Feature Engineering and Multi-Level Image Descriptors for Banana Leaf Disease Datasets Using Texture Analysis and Transfer Learning

Thomas Kolawole Waheed[1], Dr. Devender Kumar[2]
[1,2]*School of Computer Applications, Lovely Professional University, Punjab, India*

Abstract*—The recent advancement in AI applications has permitted broad utilization of AI tools in almost every aspect of human endeavor, including agriculture and the agro- food industry. Deep learning is known for its potential to extract and learn subtle texture features from images, which makes it applicable to challenges in the detection of plant diseases. Bananas are known to suffer from diverse kinds of diseases such as Sigatoka, Cordana, among others, which pose a serious threat to food security and the economy of nations. Traditional methods of addressing this threat have proved ineffective, considering the associated constraints in terms of early detection and the subtlety of some diseases, besides the technicalities that conventional methods might not be able to address to this end. Automated detection is a better substitute. The core aim of this study is to systematically compare handcrafted texture-based machine learning approaches and deep feature-learning approaches for recognition of banana leaf diseases. Specifically, Random Forest and Support Vector Machine are evaluated using Gray-Level Co-occurrence Matrix (GLCM) texture features, and EfficientNetV2 and YOLOv8n are evaluated as deep-learning approaches. Rather than proposing a new model architecture, the contribution of this work is a comparative experimental framework that examines the relative performance of these different feature-learning paradigms on the BananaLSD dataset from Kaggle, containing 937 images of Healthy, Sigatoka, Cordana, and Pestalotiopsis. The classification results showed that EfficientNetV2 achieved the highest accuracy of 96.81%, followed by SVM at 89.89%, and Random Forest at 88.30%. Under the derived image-level detection formulation, YOLOv8n achieved a verified mAP50 and mAP50-95 of 0.948. Because accuracy and mAP represent different task-specific evaluation measures, these results are interpreted separately rather than used to establish a direct numerical ranking among all models. GLCM Random Forest and GLCM SVM performance are worthwhile, but are no match for the outstanding performance of EfficientNetV2 and YOLOv8n, especially with nuanced features of banana diseases. The outcome of this research is proof that establishes the adaptability of AI techniques in smart agriculture.

***Index Terms*—Banana Leaf Disease, Feature engineering, Banana Dataset, AI and Edge computing.**

I. INTRODUCTION

The world population has been projected to reach approximately 10 billion by 2050, and the world is likely to witness a shortage of essential human needs, basically food, considering the trend and threat to food security [1], necessitating the introduction of advanced technology, particularly AI algorithms, to address the challenge of food security and climate change [2]. Musaceae, or Bananas, are known for sweetness and nutritional value [3][4], classified as the most significant fruit in the global market [5]. Bananas are a crucial staple food of great significance and mostly grow in tropical and subtropical regions of the world. About 124 million metric tons were produced in 2021 (FAO 2023), but it also suffers huge losses due to diverse diseases caused by pathogens such as Fusarium Wilt, Sigatoka, Panama Disease (Fusarium Wilt), and many others, ranging from 30 to 100% [6][7]: besides, there are climate effects on the plant[8]. The diseases affect parts of the plant, but more often the leaves; they have a significant impact on banana productivity[10]. These diseases are a threat to the sustainability of banana production and the global economy [11], as confirmed by a [12] survey; therefore, they demand continuous research effort to find a lasting solution to the threat and management of the disease [13].

Plant disease is known to be a conditions that affect the well-being of the plant, its growth and yield, and,

in severe cases, death of the plant [14]. Plants encounter enormous health challenges whether due to changing seasons, pathogen attack, erosion resulting in the loss of essential minerals, or human activities, making the detection of plant diseases a rigorous exercise. [15] Pathologists and experts therefore adopt different ecological methods and molecular analyses to determine parameters of infected plants and the spread. The plant healthcare industry 4.0 has experienced a series of transformations over the decades, building on the existing knowledge and technology by researchers and experts, aimed at improving detection and sustaining productivity of crops [16]. Timely detection and diagnosis of diseases are vital for mitigating losses and improving crop management. Early investigation involves engineering feature extraction and ML classifiers, achieving fair results [17]. Traditional methods of addressing this threat have proved ineffective, considering the associated constraints in terms of early detection and the subtlety of some diseases, among others [18]; this therefore calls for a better alternative for automated detection [19].

The emergence of advanced technologies such as image processing, texture pattern analysis, multilevel thresholding [14], and deep learning has brought a revolution to traditional agricultural practices, resulting in increased productivity, better quality control of the process, and higher accuracy [20]. Hinton et al. [21] introduced Deep learning based on the concept of artificial neural networks (ANN), which led to renewed research efforts in neural networks and applications [17]. Different deep learning architectures have been successfully applied in resolving plant diseases based on leaf images [22], though with huge dataset requirements and associated constraints [23]. Precision agriculture is a means to transform agricultural practice, ease labor, increase productivity, and revenue generation of local farmers and nations through the agent of AI technology [24]; furthermore, classification of plant diseases has been enhanced by the introduction of cutting-edge techniques, such as remote sensing and imagery [25]. AI-based systems coupled with sophisticated algorithms are available to process datasets and extract nuanced features imperceptible to the human eye[26], and impossible with traditional methods, and to estimate of the severity of plant disease [27]. The recent advancement in agricultural practices not only reduces human labor but also enhances decision-making in real time, which is crucial in maintaining the competitiveness and sustainability of Argo and achieving the United Nations Sustainable Development Goals (SDGs).

II. LITERATURE REVIEW

AI algorithms, especially deep learning, have emerged as evolutionary tools to address different human challenges, of which plant diseases are crucial. [6] Applied AI tools and high-resolution satellite imagery to detect and classify diseased (Bunchy top disease and Xanthomonas Wilt) and healthy banana leaves. RetinaNet, an object detector and classifier model, achieved the highest accuracy of 99.4% for banana bunchy top disease and exceeded other state-of -the-art models. A deep learning-based solution (CNNs) was proposed to classify a dataset of healthy and four disease leaves (Sigatoka, Panama Disease, Banana Bunchy Top Virus (BBTV), and Xanthomonas Wilt) by [13]. The performance of the proposed model was outstanding, with an accuracy of 92%, compared to traditional approaches.

[26] presented a hybrid framework integrating Vision Transformer (ViT) and SVM for robust classification of banana leaf disease; self-supervised and semi-self-supervised mechanisms were applied. The four-class BananaImageBD and the six-class BananaSet datasets were used. The proposed model achieved accuracies of 99.86% ± 0.099 for BananaSet and 99.70% ± 0.17 for BananaImageBD, compared to the traditional approaches. The model displayed the capability to extract texture features of importance for applications, especially for the banana. The author[28] proposed a deep transfer learning approach to classify texture features of banana leaves using real-time images: the DL model successfully classified healthy bananas from diseased ones with accuracies ranging between 70 -99% for the test. [29] Proposed an integrated learning-based (CNN &FSVM) solution for classification of banana leaf disease; the dataset comprised 3500 images made up of infected and healthy parts of the banana collected in India. This study is beneficial to farmers in timely detection of disease and application of fertilizer. The proposed model achieved an accuracy of 99% among other metric evaluations compared to other deep-learning techniques.

Author [30] employed MobileNetV3 to demonstrate real-life application of DL, using the PlantVillage dataset to classify leaf disease, the model accuracy 99.50%, which is cost-effective, and post training optimization increase efficiency of the model further, and MobileNetV3 deployed in ONNX format for field work. [31] ML optimization for classification of banana leaf disease; the methodology adopted includes preprocessing, augmentation, and feature extraction techniques. SVM, KNN, Random Forest, and Gradient Boosted Stacking Ensemble were implemented. Gradient Boosted Stacking Ensemble with an accuracy of 95.53% excelled; the results of the research display the framework of a reliable solution to plant disease automation. Author [6] carried out segmentation of Fusarium Wilt and Black Sigatoka using U-Net semantic segmentation DL, with data collected using a cellphone and labeled with the assistance of a pathologist. The model achieved a Dice coefficient of 96.45% and an IoU of 93.23%, and made a clear segmentation of the diseased portion of the plant; the annotations are available in an open–access repository.

