Smart Agricultural Technology 4 (2023) 100214

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases

Md. Abdullahil Baki Bhuiyan [a], Hasan Muhammad Abdullah [b,*], Shifat E. Arman [c,*], Sayed Saminur Rahman [d], Kaies Al Mahmud [c]

[a] Department of Plant Pathology, Bangabandhu Sheikh Mujibur Rahman Agricultural University, Gazipur 1706, Bangladesh
[b] GIS and Remote Sensing Lab, Department of Agroforestry and Environment, Bangabandhu Sheikh Mujibur Rahman Agricultural University, Gazipur 1706, Bangladesh
[c] Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka-1000, Bangladesh
[d] Department of Computer Science and Engineering, APJ Abdul Kalam Technological University, Kerala-695016, India

ARTICLE INFO

Editor: Spyros Fountas

Keywords:
Deep learning
Convolutional neural networks (CNN)
Pestalotiopsis
Sigatoka
Cordana
Banana leaf diseases
Bayesian optimization

ABSTRACT

All over the world, bananas are one of the most common fruits. It accounts for nearly 16% of global fruit production. However, every year, a large amount of banana yield losses occur due to different diseases of the banana leaf. It is essential to identify these diseases at an early stage in order to increase banana production. A visual inspection is the most common method of identifying banana leaf diseases. With a visual inspection, errors are common, time is a factor, and expertise is required. This study shows how deep learning and Bayesian optimization can be used to effectively diagnose banana leaf diseases from images without any human intervention. We collected the Banana Leaf Spot Diseases (BananaLSD) dataset from various locations in Bangladesh. The dataset consists of images of three banana leaf diseases: Pestalotiopsis, Sigatoka, and Cordana. Our proposed BananaSqueezeNet model performed exceptionally well in diagnosing banana leaf diseases from the images with an overall accuracy of 96.25%, precision of 96.53%, recall of 96.25%, specificity of 98.75%, F1-score of 96.17%, and MCC of 95.13%. The BananaSqueezeNet model outperforms some state-of-the-art convolutional neural networks that include EfficientNetB0, MobileNetV3, ResNet-101, ResNet-50, InceptionNet-V3, and VGG16. The BananaSqueezeNet model also detected seven other diseases that affect banana leaves, fruits, and stems, including banana fruit scarring beetle, black sigatoka, bacterial soft rot, pseudo stem weevil, yellow sigatoka, banana aphids, and panama disease, with an accuracy of 95.13%. BananaSqueezeNet will enable banana growers to detect banana diseases early, and we hope that it will ultimately lead to an increase in banana production in Bangladesh and around the world.

1. Introduction

Bananas are grown in over 130 countries, primarily in tropical and subtropical regions. Their origin can be traced back to South-East Asia [1]. Bananas are a highly sought-after staple food, accounting for nearly 16% of global fruit production and ranking as the second largest fruit behind citrus. In terms of world trade, bananas are the fifth most important food crop, following coffee, cereals, sugar, and cacao [2]. There are two main types of bananas: sweet or dessert bananas and kitchen bananas or plantains. They can be consumed raw or processed, and contain various bioactive molecules such as phenolics, carotenoids, biogenic amines, and phytosterols, which are beneficial for human health. They also have high levels of antioxidants. Historically, bananas have been used to treat various chronic degenerative disorders. The World Health Organization recommends that individuals consume 400 grams of fruit and vegetables per day [3]. To meet the global demand, bananas are produced in many countries. India is the largest producer of bananas, accounting for 27% of global production [2]. Global banana production totals 128,778,738 tonnes from an area of 5,517,027 hectares. Bangladesh produces 833,309 tonnes of bananas from 48,850 hectares of land [4].

