Xu, Z. Wei, Z. Li, Y. Wang, and X. Rao, "Enhancing citrus surface defects detection: A priori feature guided semantic segmentation model," Artificial Intelligence in Agriculture, vol. 1, pp. 67–78, 2025, doi: 10.1016/j.aiia.2025.01.00
[33] Breiman, L., 2001. Random forests. *Machine learning*, *45*(1), pp.5-32.
[34] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., 1984. Classification and regression trees Belmont. *CA: Wadsworth International Group*.