[32] Aimed at improving the portion of detection of Black Sigatoka using deep learning, employed the service of drones and smartphones for the collection of the dataset. The study includes construction of an original dataset via drone images and false–color images using multispectral sensors, totaling 1192 diseased images and 3102 healthy leaves. MobileNetV2 recorded 86.5% accuracy for the classification. Author [28] carried out AI-based banana detection using DCNN. Images were collected in Africa and South India; six models were trained for eighteen classes of banana leaf diseases. RestNet50 and InceptionV2 based model achieved 90% accuracy, a superior result compared to other state–of–the–art models. The author [33] applied feature engineering (GLCM and NGTDM) to detect and classify Sigatoka and Xanthomonas, Support Vector Machine achieved 90% accuracy above other models implemented for classification.

[34]proposed application of AI algorithms in real-time monitoring of plant disease; grape leaf disease, modified MobileNetV3 Large achieved a higher 99.66% accuracy, compared to other models. MobileNetV3 was finally deployed on an edge device (Nvidia Jetson Nano), and Grad-CAM Visualization was conducted for the classification.[35] Designed a DCNN for early classification of plant illness. The proposed model achieved 99.0% accuracy compared to models implemented; the result confirms CNN reliability and precision in classification of banana leaf disease. The author [36] proposed an architectural modification of DL for Moko and Sigatoka classification, using imagery collected through UAV. The best hyperparameter configuration yielded mean precision 79.6%, recall 80.3%, mAP@0.5 84.9%, and mAP@0.5:0.95 62.9%; segmented YOLOv8 performance surpasses other implemented models. [40] Developed AI-IoT smart agriculture for classification across a multi-crop dataset of 25,940, which was augmented and comprises eleven classes, including healthy. RestNet50 achieved 99.8% accuracy, among other metrics, compared to state-of – the –art models.

Research Novelty and Contributions

AI techniques have been extensively applied in plant disease recognition, but there is a need for studies such as a systematic investigation of different feature learning techniques in the same domain, as we have done in this study for traditional texture-based machine learning, EfficientNetV2 transfer learning, and YOLO-based methods on banana leaf diseases using BananaLSD dataset.

The novelty of our study lies solely in the comparative experimental setup whereby various feature learning paradigms are evaluated for banana leaf diseases, rather than proposing a new neural network architecture.

- Comparative evaluation among feature-learning paradigms: The work compares the performance of models based on handcrafted texture features with automatic feature extraction for the classification of BananaLSD dataset, which comprises Healthy, Sigatoka, Cordana, and Pestalotiopsis. Machine-learning models were trained using features extracted by the Gray Level Co-occurrence Matrix (GLCM), and for EfficientNetV2, transfer learning was applied to learn discriminative image representations automatically.
- Evaluation of conventional and deep-learning approaches within the same application domain: The outcome of our study creates a basis for future comparative analysis of traditional machine-learning and deep-learning approaches, and the

relative performance of the models was also highlighted.

- Evaluation of texture-based and deep feature representations: The research equally investigates the effectiveness of handcrafted GLCM texture descriptors relative to deep representations learned through transfer learning. This comparison is informative, revealing the extent to which conventional texture features can sustain the discrimination of diseases compared to deep feature learning.
- Class-wise performance analysis: Besides the overall performance measures, the study evaluates the model behavior on the four different leaf disease classes, and consequently reveals the ability of the methods to distinguish between healthy and infected images of BananaLSD.

In particular, the contributions of this study are: A practical baseline for future studies on banana disease detection, establishing a baseline for performance evaluation of conventional machine-learning and deep-learning approaches on the BananaLSD dataset. Furthermore, it serves as a reference study regarding development and improvement in model architectures, hybrid models, and multi-object detection, among others. The results of the research reveal the degree to which extracted texture features can compete with deep-feature learning.

This paper is organized as follows: Section1 Introduction, Section2 Review of literature, research novelty and contributions, and history of banana disease, Section3 Description of the materials and methodology; model implementation, Section 4 Results, followed by comparative discussion of models, Section 5 Summary of specific task performance, limitations, and conclusion.

III. BANANA DISEASE'S HISTORY, SYMPTOMS, CAUSES, AND MANAGEMENT STRATEGY

Banana plants are known to be vulnerable to diseases and are recognized by different symptoms [37], which pose a significant threat to their survival, productivity, and revenue generation expected from the crop. Generally, banana diseases are categorized into fungal, bacterial, and viral diseases [9]. It is a global phenomenon; some of the major banana diseases are discussed below. Though Cordana is considered a minor disease, it can cause leaves to fall after the rainy season [39].

Panama disease is a banana disease caused by Fusarium oxysporum f., a soil-borne fungus. It became known in the 20th century after it wiped out vast plantations of the Gros Michel banana variety, leading to the introduction of the Cavendish variety, though a new strain of fungus had been discovered. The fungus mostly attacks the vascular system, causing it to wilt. Fusarium wilt spreads quickly, aided by infected soil, water, and plant material, among others. The pathogen can survive for decades in the soil and is hard to eliminate. Control measures include planting resistant varieties, crop rotation, and the use of chemicals, though not without side effects.

Sigatoka has two types: Black sigatoka, caused by Mycosphaerella fijiensis, and yellow sigatoka, caused by Mycosphaerella musicola, and they are known to be a global threat to banana crops. The fungi affect leaves and their ability to photosynthesize, leading to low yield and quality. The symptoms appear as small, light-colored streaks on the leaves at the start[38]. For Black Sigatoka, these streaks may develop into dark brown or black spots, coalesce, and cause death of the leaves. But Yellow Sigatoka usually develops yellowish streaks that turn brown as the disease progresses further [41]. It is recommended to apply fungicides regularly and grow resistant banana varieties as control measures.

Banana Bunchy Top Virus (BBTV) is known to be a severe viral disease. The banana aphid, Pentalonia nigronervosa is the major agent of transmission; it spreads through infected materials and leads to a decline in banana yield. The symptoms include dark green streaks on the leaf veins and petioles, and at an advanced stage, newly grown leaves become narrow, smaller, and have a bunchy top. To prevent further spreading of the virus into new plantations, it is recommended to use virus-free certified planting materials; other control measures include the use of insecticides, biological control agents, and regular weeding.

Xanthomonas Wilt is a severe bacterial disease caused by *Xanthomonas campestris* pv. *Musacearum*. It affects all varieties of banana and can lead to huge losses to farmers and the government due to the death

or poor yield of the crop. BXW was first reported in Ethiopia and is known to manifest a variety of symptoms on different parts of the plant, such as wilting and yellowing of leaves.

BXW is known to spread through multiple vectors, but primary transmission is through insects; the bacteria also spread through contaminated soil and surrounding waters. To prevent BXW from spreading into neighboring plantations, banana farmers are advised to use disease-free planting material and adhere to best sanitation practices.

IV. PROPOSED SYSTEM

The core aim of this study is to systematically compare handcrafted texture-based machine learning approaches and deep feature-learning approaches for the recognition of banana leaf diseases. BananaLSD from Kaggle, containing 937 images, was utilized. Below is the flowchart of the ML and DL classification, and a brief explanation of the methodology.