Many diseases frequently infect banana crops. There are multiple pathogens responsible for leaf spot diseases worldwide, which cause severe yield losses every year. Some commonly reported leaf spot diseases include sigatoka diseases such as black sigatoka (*Mycosphaerella fijiensis*), yellow sigatoka (*Pseudocercospora musicola*), eumusae leaf spot

* Corresponding authors.
E-mail addresses: hasan.abdullah@bsmrau.edu.bd (H.M. Abdullah), shifatearman@du.ac.bd (S.E. Arman).

https://doi.org/10.1016/j.atech.2023.100214
Received 30 December 2022; Received in revised form 20 February 2023; Accepted 4 March 2023
Available online 9 March 2023

(*Mycosphaerella eumusae*) [5,6], exserohilum leaf spot (*Exserohilum rostratum*) [7], cordana leaf spot (*Cordana musae*) [8], plantain zonate leaf spot (*Pestalotiopsis menezesiana*) [9], banana freckle disease (*Phyllosticta musarum*) [10]. An assessment of the banana leaf blight fungal population was recently conducted in several areas of Bangladesh. Consequently, a new banana leaf blight (*pestalotiopsis microspora*) was identified in Bangladesh [11]. Unlike the zonate leaf spot, the initial symptom of this new leaf spot was found as a narrow dark brown lesion, which later on turned into an irregular brown spot with a golden yellowish margin.

Plant diseases largely limit banana production. In order to curb the disease's progression, it is crucial to assess the severity of the disease in order to take proper control measures. Traditionally, plant pathologists estimate plant disease severity by visually inspecting the disease symptoms. Unfortunately, this technique is ineffective and very expensive if a large area is to be covered. Agriculturists are increasingly using automated disease diagnosis models because of the advent of digital cameras and computer technology. In recent times, the diagnosis of plant disease severity has been undertaken by deep learning image-based automatic analysis [12]. In Bangladesh, Artificial Intelligence (AI) and remote sensing technology have been used to detect crop diseases such as wheat blast [13]. However, the use of deep-learning techniques for detecting plant diseases is not very common in Bangladesh, despite their widespread use in smart agriculture globally. In Tanzania, deep learning techniques such as Vgg16, Resnet18, Resnet50, Resnet152, and InceptionV3 were used to detect fusarium wilt and black sigatoka in banana [14]. In Germany, LeNet Convolutional Neural Network (CNN) architecture was used to classify banana leaf diseases like sigatoka and speckle [15].

In this study, we proposed a very fast and lightweight CNN to diagnose three prominent banana leaf diseases that include Pestalotiopsis, Sigatoka, and Cordana leaf spot of banana leaf. We optimized four CNN architectures with Bayesian Optimization to achieve better results.

The main contributions of this study are:

1) **Banana Leaf Spot Diseases (BananaLSD) Dataset**: We compiled a dataset that includes images of banana leaves with three major banana leaf diseases, including sigatoka, pestalotiopsis, and cordana, as well as healthy leaves. The dataset was collected from Bangabandhu Sheikh Mujibur Rahman Agricultural University (BSMRAU) experimental field and different banana fields located in Bangladesh.
2) **BananaSqueezeNet**: We proposed a very fast and lightweight CNN architecture optimized with Bayesian Optimization titled BananaSquezeeNet that can identify three prominent banana leaf diseases from banana leaf images. The BananaSqueezeNet model can also identify diseases of banana fruit and stem.
3) **Smartphone Application for Banana Leaf Disease Identification**: We developed a smartphone application that can diagnose banana leaf diseases from real-world images of banana leaves, making it convenient for farmers and other stakeholders. The application will allow farmers to take precautionary steps to prevent banana yield loss by stopping the spread of disease.

The rest of the paper is organized as follows: In Section 2, we discussed the relevant literatures. In Section 3, we described the dataset and how it was collected. Then we discussed how we optimized the neural networks to identify banana leaf diseases using Bayesian optimization. In Section 4, the experimental results are presented. In Section 5, we discussed the findings of this research. Finally, the paper is concluded in Section 6.

2. Related works

In order to properly apply control measures and manage plant diseases, early disease detection is critical. There are several approaches for detecting plant diseases. One of the most common approaches is to visual detection. The process of visually identifying plant diseases, however, is labor-intensive and less accurate. One alternative is to perform laboratory analysis. However, this method is time-consuming, requires technical knowledge, and lab facilities, which are not available to many farmers in underdeveloped countries [16]. Aside from this conventional methods, AI has also been used to detect plant diseases in recent decades [17]. Object recognition and image classification were previously performed using features obtained from different feature extraction algorithm that includes Scale-Invariant Feature Transform (SIFT) [18], Histogram of Oriented Gradients (HoG) [19], and Speeded Up Robust Features (SURF) [20]. However, the automatic recognition of objects and the classification of images have progressed immensely in recent years with the use of Convolutional Neural Networks (CNN) [21,22]. In 2014, Hernández-Rabadán et al. developed a method for segmenting diseased plants in uncontrolled environments using a combination of SOMs and Bayesian Classifiers [23].