V. MATERIALS AND METHODS

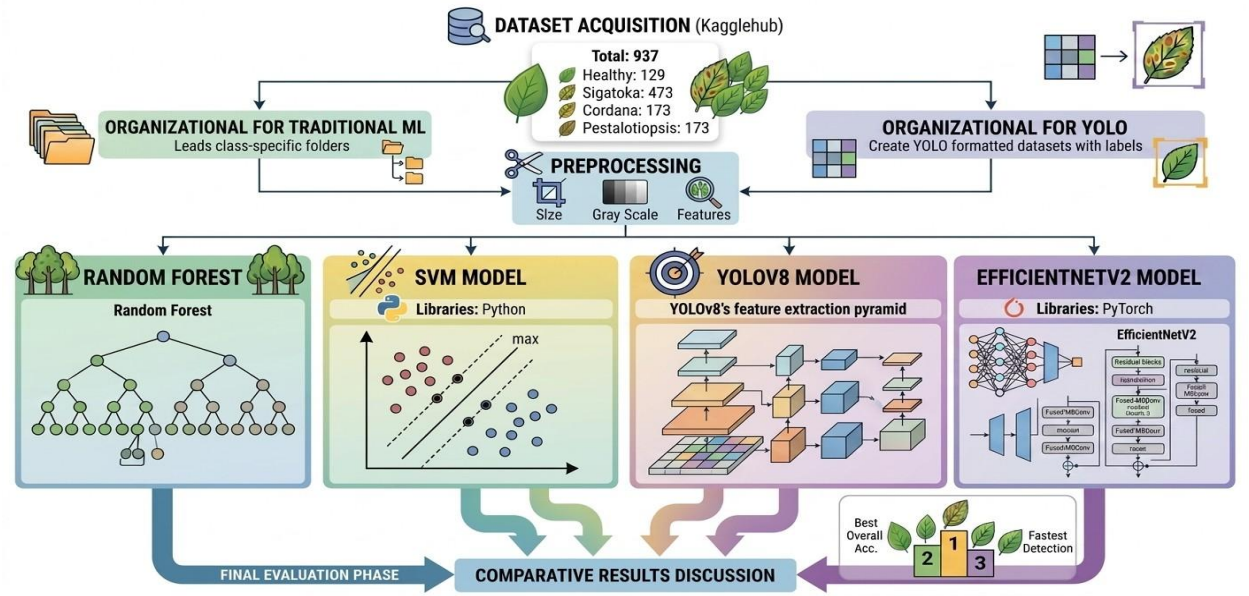

Figure 1. Flowchart of the Machine Learning and Deep Learning Classification of Banana Leaf Diseases.

5.1 Data Description and Preprocessing

Feature extraction is the preparatory ground for machine learning investigations; it is the process of converting raw data into digital form and extracting useful informative parameters/features such as shape, texture properties, and color [42], otherwise known as feature engineering, and serves as the preparatory ground for ML classification [43]; for example, morphological traits can be very helpful in the identification of spots on an infected leaf. [44] The Gray-Level Co-occurrence Matrix (GLCM), being a famous extraction technique that relates the pixel intensity to the images[47], therefore, computing the GLCM can reveal the distinct feature properties of the healthy and diseased plant. Over the past decade, several researchers have made use of texture features to train machine learning models in banana leaf disease detection[14][45]. The BananLSD classification was implemented in a Google Colab environment. The dataset was acquired from KaggleHub and contains 937 images in total (Original), comprising 129 healthy, 473 Sigatoka, 162 Cordana, and 173 Pestalotiopsis images, organized into class-specific folders for traditional machine learning and creating a YOLO formatted dataset with corresponding label files for the object detection task, respectively.

5.2 Experimental Protocol and Dataset Partitioning

The experimental protocol was designed to evaluate conventional machine learning and deep learning approaches for banana leaf disease recognition using

the BananaLSD dataset. The dataset consists of 937 images belonging to four categories: Healthy, Sigatoka, Cordana, and Pestalotiopsis.

For image classification experiments, the dataset was partitioned into training and validation subsets, using an 80:20 split ratio. The training subset was used for model fitting, whereas the validation subset was used to monitor model performance and evaluate the trained models. The reported validation results for the image classification experiments were obtained from the held- out subset containing 188 images, with the remaining images used for training.

The same general experiment framework was used to compare the classification approaches: Random Forest and Support Vector Machine were trained using the GLCM–derived image features, whereas EfficientNetV2 was trained using image data and transfer learning. Their performance was evaluated primarily using metrics, including accuracy and class-wise performance measures.

YOLOv8n was evaluated separately under the derived image-level object formulation described in Section 8.2.1. For this experiment, images were organized into YOLO training and validation sets, and each image was associated with one automatically generated full-image bounding box and the corresponding disease category label. YOLOv8n performance was evaluated using object-detection metrics, including Precision, Recall, mAP50, and mAP50-95.

No claim of a common independent test set across all four models is made unless such a test set was explicitly created and used in the experiment. Therefore, the reported results are interpreted according to the held-out evaluation procedure and task- specific metrics used for each modelling approach. The complete experimental protocol is summarized in Table 1.

Table1 Experimental dataset partitioning and evaluation protocol

Experimental approach	Model	Input representation	Dataset partitioning	Training data	Held-out data	Primary evaluation metrics
Machine Learning	Random Forest	GLCM texture features	80:20 split	80% of dataset	20% of dataset	Accuracy, Precision, Recall, F1-score
Machine Learning	SVM	GLCM texture features	80:20 split	80% of dataset	20% of dataset	Accuracy, Precision, Recall, F1-score
Deep Learning	EfficientNetV2	Original images	80:20 split	Approximately 80% of the dataset	188 images	Accuracy, Precision, Recall, F1-score
Deep Learning	YOLOv8n	Images + generated full-image bounding boxes	80:20 train/validation organization	Approximately 80%	Approximately 20%	Precision, Recall

Note: The classification and YOLOv8n experiments address different predictive tasks and were therefore evaluated using task-specific metrics. The 80:20 partition was used to separate model training from held-out evaluation.

A separate independent test set is not reported in the present study unless explicitly stated for a specific experiment.

The figure should follow this sequence:

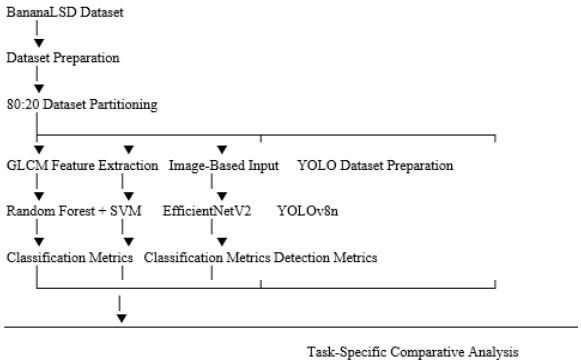

Figure2. Experimental protocol used for the comparative evaluation of GLCM-based Random Forest and SVM, EfficientNetV2, and YOLOv8n using the BananaLSD dataset

Table2 Class distribution across partitions

Disease category	Total images	Training images	Held-out images
Healthy	129	107	22
Sigatoka	473	385	88
Cordana	162	123	39
Pestalotiopsis	173	134	39
Total	937	749	188

Note: The dataset partitioning was performed while preserving, as closely as possible, the class distribution across the training and held-out subsets.

5.3 Model Evaluation Procedure

The experimental procedure for this study was carried out in stages. The first stage is the preprocessing of the images and partitioning of images into training and held-out validation subsets (80:20). Next is the GLCM feature extraction for Random Forest and SVM classification. EfficientNetV2 transfer learning was imported and used on the original dataset partitioned into training (80%) and held-out validation subsets. The model was trained, and performance evaluation was done. Finally, the images were organized into YOLO format for the YOLOv8n experimentation; for each image, one automatically generated full-image bounding box was assigned. Eighty percent (80%) of the images in the training subset were used for fine-tuning of the model, while the remaining (20%) validation subset was utilized for performance evaluation.

The results of the experimentation are presented according to task-specific metrics: classification models were assessed using accuracy, F1-score, precision, and recall, while YOLOv8n was assessed based on appropriate object-detection metrics.

Note: In this study, the term “cross-validation” is assigned to fold-based repeated evaluation of Random Forest and SVM models. The terms “validation” and “held-out validation” are used for the designated subsets used for EfficientNetV2 and YOLOv8n.

VI. GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) AND MACHINE LEARNING MODELS

The first step in applying conventional machine learning techniques is to conduct preprocessing of each image of BananaLSD, resize it to 128 * 128 pixels, and convert it to grayscale. The grayscale image was used as input for calculating the Gray-Level Co-occurrence Matrix (GLCM). The computation was done taking into account the spatial co-occurrence of grayscale intensity values at predefined offsets.

Specifically, the GLCM was run with distances of 1, 3, and 5 pixels, angles of 0°, 45°, 90°, and135°, and 256 as the number of gray levels in the GLCM, and the matrix was calculated with symmetric= True. For preprocessing, each image was converted from RGB to CIELAB color space, and the selected channel was used as input for the texture extraction.

For every distance-angle combination, five texture descriptors were extracted, namely contrast, dissimilarity, homogeneity, energy, and correlation. The feature characteristics and image texture are complementary; texture features are thus defined: Contrast is the measurement of local intensity variation, Dissimilarity quantifies the difference between neighboring intensity values, Homogeneity is the measurement of the closeness of GLCM elements to the main diagonal, while Energy reflects the uniformity of the texture distribution, and Correlation is the description of the linear dependency between neighboring pixel intensities.

Let P (i, j) denote the normalized GLCM element representing the co-occurrence of grayscale levels i and j. The texture features were computed from the GLCM for each specified spatial configuration. The resulting feature values from all distance-angle combinations were then concatenated to produce the final feature vector for each image.

Consequently, each image was represented by 420 texture features calculated as:

Nfeatures=Nproperties*Ndistance*Nangles

Where Nfeatures =7, Nproperties=5, Ndistance=3, and Nangles=4 for contrast, dissimilarity, homogeneity, energy, and correlation.

1 (grayscale) + 3 (RGB) + 3 (LAB) channels = 7

7 channels * 3 distances * 4 angles * 5 properties = 420 numerical features per images

The resulting feature matrix was used as input to the Random Forest and Support Vector Machine classifiers. Thus, conventional machine-learning models were trained using the same GLCM-derived texture representation rather than the original image pixels.

Table3 GLCM extraction parameters used for the Random Forest and SVM experiments

Parameters	Value used in the experiment
Input image size	128*128
Image representation	Grayscale
GLCM implementation	Skimage.feature/ graycomatrix/graycoprops
Gray-level quantization (level)	256
Distances	1, 3, 5 pixels
Angles	0°, 45°, 90°, 135°
Symmetric GLCM	True
Normalized GLCM	True
Texture properties	Contrast, Dissimilarity, Homogeneity, Energy, Correlation
Number of distance-angle combinations	420
Feature aggregation	Concatenation
Final features per image	Gray Level Co-occurrence Matrix (GLCM) texture features
Feature scaling before SVM	0, 1
Feature scaling before RF	0, 1

For each image, the five GLCM properties were calculated for every specified distance-angle combination. The extracted values were then concatenated to form a single fixed-length. Accordingly, the feature vector contained 420 variables per image. For the complete dataset of 937 images, the resulting feature matrix had dimensions 937 * 420, before partitioning into the subsets used for model development and evaluation.

6.1 GLCM Texture Measures

The principal GLCM texture descriptors used in this work were as follows. P (i, j) denotes the normalized co-occurrence probability:

Contrast

Contrast=i, j∑ (i-j) 2P(i,j)

Dissimilarity

Dissimilarity =i, j∑|i-j|P (i,j)

Homogeneity

Homogeneity =i, j∑1+ (i-j)2P(i,j)

Energy

Energy=i, j∑P (i,j) 2

Correlation

Correlation=i, j∑σiσj (i-μi) (j- μj) P (i,j)

Where μi and μj are the mean gray-level values, and σi and σj are their corresponding standard deviations.

6.2 GLCM Computation and Machine Learning

The Random Forest was trained using GLCM texture feature extraction from the preprocessed images discussed above, each consisting of 420 texture variables. RandomizedSearchCV to determine the hyperparameter optimization was conducted the details are in Table 5.

SVM was trained using the same texture extraction of GLCM as for RF. Therefore, any differences between the two machine-learning models should be considered as differences in the learning algorithms, rather than differences in the extracted texture input.

We optimized the hyperparameters using RandomizedSearchCV, and the best configuration was a polynomial kernel with c=6.3330 and y=0.0341; details in Table5. MiniMaxScaler from sklearn. preprocessing was applied for feature scaling, typically ranging from" 0 to 1".

VII. DEEP LEARNING:

7.1 EfficientNetV2 Transfer Learning

For EfficientNetV2, the raw dataset was used; PyTorch and torchvision were imported libraries, and then the path for the images (training and validation/test sets) was clearly defined. Necessary transformations such as resizing, normalization, and data augmentation were implemented. From the library, a pretrained EfficientNetV2-S was loaded with the following parameters: (device: 'CPU', image size: '224'*'224', bath size:'32', number of workers: 2, loss function: 'CrossEntropyLoss', optimizer: 'Adam', learning rate explored:'0.001', '0.0005', '0.0001', '0.00005', epochs: 20), and the head of the classifier was modified for four classes in accordance to the dataset, with the learning rate of the model.

The dataset was split into an 80% training subset (plus augmentation) and a 20% validation/test subset excluding augmentation. Hyperparameter tuning was conducted, and the model was trained and evaluated appropriately.

7.2 .1 YOLOv8n Detection Target and creation of bounding boxes

YOLOv8n for banana leaf disease recognition in an object detection framework. The experimental dataset BananaLSD was not manually annotated, but lesion-level bounding boxes were automatically generated using a word-derived detection-labeling strategy. The whole image is regarded as one object, and the relevant disease category was assigned as the class label of the object. The target is the image level of the disease classes, and not individual lesions or spots, nor the accurately segmented infected area.

Following the YOLO normalized annotation format.
(class_id, xc,yc,w,h),
The generated bounding box for every image was:
(Class_id, 0.5, 0.5, 1.0, 1.0),
Where xc=0.5 and yc=0.5 represent the center of the image, while w=1.0 and h=1.0 represent the complete normalized image width and height, respectively. The class_id corresponds to one of the four BananaLSD categories.

Thus, each YOLO label file had one annotation in accordance with the general format: class_id 0.5 0.5 1.0 1.0

This approach made possible the classification of the BananaLSD dataset using the YOLOv8n detection architecture. However, the localization was simplified because the ground-truth object extended and corresponded to the entire image. Therefore, the results should be interpreted as the joint performance of image-level category recognition and full-image object localization under the adopted detection formulation, and never as evidence of individual lesion-level or detection of disease spots. The reported mAP50 and mAP50-95 are a reflection of strong agreement between the predicted boxes and the generated full-image bounding boxes at the evaluated Intersection over Union (IoU) threshold, but should not be treated as the capability of YOLOv8n to precisely delineate individual disease lesions or spots.

Table4. YOLOv8n class labels and detection targets

Class ID	Class name	Detection target	Bounding-box definition
0	Healthy	Image-level Healthy category	Entire image
1	Sigatoka	Image-level Sigatoka category	Entire image
2	Cordana	Image-level Cordana category	Entire image
3	Pestalotiopsis	Image-level Pestalotiopsis category	Entire image

Each input image was therefore represented by one class label and one automatically generated bounding box covering the complete image.

7.2.2 Automatic Generation of YOLO Labels

The YOLO annotation files were generated programmatically during dataset preparation. For each image, its disease class was determined from the corresponding dataset category, and a text file with the same filename as the image was created.

The annotation contained the numerical class identifier followed by the normalized bounding-box coordinates. Since the full image was defined as the detection target, the coordinates were fixed at 0.5, 0.5, 1.0, and 1.0 for all images. No manual lesion-level bounding-box annotation was performed in this experiment.

VIII. EXPERIMENTAL REPRODUCIBILITY FOR THE MODELS

To ensure experimental reproducibility, the complete training and model-selection configurations used for Random Forest, Support Vector Machine, EfficientNetV2, and YOLOv8n models are summarized in table5-7. Hyperparameters obtained through randomized search are reported separately from final training parameters. Unless otherwise specified, the reported values correspond to the final experimental configurations used to generate the results presented in this study Table 5

Table5. Hyperparameter optimization and final configurations for the conventional machine-learning models.

Model	Parameters	Search range/configuration	Final selected value
Random Forest	n_estimmator	Random integer from 100 to 499	299
Random Forest	max_feature	sqrt, log2, 0.8, 1.0	sqrt
Random Forest	max_depth	Random integer from 10 to 99	65

Random Forest	mini_samples_split	Random integer from 2 to 19	3
Random Forest	Mini_samples_leaf	Random integer from 1 to 9	1
Random Forest	Model random seed	Fixed	42
Random Forest	Hyperparameter search method	RandomizedSearchCV	RandomizedSearchCV
Random Forest	Search iterations	Fixed	50
Random Forest	Cross-validation folds during search	Fixed	5
Random Forest	Search random seed	Fixed	42
SVM	Kernel	Candidate distribution defined in the search space	poly
SVM	C	Search distribution defined in svm_param_dist	6.3330
SVM	gamma	Search distribution defined in svm_param_dist	0.0341
SVM	Model random seed	Fixed	42
SVM	Hyperparameter search method	RandomizedSearchCV	RandomizedSearchCV
SVM	Search iteration	Fixed	50
SVM	Cross-validation folds during search	Fixed	5
SVM	Search random seed	Fixed	42

Note: For both conventional machine-learning models, hyperparameter optimization was performed using 50 sampled parameter configurations with five-fold cross-validation and random_state=42. The Random Forest and SVM models were subsequently evaluated using the test subset.