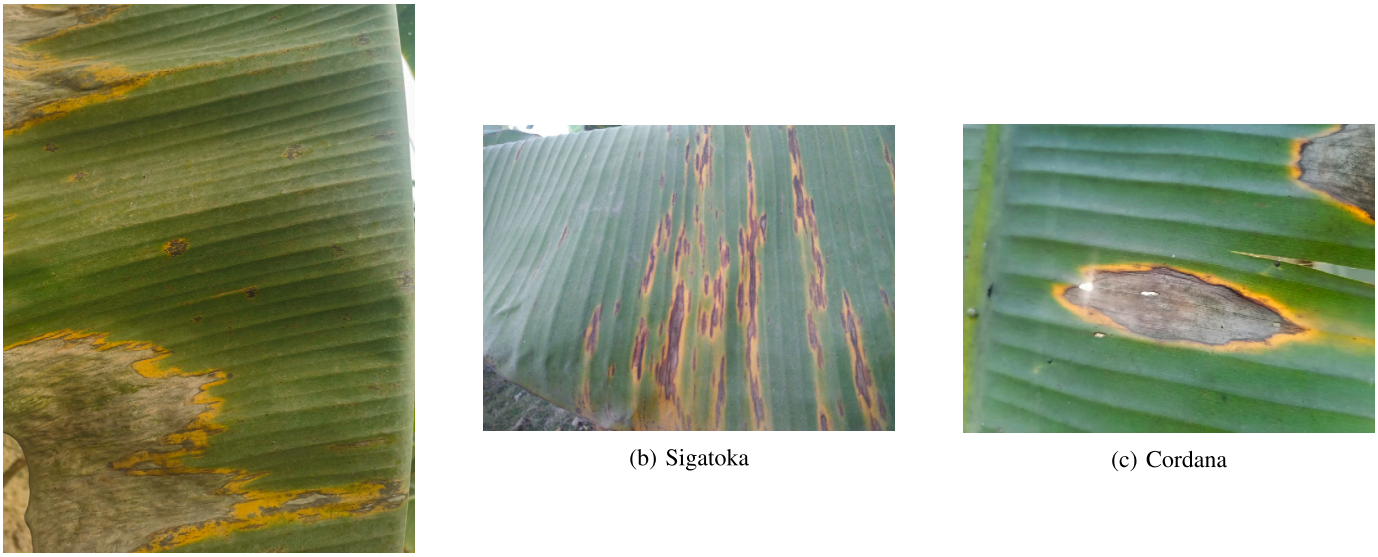
In 2022, Bhuiyan et al. [24] first reported the occurrence of Pestalotiopsis microspora causing leaf blight of banana in Bangladesh. In 2022, Medhi & Deb [25] released a dataset named PSFD-MUSA that contains images of different diseases associated with banana leaves, fruits, and stems. Banana plants and their major diseases can be detected from aerial images using different machine learning methods [26]. Deep transfer learning based method to detect banana pests and disease symptoms from banana field images [27]. Sujithra et al. used different deep neural networks that include Convolution Neural Network (CNN), Radial Basis Neural Network (RBNN), and Feed-Forward Neural Network (FFNN) to diagnose leaf diseases of banana and sugarcane [28]. The proposed models can classify black and yellow Sigatoka diseases of banana leaves. Banana leaf diseases can also be detected using Gabor Extraction and Region-Based Convolution Neural Network (RCNN) [29]. Gopinath et al. presented an automated big data framework for classifying plant leaf disease using a Convolutional Recurrent Neural Network Classifier (CRNN) algorithm [30]. Using their method, they were able to differentiate between healthy and unhealthy leaves of different plants that include bananas, peppers, potatoes, and tomatoes. In order to achieve good performance on a particular task, optimization or hyperparameter tuning is very important. Several optimization techniques are reported in the literature that can be used to optimize neural networks [31–34]. Arman et al. [35] showed how Bayesian optimization and deep learning can be used effectively diagnose COVID-19 from chest X-ray images. Doke et al. showed how CNN and Bayesian optimization can be used to identify cerebral micro-bleeds [36].