For Random Forest and SVM experiments, model hyperparameters were selected using RandomizedSearchCV on the training data. The search procedure used five-fold cross-validation (cv =50), 50 sampled configurations (n_iter=50), and random_state=42. For Random Forest, the final selected configuration consisted of 229 “sqrt” = max_features, max_depth=65, min_samples_split=3, and min_samples_leaf = 1. For SVM, the selected configuration used a polynomial kernel with C-6.3330 and γ=0.0341. The tuned estimators were subsequently used for prediction on the held-out test.

Table6 EfficientNetV2-S training configuration

Parameter	Configuration
Base architecture	EfficientNetV2-S
Number of output classes	4
Input image size	224*224 pixels
Batch size	32
Number of DataLoader workers	2
Device	CPU
Loss function	CrossEntropyLoss
Optimizer	Adam
Learning rate explored	0.001, 0.0005, 0.00001, 0.00005
Number of epochs	20
Training data augmentation	Applied
Image normalization	Applied
Validation data augmentation	Not applied
Dataset partition	Report separately according to the experiment split table
Random seed for dataset split	torch. Generator (). manual_seed(42)
Final selected learning rate	0.0001 (or 1e-4)

EfficientNetV2-S was trained by fine-tuning a pretrained network after modifying its classifier head to produce predictions for the four BananaLSD categories.

Images were resized to 224 * 224 pixels, and the training pipeline included data augmentation and normalization, whereas the validation data were transformed without training-time augmentation.

The four learning rates- 0.001, 0.0005, 0.0001, and 0.00005- were explored with the Adam optimizer over 20 epochs. The final learning rate used to obtain the result 0.0001.

Table7 YOLOv8n training and augmentation configuration

Parameter	Configuration
Base model	yolov8n.pt
Model type	YOLOv8 nano
Number of classes	4
Dataset configuration	yolo_data. yaml
Training epochs	30
Image size (imgsz)	640*640 pixels
Batch size	16
Device	CPU
Optimizer setting	auto
Optimizer selected for the recorded run	AdamW
Learning rate (1r0)	0.00125
Momentum	0.9
Early-stopping patience	10 epochs
Plot generation	True
Hue augmentation (hsv_h)	0.015
Saturation augmentation (hsv_s)	0.7
Value augmentation (hsv_v)	0.4
Vertical flip (flipud)	0.0
Horizontal flip (flip1r)	0.5
Mosaic	1.0
MixUp	0.0
Copy-paste	0.0
Degrees	0.0
Translation	0.1
Scale	0.5
Shear	0.0
Weight decay	0.0005
Warmup epochs	3.0
Random seed	random. seed (42)
Deterministic training	True

YOLOv8n was initialized from the pretrained yolov8n model and trained for 30 epochs using 640 * 640-pixel inputs and a batch size of 16. The training run used the automatic optimizer selection setting, for which the recorded run selected AdamW with a learning rate of 0.00125 and momentum of 0.9. Early stopping used a patience of 10 epochs. Data augmentation included hue, saturation, and value transformations, horizontal flipping, and mosaic augmentation, while MixUp and copy-paste augmentation were disabled. The complete training arguments, including all remaining optimization and augmentation parameters, should be reproduced directly from the args.YAML file associated with the reported run.

IX. REPRODUCIBILITY OF MODEL CONFIGURATIONS

The hyperparameter configurations reported in Table5-7 correspond to the experimental settings used for the reported results. For complete reproducibility, the study retains the Random Forest and SVM hyperparameter-search outputs, the EfficientNetV2 training configuration and training history, and the YOLOv8n args. yam1, results.csv, and associated validation outputs.

Parameters not explicitly optimized were retained at the software defaults or at the values recorded in the corresponding training configuration files. No parameter value should be inferred retrospectively where the original experimental record is available.

Table8 Distinction between validation and cross-validation procedures used in the study

Procedure	Meaning in the study	Models
Training	Model parameters fitted using the training subset	RF, SVM, EfficientNetV 2, YOLOv8n
Held-out validation	Performance evaluated using a designated subset not used for parameter fitting	EfficientNetV 2, YOLOv8n
Cross-validation	Repeated fold-based evaluation performed during model development/hyperparam eter selection	Random Forest, SVM
Final held-out evaluation	Performance assessment using the designated evaluation subset after model development	RF, SVM, where applicable

Note: A validation subset and a cross-validation fold serve different roles.
Validation evaluates the model on a designated held-out subset, whereas cross-validation repeatedly rotates the validation role across folds of the available development data.

X. RESULTS AND DISCUSSION:

Based on calculated means and standard deviations of texture features, observed patterns and distinguishing features across different classes are stated below:

10.1 Gray Level Co-occurrence Matrix (GLCM) texture features from the images.

Table9 Grayscale Contrast (`grayscale_contrast_1_0`):

Classes	Mean
Healthy	42.0
Cordana and Pestalotiopsis	69.0 – 77.0
Sigatoka	47.0

Healthy leaves have the lowest mean contrast 42, suggesting smoother and less varied gray- level transitions. For Cordana and Pestalotiopsis, the mean contrast values range from 69 to 77, implying more abrupt changes in pixel intensity, which may be due to the presence of spots, lesions, or other distinct texture elements, while Sigatoka displayed a moderate mean contrast of 47, suggesting variability within the class.

Table10 Red Channel Energy (`red_energy_1_0`):

Classes	Mean
Healthy	~0.08
Sigatoka, Cordana and Pestalotiopsis	0.04-0.05

Energy is the measure of local uniformity and repetitions of pixel pairs. So, high energy values indicate the uniformity of texture, as established with healthy leaves, and low energy values indicate the distortion of texture, as in diseased leaves.

Table11 L-channel Homogeneity (`L_homogeneity_1_0`):

Classes	Mean
Healthy	~0.95-0.96
Sigatoka, Cordana and Pestalotiopsis	~0.90-0.94

Homogeneity is a measure of the closeness of the distribution of elements in GLCM to the GLCM diagonal.

It indicates the homogeneity of the texture of the image for a specific class; thus, higher values suggest a more homogeneous texture, as observed in healthy leaves, while lower values indicate a less homogeneous texture, as observed in Sigatoka, Cordana, and Pestalotiopsis

10.2 Overall Patterns and Imbalances:

GLCM features are essential in differentiating between healthy and infected leaves. From the above observations, Healthy leaves showed lower contrast, higher red energy, and homogeneity over different channels and distances, which means uniformity in texture. However, for Sigatoka, Cordana, and Pestalotiopsis, the opposite is true, showing higher contrast, lower red energy, and lower homogeneity under the same conditions as healthy leaves.

The variation suggests a complexity of texture, possibly due to the presence of spots or lesions on the leaves. The observed variation in the calculated standard deviation indicates variability within the class as observed for Sigatoka. The results therefore point to the fact that GLCM features are important parameters in distinguishing between healthy and diseased leaves, and between different diseases [46].

10.3 Average RGB Histograms

RGB histogram is a plot that reflects the distribution of the pixel intensities for red, green, and blue color channels within an image.

For each color, the graph shows how many pixels have a certain intensity value, ranging from 0 (No intensity) to 255 (Full intensity). By comparing the average histograms across the different classes of banana leaf disease, one can observe differences in color distributions that might be characteristic of each disease, and a guide to feature engineering in classification of images[48].

For instance:

- Healthy leaves are prone to show a dominant green peak.
- Diseased leaves might show any of these, or a combination:
  - A shift in the green peak to lower intensities or a broader distribution.
  - Increased activity in the red channel (indicating reddish or brownish spots).
  - Potential changes in the blue channel, although often less pronounced for typical leaf diseases.