In this study, we proposed a fast and lightweight convolutional neural network named BananaSqueezeNet that can diagnose banana leaf diseases from images of banana leaves. As the BananaSqueezeNet model is very small, it can be easily deployed in embedded devices. The model can identify three banana leaf diseases - Sigatoka, Cordana leaf spot, and Pestalotiopsis leaf blight diseases in real-time using a smartphone camera. We also proposed a new dataset containing images of different disease affected banana leaves titled Banana Leaf Spot Diseases (BananaLSD) dataset.

3. Materials and methods

3.1. Dataset description

The images of the banana leaves in the Banana Leaf Spot Diseases (BananaLSD) dataset were collected from the Bangabandhu Sheikh Mujibur Rahman Agricultural University (BSMRAU) experimental field and different banana fields located in Bangladesh. Multiple smartphones were used to take the images. There are 937 images of four classes: healthy, Pestalotiopsis leaf blight, Sigatoka, and Cordana. These images were then labeled as one of the four classes by an expert plant pathologist. In the train set, there were 89 images of healthy leaves, 133 images of leaves affected by Pestalotiopsis leaf blight, 433 images of leaves



(a) Pestalotiopsis (b) Sigatoka (c) Cordana

Fig. 1. Banana leaves containing three prominent banana leaf diseases.

Table 1
Class distribution of images in the BananaLSD dataset.

	Raw Dataset			Augmented Dataset		
Class	Train	Validation	Test	Train	Validation	Test
Healthy	89	20	20	400	20	20
Pestalotiopsis	133	20	20	400	20	20
Sigatoka	433	20	20	400	20	20
Cordana	122	20	20	400	20	20
Total	**777**	**80**	**80**	**1600**	**80**	**80**

affected by Sigatoka disease, and 122 images of leaves affected by Cordana disease. There are 80 images in the validation set, and each class has 20 images. The test set also has 80 images, 20 images in each class. We also performed several augmentation techniques that include Gaussian blur, horizontal flip, cropping, linear contrast, shear, translate, and rotate shear. Augmentation will help to address the data imbalance issue and incorporate diversity into the data. These augmentations were performed only on the training set. The validation and the test set were kept the same. There are 400 images in each category of the training set in the augmented dataset.

Table 1 illustrates the number of images of different classes in the train, validation, and test set.

Random samples of images of Pestalotiopsis, Sigatoka and Cordana diseases from the BananaLSD dataset are shown in Fig. 1.

3.2. Measurement of performance

Several performance metrics have been used to measure the performance of our models, including accuracy, precision, recall, the F1-Score, and Receiver Operating Characteristics (ROC). The following equations were used to calculate accuracy, precision, recall, and F1-score:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

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

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} = \frac{2\times TP}{2\times TP+FP+FN} \tag{4}$$

The quantities of true positive, true negative, false positive, and false negative samples are represented by TP, TN, FP, and FN, respectively.

Receiver Operating Characteristic (ROC) curves plot the performance of binary classifier systems as their discrimination thresholds are adjusted. The area under the ROC curve (AUC) provides a measure of how capable the model is in terms of distinguishing between different classes.

3.3. Bayesian optimization

Training a deep neural network involves optimizing numerous parameters to minimize a loss function. While the primary goal of the training process is to maximize predictive performance by accurately estimating the network parameters, the training process is dependent upon many other parameters as well, termed as hyperparameters. These hyperparameters do not undergo any training and remain fixed throughout the process. The choice of hyperparameters greatly impact the overall training process, make convergence faster while also avoiding local minimas. However, deep learning models are expensive to train, and re-evaluating models with different hyperparameters are highly inefficient. To mitigate this issue, various optimization algorithms have been developed to find the best set of hyperparameters. Bayesian Optimization is one such algorithm that utilizes prior knowledge to determine the best candidate to evaluate in the next stage. Given a function $f : \mathbb{R}^d \mapsto \mathbb{R}$, Bayesian Optimization seeks to find the location $x \in \mathbb{R}^d$ that corresponds to the global maximum or minimum of f. Bayesian Optimization uses prior knowledge to obtain the posterior using the Bayes rule, which is then evaluated using an acquisition function to determine the next set of hyperparameters to test.