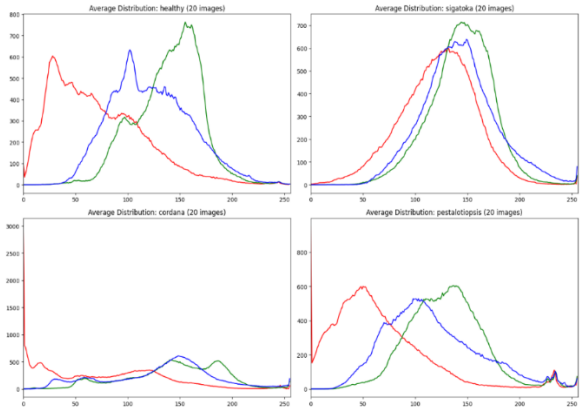

Fig.3 Average RGB Histograms for each of the classes: 'healthy', 'sigatoka', 'cordana', and 'pestalotiopsis'

XI. TRADITIONAL CLASSIFIERS: RANDOM FOREST AND SVM

Table12 Random Forest classification performance (on the held-out test set)

Class/Measure	Precision	Recall	F1-score	Support
Cordana	0.93	0.84	0.89	32
Healthy	0.79	0.85	0.81	26
Pestalotiopsis	0.83	0.71	0.77	35
Sigatoka	0.91	0.97	0.94	95
Accuracy	-	-	0.8830	188
Macro avg.	0.87	0.84	0.85	188
Weighted avg.	0.88	0.88	0.88	188

Table13 SVM classification performance (on the held-out test set).

Class/Measure	Precision	Recall	F1-score	Support
Cordana	0.97	0.91	0.94	32
Healthy	0.79	0.85	0.81	26
Pestalotiopsis	0.87	0.74	0.80	35
Sigatoka	0.92	0.97	0.94	95
Accuracy	-	-	0.8989	188
Macro avg.	0.88	0.87	0.87	188
Weighted avg.	0.90	0.90	0.90	188

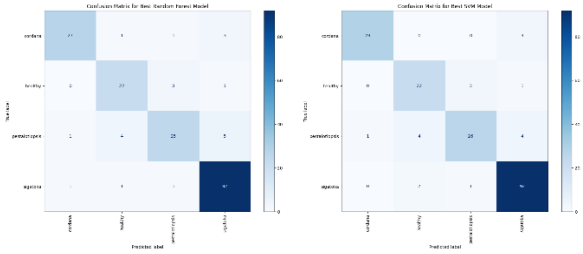

Figure 4 Confusion Matrix for Random Forest and SVM

XII. DEEP LEARNING: EFFICIENTNETV2, AND YOLO (YOLOV8N)

Table 14 EfficientNetV2 classification performance on the validation set

Class/Measure	Precision	Recall	F1-score	Support
Cordana	0.95	0.92	0.94	39
Healthy	0.96	1.00	0.98	22
Pestalotiopsis	0.97	0.95	0.96	39
Sigatoka	0.98	0.99	0.98	88
Accuracy	-	-	0.9681	188
Macro avg.	0.96	0.97	0.96	188
Weighted avg.	0.97	0.97	0.97	188

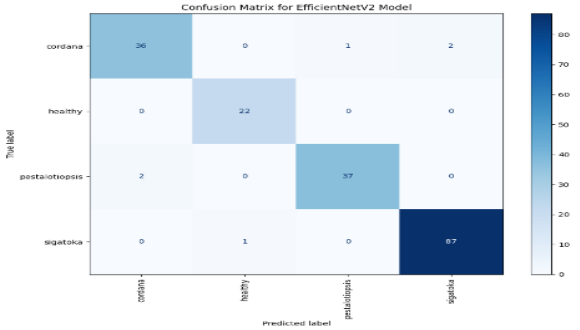

Figure 5 Confusion Matrix for EfficientNetV2 model

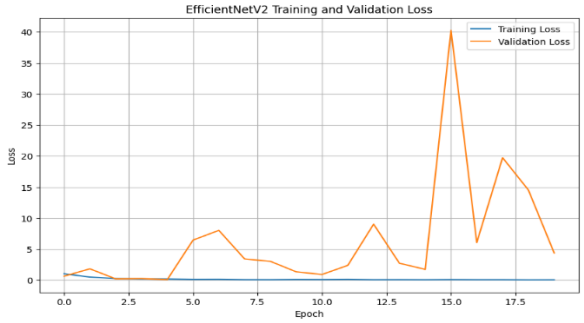

Figure 6: EfficientNetV2 Training and Validation Loss Curve

XIII. YOLOV8N (OBJECT DETECTION/CLASSIFICATION)

- mAP50: 0.948 and mAP50-95: 0.948, across different Intersection over Union (IoU)
- Class-wise mAP50 scores: Healthy: 0.847, Sigatoka: 0.994, Cordana: 0.976, and Pestalotiopsis: 0.976.

The high mAP values signified strong agreement between YOLOv8n predictions and the generated full-image bounding-box annotations used in this experiment, because each ground-truth box covered the complete image. The results demonstrate the suitability of YOLOv8n for this derived detection framework; however, they do not constitute an evaluation of individual lesion-level detection accuracy.

The mAP50 metric represents mean Average Precision at an IoU threshold of 0.50. The mAP50-95 metric represents mean Average Precision averaged over IoU thresholds from 0.50 to 0.95 at increments of 0.50. Since the latter incorporates progressively stricter localization criteria, mAP50-95 provides a more comprehensive assessment of detection performance. Both are reported independently and are not interchangeable.

Table15 Verified YOLOv8n validation metrics

Metric	Definition	Verified value
Precision	Proportion of predicted detections that were correct	Associated with mAP
Recall	Proportion of ground-truth objects correctly detected	Associated with mAP
mAP50	Mean Average Precision at IoU=50	0.948
mAP50-95	Mean Average Precision averaged across IoU=0.50-0.95	0.948

Note: The reported YOLOv8n values were taken from the final validation output of the selected training run. mAP50 and mAP50-95 are reported separately because they represent different IoU evaluation criteria.

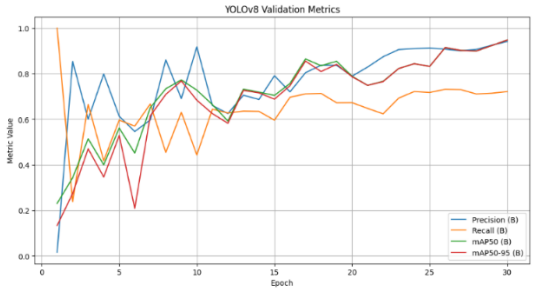

Figure 7 YOLOv8n Validation Metrics Curve

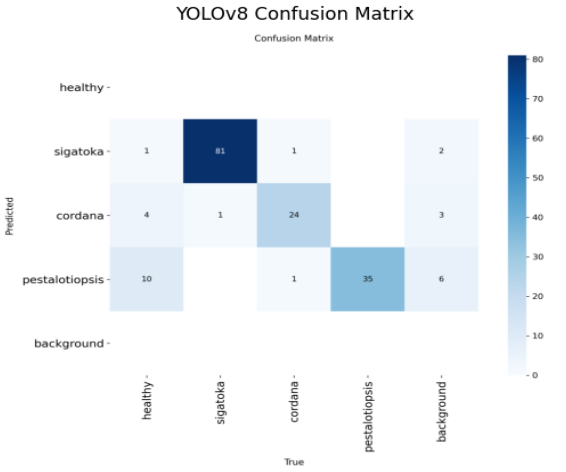

Figure 8 YOLOv8n Confusion Matrix

XIV. CONSISTENT AGGREGATE CLASSIFICATION PERFORMANCE

Macro-F1-score and weighted F1-score were considered important to address the imbalance in the dataset and ensure consistency in aggregate classification performance. Macro F1-score is the unweighted mean of the F1-score obtained from all the classes; it assigns equal importance to all classes and thus is an indication of a balanced performance. In contrast, weighted F1-score accounts for the number of samples in each class by weighting the class-specific F1-scores in accordance with the respective support; it is a reflection of overall performance based on class frequency.

Contrary to classification model performance evaluation, YOLOv8n was evaluated under distinct metrics: the object-detection framework mAP50 and mAP50-95, for which macro F1-score and weighted F1-score are not significant.

Table16. Consistent Classification Performance of Models

Model	Evaluation subset	Accuracy	Macro F1-score	Weighted F1-score
Random Forest	Held-out test set	0.8830	0.85	0.88
SVM	Held-out test set	0.8989	0.87	0.90
EfficientNetV2	Validation set	0.9681	0..96	0.97

Table 17. Task-specific performance of the evaluated models

Table 17:1 classification accuracy of image classification models.

Models	Random Forest	SVM	EfficientNetV2
Accuracy%	88.30	89.89	96.81

Table 17:2 YOLOv8n performance under the derived image-level object detection formulation

mAP50	0.948
mAP50-95	0.948

Table 17 presents classification and detection model results separately, because the models were evaluated using different task- specific metrics. For classification models, accuracy quantifies, which quantify correct image-level assessment (Table17:1), while mAP, applicable to YOLOv8n performance evaluation, represents detection performance with the full-image bounding box, not a manually annotated lesion. The two metrics are distinct from each other.

XV. COMPARATIVE DISCUSSION OF ALL THE MODELS.

The design of this study is such that different tasks are allotted to different models and are therefore evaluated based on task-specific performance. The dataset partitioning and evaluation procedure are explicitly reported in Table 1, with a clear distinction between classification and detection tasks and evaluation. For instance, Random Forest achieved an accuracy of 0.8830, a macro F1-score of 0.85, and a weighted F1-score of 0.88. The SVM achieved an accuracy of 0.8989, a macro F1-score of 0.87, and a weighted F1-score of 0.90. EfficientNetV2 achieved an accuracy of 0.9681, a macro F1-score of 0.96, and a weighted F1-score of 0.97.

The difference between macro and weighted F1-scores for Random Forest and SVM suggests that the aggregate performance measures were affected by class frequency.[28] The weighted F1-score was higher than the macro F1-score for both models, indicating better performance on some classes with higher representation than the lower class-specific F1-score for Pestalotiopsis and Healthy, as compared to Sigatoka [49].

EfficientNetV2 had a smaller difference between macro F1- score (0.96) and weighted F1-score (0.97), indicating more balanced class-wise performance on the reported validation subset. However, these results have to be considered in conjunction with the class-specific precision, recall, F1-scores, and the evaluation protocol, and not just a single aggregate metric. [50] AI algorithms are undoubtedly emerging tools for real-life applications and provide farmers with quick and accurate diagnosis of plant diseases.

The identical values found in the present experiment should be interpreted in the light of the adopted full-image bounding-box annotation strategy. Each image had one ground-truth bounding box covering the entire image with normalized coordinates (0.5, 0.5, 1.0, and 1.0). Therefore, the localization part of the detection task was greatly simplified relative to traditional lesion-level object detection. The reported mAP50 = 0.948 and mAP50-95=0.948 equality shows that in this particular full-image detection formulation, the model was able to retain the reported Average Precision across the evaluated IoU thresholds.

XVI. SUMMARY OF TASK-SPECIFIC PERFORMANCE

The results of the experimentation for classification models revealed that EfficientNetV2 achieved the highest accuracy of 96.81%, reflecting strong image-level disease classification, unlike SVM at 89.89% accuracy and Random Forest at 88.30%. YOLOv8n, based on object detection metrics, reported mAP50= 0.948 and mAP50-95 = 0.948 with respect to the full-image bounding box formulation adopted for this study.

In the final analysis, EfficientNetV2 is the best-performing model for the image classification task with respect to accuracy. While YOLOv8n, mAP based on the class prediction and the ground-truth boxes at a given Intersection over Union (IoU)

thresholding demonstrated credible performance. There is a clear distinction between the two metrics, and they are not numerically comparable.

XVII. LIMITATIONS OF THE STUDY

There are several limitations of interest in the interpretation of the reports of this study, among which is the imbalance in class distribution of the BananaLSD dataset, which comprises Healthy 143 images, Sigatoka 473 images, Cordana 162 images, and Pestalotiopsis 173 images, respectively. Which necessity the inclusion of macro F1-score and weighted F1-score in the report.

A limitation in YOLOv8n experimentation is the simplification of the localization task because the bounding boxes were automatically generated for full images, rather than manually annotated for individual disease lesions or spots. Though, the strategy made it possible to use an image-level disease classification dataset within an object detection architecture; consequently, the mAP50 and mAP50-95 should not be read as direct evidence of precise lesion-level localization.

For future work, we hope to use field-acquired datasets, with the view to assess robustness considering varying field conditions such as background, geographical location, amongst others. As part of prospects, we suggest evaluation and external validation in the proposed deployment field, in order to translate experimental results into practical use, including generalization beyond public datasets.

Finally, the present study should therefore be read as establishing a comparative experimental baseline for banana leaf disease recognition.

17.1 Limitations to Real-World Application

The experimental results of this study deserve exercise of constraint in application to real-world settings, though the model's performance was credible using the BananaLSD dataset. The reported performance reflects generalization over held-out samples taken from the experimental dataset, and not samples from an independent field; consequently, the reported results should therefore be taken as evidence of good performance, with potential for real-world applications, but not as confirmation of operational robustness, especially since external evaluation and validation are not conducted through an independent dataset. This likely introduces variations, such as lighting, background effects, and leaf orientation, different geographical locations, and many more in the dataset.

For future work, we hope to make use of field-acquired datasets, with diverse field conditions such as illumination, lighting, and banana cultivars, among others, for robustness and generalization assessment beyond public datasets. For object detection, we hope to carry out manual annotation of the dataset; the outcome of this study may be taken further in the development of a computer-assisted disease monitoring system after conducting proper field validation.

XVIII. CONCLUSION

This study presents a comparative investigation of handcrafted texture-based machine learning and deep feature learning using BananaLSD dataset for banana leaf disease recognition. The primary contribution of this work is the systematic evaluation of four feature learning paradigms, namely conventional machine learning GLCM-based Random Forest and GLCM-based Support Vector Machine, EfficientNetV2, and YOLOv8n in the same application domain. For the classification task, EfficientNetV2 scored the highest accuracy of 96.81%, followed by SVM at 89.89% accuracy, and Random Forest at 83.30%. YOLOv8n, evaluated with mAP, reports mAP50 of 0.948 and mAP50-95 of 0.948, respectively. mAP50=0.948 and mAP50-95=0.948 values are an indication of strong performance with respect to the derived full-image bounding box annotation strategy adopted in this study, but should not be read as direct evidence of precise lesion-level localization. Furthermore, both classification accuracies and mAP represent different task assessments; therefore, there is no basis for numerical comparison. In summary, EfficientNetV2 with the highest accuracy is the most suitable for the classification task compared to SVM and Random Forest. But YOLOv8n equally demonstrated strong performance in the object detection task as adopted for this study. The reported results emphasize the effectiveness of automatically learned deep image representation in the classification task over the handcrafted GLCM texture feature used by conventional machine-learning models. In addition, the choice between individual approaches should be

based on intended use and evaluation objectives, and not necessarily on accuracy and mAP. Future studies, we suggest manual annotation of the lesion level, and larger and more a larger dataset, multi-object detection, and extension of the study through hybrid or ensemble models. Evaluation and validation in practical fields.

REFERENCE

[1] R. Saxena *et al*., "Genome editing: A new age technology for sustainable agriculture and developing abiotic stress resilience in crops," vol. X, no. XX, pp. 1–11, 2025, doi: 10.7324/JABB.2025.232158.

[2] C. Series, "Deep learning and its role in smart agriculture," doi: 10.1088/1742-6596/1399/4/044109.

[3] B. Chin, "Can banana be a success story for Malaysia?" vol. 9, no. 1, pp. 13–22, 2022, doi: 10.56527/jabm.9.1.2.

[4] B. Evan, L. Batu, W. A. Saputra, and A. Sa, "Banana and orange classification detection using convolutional neural network," Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI), vol. 13, no. 3, pp. 561–570, 2024.

[5] M. R. Sheikh, M. A. Hossain, M. Hossain, M. M. Islam, and G. M. S. Himel, "BananaSet: A dataset of banana varieties in Bangladesh," Data in Brief, vol. 54, Art. no. 110513, 2024, doi: 10.1016/j.dib.2024.110513.

[6] C. A. Elinisa et al., "Image segmentation deep learning model for early detection of banana diseases," Applied Artificial Intelligence, vol. 39, no. 1, 2025, doi: 10.1080/08839514.2024.2440837.

[7] A. Nayak, S. Chakraborty, and D. K. Swain, "Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection," Smart Agricultural Technology, vol. 4, Art. no. 100195, 2023, doi: 10.1016/j.atech.2023.100195.

[8] A. Mitra et al., "Cotton yield prediction: A machine learning approach with field and synthetic data," IEEE Access, vol. 12, pp. 101273–101288, 2024, doi: 10.1109/ACCESS.2024.3418139.