In this experiment, the Expected Improvement (EI) acquisition function was used. EI picks the next set of points that achieves the highest expected improvement over the current best $f(x^+)$ where $x^+ = argmax_{x_i \to x_{1:t}} f(x_i)$ and x_i is the queries at the i^{th} time step.

The $t+1^{th}$ point $x_{(t+1)}$ is selected using the equation:

$$x_{t+1} = argmin_x \mathbb{E}(||h_{t+1}(x) - f(x^*)|||D_t) \tag{5}$$

Here, f is the ground truth function, h_{t+1} is the posterior at $t+1^{th}$ timestep, D_t is the training data and x^* is the position where f takes its maximum value.

Thus, the function tries to select the point that minimizes the distance to the maximum of the objective function. However, as the ground

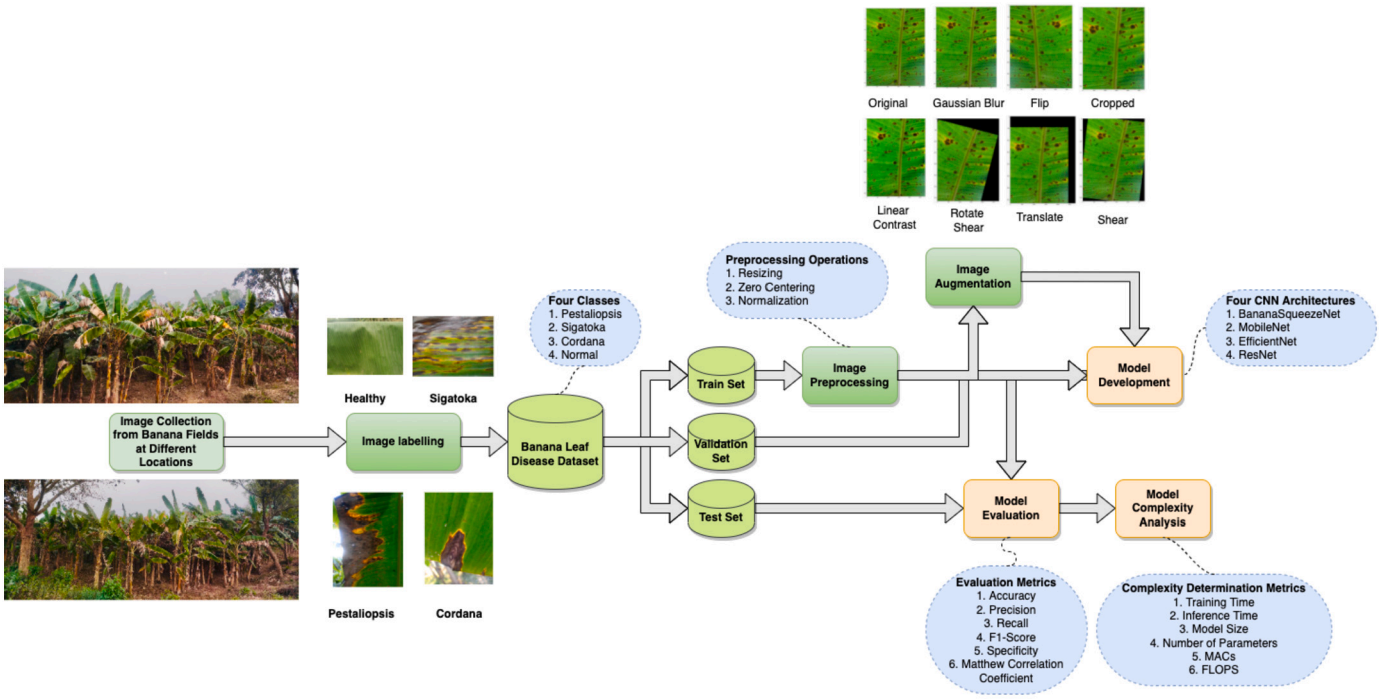


Fig. 2. Methodology of our work.

truth function f is unknown, a modified acquisition function was proposed.

$$x_{(t+1)} = argmax_x \mathbb{E}(max\{0, h_{t+1}(x) - f(x^+)\}|D_t) \tag{6}$$

Here, $f(x^+)$ is the maximum value observed thus far. The equation for GP surrogate is:

$$EI(x) = \begin{cases} (\mu_t(x) - f(x^+) - \epsilon)\Phi(Z) + \sigma_t(x)\phi(Z), & if\ \sigma_t(x) > 0 \\ 0, & if\ \sigma_t(x) > 0 \end{cases} \tag{7}$$

$$Z = \frac{\mu_t(x) - f(x^+) - \epsilon}{\sigma_t(x)}$$

Here, $\Phi(.)$ indicates the CDF and $\sigma(.)$ indicates the PDF. Hence, from the equations it can be observed that the EI will be high when the expected value of $\mu_t(x) - f(x^+)$ is high or when the uncertainty $\sigma_t(x)$ around a certain point is high.

Hence, to summarize, Bayesian Optimization utilizes an acquisition function to evaluate the effectiveness of a set of hyperparameters on a surrogate function, an estimate of the objective function, which is the true but unknown function. Bayesian optimization is most effective when the objective function is computationally expensive to evaluate repeatedly.

3.4. Our approach

The methodology of our work is presented in Fig. 2. At first, the data collection was done. The dataset consists of images of banana leaves of four categories. These are healthy leaves, Cordana affected leaves, Pestalotiopsis affected leaves, and Sigatoka affected leaves. Experts carefully labeled the images. A data cleaning step by an expert data scientist was taken to remove the inconsistent images. The dataset is then split into the train-validation-test sets. The training set consists of 777 images with four categories (Healthy: 89, Pestalotiopsis leaf blight: 133, Sigatoka: 433, and Cordana: 122). The validation and test sets consist of 20 images from each category. The images are preprocessed using different preprocessing techniques that include - resizing, zero-centering, and normalization.

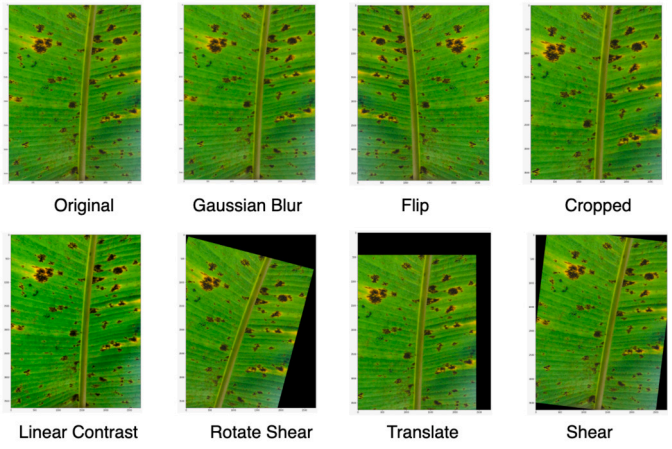


Fig. 3. Augmentation of banana leaf images.

In order to handle data imbalance and incorporate diversity in our data, we used several augmentation techniques that include - Gaussian blur, horizontal flip, cropping, linear contrast, shear, translate, and rotate shear. These augmentations were performed only on the training set during training. The validation and the test set were kept the same. A demonstration of the augmentations that we have performed in this work is shown in Fig. 3.

We used Convolutional Neural Network (CNN) to perform the classification task. CNN has two types of layers - feature extraction layers and classification layers. The feature extraction layers extract the features from the image, whereas the classification layer performs the classification of the image based on the extracted features. The CNN model size must be small since we intend to deploy it on a mobile device so that farmers can use it in their fields. For this reason, we trained some lightweight CNN models - SqueezeNet [37], MobileNet V3 [38], EfficientNet [39]. The benefit of using these lightweight models is that these models can be deployed in embedded devices as they are very small in size.

SqueezeNet [37] is a lightweight CNN architecture designed to preserve model accuracy with fewer parameters. Fire modules are used in order to achieve this. A fire module has a squeeze and expand layer. A squeeze layer has only a 1 x 1 filter, whereas an expand layer has both 1 x 1 and 3 x 3 filters. Many fire modules are stacked one over another with some convolution layer and skip connections to form the SqueezeNet architecture. One exciting thing about SqueezeNet is that it doesn't have any fully connected layer. This greatly helps to reduce the model parameters, and hence keeping the model size small. SqueezeNet reached AlexNet level accuracy even after having 50x fewer parameters and less than 0.5 MB model size.