[9] 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, doi: 10.2478/acss-2023-0009.

[10] A. Prasetyo and E. Utami, "Detection and classification of banana leaf diseases: Systematic literature review," Telematika, vol. 17, no. 2, pp. 128–141, 2024.

[11] W. B. Demilie, "Plant disease detection and classification techniques: A comparative study of the performances," Journal of Big Data, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00863-9.

[12] K. V. Reddy et al., "Edge AI in sustainable farming: Deep learning-driven IoT framework to safeguard crops from wildlife threats," IEEE Access, vol. 12, pp. 77707–77723, 2024, doi: 10.1109/ACCESS.2024.3406585.

[13] R. Jeyachandra and D. Vasumathi, "Banana leaf disease prediction using convolutional neural networks," vol. 33, no. 8, pp. 1881–1894, 2024.

[14] S. M. Hassan et al., "A survey on different plant diseases detection using machine learning techniques," Electronics, vol. 11, no. 17, pp. 1–29, 2022, doi: 10.3390/electronics11172641.

[15] N. B. Devadiga and K. N. Akshatha, "GLCM and LSTM recurrent neural networks integrated with machine learning techniques to identify plant disease," vol. 3075, no. 9, pp. 44–46, 2022, doi: 10.35940/ijitee. G9243.0811922.

[16] I. Ahmed and P. K. Yadav, "A systematic analysis of machine learning and deep learning-based approaches for identifying and diagnosing plant diseases," Sustainable Operations and Computers, vol. 4, pp. 96–104, 2023, doi: 10.1016/j.susoc.2023.03.001.

[17] C. C. Ukwuoma, Q. Zhiguang, B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent advancements in fruit detection and classification using deep learning techniques," vol. 2022, 2022, doi: 10.1155/2022/9210947.

[18] S. A. Wagle, R. Harikrishnan, V. Varadarajan, and K. Kotecha, "A new compact method based on a convolutional neural network for classification and validation of tomato plant disease," Electronics, vol. 11, no. 19, 2022, doi: 10.3390/electronics11192994.

[19] A. Patel, M. Rajarajan, N. Kesswani, R. Misra, and V. Bharadwaj, Internet of Things and

Connected Technologies, 2021, p. 98, doi: 10.1007/978-3-030-94507-7.

[20] M. Shoaib et al., "An advanced deep learning models-based plant disease detection: A review of recent research," Frontiers in Plant Science, vol. 14, pp. 1–22, 2023, doi: 10.3389/fpls.2023.1158933.

[21] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," SN Computer Science, vol. 2, no. 6, pp. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.

[22] M. S. Memon, P. Kumar, and R. Iqbal, "Meta deep learn leaf disease identification model for cotton crop," Computers, vol. 11, no. 7, 2022, doi: 10.3390/computers11070102.

[23] E. S. Dreier, R. Malm, L. K. Hansen, and T. Lenka, "Challenges in explaining deep learning models for data with biological variation," pp. 1–29.

[24] K. Neupane and F. Baysal-Gurel, "Automatic identification and monitoring of plant diseases using unmanned aerial vehicles: A review," 2021.

[25] J. J. Mora, M. G. Selvaraj, C. I. Alvarez, N. Safari, and G. Blomme, "From pixels to plant health: Accurate detection of banana Xanthomonas wilt in complex African landscapes using high-resolution UAV images and deep learning," Discover Applied Sciences, vol. 6, no. 7, 2024, doi: 10.1007/s42452-024-06073-z.

[26] E. Ergün, "High precision banana variety identification using vision transformer-based feature extraction and support vector machine," pp. 1–16, 2025.

[27] H. N. Ngugi, A. E. Ezugwu, A. A. Akinyelu, and L. Abualigah, "Revolutionizing crop disease detection with computational deep learning: A comprehensive review," vol. 196, no. 3, 2024, doi: 10.1007/s10661-024-12454-z.

[28] M. G. Selvaraj et al., "AI-powered banana diseases and pest detection," Plant Methods, pp. 1–11, 2019, doi: 10.1186/s13007-019-0475-z.

[29] K. L. Narayanan et al., "Banana plant disease classification using hybrid convolutional neural network," vol. 2022, 2022, doi: 10.1155/2022/9153699.

[30] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," pp. 1–10, 2023, doi: 10.3389/fpls.2023.1308528.

[31] S. Meinam, K. Nongmeikapam, and N. B. Singh, "Optimized machine learning framework for detection of banana leaf disease," vol. 10, pp. 142–155, 2025.

[32] R. Linero-Ramos, C. Parra-Rodr, A. Espinosa-Valdez, G. Jorge, and M. Gongora, "Assessment of dataset scalability for classification of black Sigatoka in banana crops using UAV-based multispectral images and deep learning techniques," 2024.

[33] A. Siddig, "Banana leaf disease detection with multi feature extraction techniques using SVM," Sai Rajasekhar Reddy Evuri, National College of Ireland, Supervisor: [details not provided].

[34] J. Karim, O. F. Goni, M. Ahsan, J. Haider, and M. Kowalski, "Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM," Scientific Reports, pp. 1–24, 2024, doi: 10.1038/s41598-024-66989-9.

[35] "Early detection of banana leaf disease using novel deep convolutional neural network," pp. 1–8, 2024, doi: 10.47852/bonviewJDSIS42021530.

[36] B. Oviedo, C. Zambrano-Vega, R. O. Villamar-Torres, D. Yánez-Cajo, and K. C. Campoverde, "Improved YOLOv8 segmentation model for the detection of Moko and black Sigatoka diseases in banana crops with UAV imagery," pp. 1–24, 2025.

[37] D. Mathew and C. S. Kumar, "Classification of leaf spot diseases in banana using pre-trained convolutional neural networks," 2023, doi: 10.1109/ICCC57789.2023.10165629.

[38] J. B. Kakati and T. K. Das, "A convolutional fusion architecture for processing smartphone-based images to detect banana leaf diseases," vol. 10, no. 4, pp. 177–191, 2025.

[39] M. Hernández-Restrepo, J. Z. Groenewald, and P. W. Crous, "Neocordana gen. nov., the causal organism of Cordana leaf spot on banana," 2015, doi: 10.11646/phytotaxa.205.4.2.

[40] A. S. Ibrahim et al., "AI-IoT based smart agriculture pivot for plant diseases detection and treatment," pp. 1–16, 2025.

[41] A. Jadhaw and R. Bhandari, "Detection of banana leaf disease and its analysis using different

techniques," vol. 13, no. 1, pp. 18–27, 2024, doi: 10.4172/2347-226X.13.1.005.

[42] M. Badiger, V. Kumara, S. C. N. Shetty, and S. Poojary, "Leaf and skin disease detection using image processing," 2025, doi: 10.1016/j.gltp.2022.03.010.

[43] A. Jafar, N. Bibi, R. A. Naqvi, A. Sadeghi-Niaraki, and D. Jeong, "Revolutionizing agriculture with artificial intelligence: Plant disease detection methods, applications, and their limitations," Frontiers in Plant Science, vol. 15, pp. 1–20, 2024, doi: 10.3389/fpls.2024.1356260.

[44] M. T. Scholar, "Integrated GLCM texture features and CNN for automated cotton disease identification," vol. 102, no. 21, pp. 7589–7600, 2024.

[45] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, no. 3, pp. 1636–1651, 2019, doi: 10.3906/elk-1809-181.

[46] J. D. Thiagarajan et al., "Analysis of banana plant health using machine learning techniques," Scientific Reports, vol. 14, no. 1, pp. 1–23, 2024, doi: 10.1038/s41598-024-63930-y.

[47] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, "Computer vision system for mango fruit defect detection using deep convolutional neural network," Foods, vol. 11, no. 21, 2022, doi: 10.3390/foods11213483.

[48] S. Vijayan and C. L. Chowdhary, "Hybrid feature optimized CNN for rice crop disease prediction," Scientific Reports, vol. 15, no. 1, pp. 1–20, 2025, doi: 10.1038/s41598-025-92646-w.

[49] D. Y. P. Agriculture and D. Y. P. Agriculture, "Optimized sequential model for superior classification of plant disease," pp. 1–12, 2025.

[50] A. C. Bhowmik, D. M. T. Ahad, and Y. R. Emon, "Machine learning-based Jamun leaf disease detection: A comprehensive review," 2023.