MobileNets [38] are lightweight CNN architecture mainly designed for mobile and embedded device applications. Along with the convolution layers for feature extraction, MobileNets also use depthwise separable convolution [40], which significantly reduces the number of parameters in a neural network, thereby decreasing its size. MobileNet also uses two hyperparameters, namely - width multiplier and resolution multiplier, in order to shrink the models. In our experiments, we used the MobileNet V3 [41], which was designed by bringing some iterative improvements on MobileNet V1 [41] and MobileNets V2 [42]. In MobileNet V3, some expensive layers are redesigned, and a squeeze-and-excitation block [43] was introduced.

EfficientNets [39] are a family of neural network architectures designed using neural architecture search technique. Various types of scaling techniques like width, depth, resolution, compound scaling are used to scale the baseline network in order to obtain a family of models. EfficientNets were able to surpass state-of-the-art accuracy on ImageNet and five other datasets with fewer parameters and FLOPS.

A residual neural network (ResNet) [44] was created to make neural networks go deeper. In CNN, each layer usually identifies only one distinct feature. So, a network with a lot of layers should be able to find more patterns than one with a few layers. However, this causes a problem known as the "vanishing gradient," where the information gets lost in the process of moving through many layers. ResNet solves this issue by making use of skip connections amongst residual blocks. This results in the ability to train larger networks without losing much information. There are different kinds of ResNet, like ResNet50, ResNet101, and ResNet152. The number at the end of the name shows how deep the model is.

We also used transfer learning technique to train our CNNs [45]. Transfer learning technique involves transferring knowledge from one task to another. The model improves much faster and converges to a better final result if transfer learning is used [45]. This method is very useful for image classification, as we often do not have a large enough dataset to train large models from scratch. But with transfer learning, we start from models pre-trained on large datasets like the ImageNet [46]. Hence, we can use these learned feature maps instead of starting from scratch every time. The pre-trained model is fine-tuned on the target dataset in the next step.

4. Experimental results

4.1. Implementation details

Table 2 illustrates the training configurations of the neural networks for our experiments. In addition to influencing the training process, these parameters significantly impact the model's performance. Optimizers determine how to update the model's weights during training, while loss functions determine how the model will be penalized when wrong predictions are made. The number of epochs determines the number of times the model will encounter the entire training dataset. The mini-batch size determines the number of training examples used in each training iteration. In cases where the model does not progress, early stopping and patience are used to determine how long the training should continue before stopping. Finally, transfer learning indicates whether the model will be initialized with pre-trained weights. We used the Adadelta, Adam, RMSprop, SGD optimizers, and categorical cross-entropy loss function in this work. The number of epochs was set to 15, and the mini-batch size was set to 32. Early stopping with patience of 5 was used during the experiments. Transfer learning was used during the experiments to improve training and performance.

Table 2
Training configuration.

Training Configuration	
Optimizer	Adadelta, Adam, RMSprop, SGD
Loss Function	Categorical cross-entropy
Transfer Learning	Yes
Mini-Batch Size	32
Number of epochs	15
Early Stopping	Yes
Patience	5

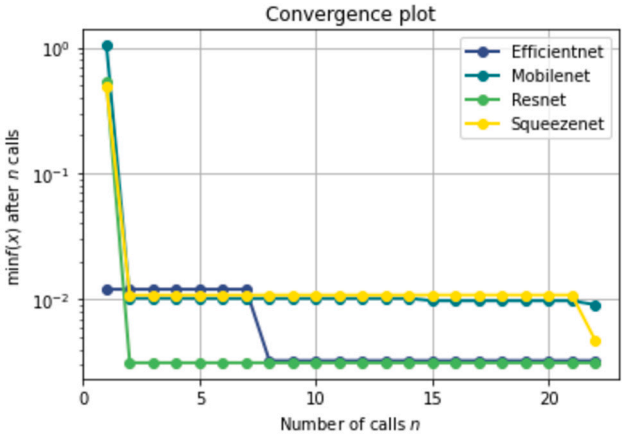
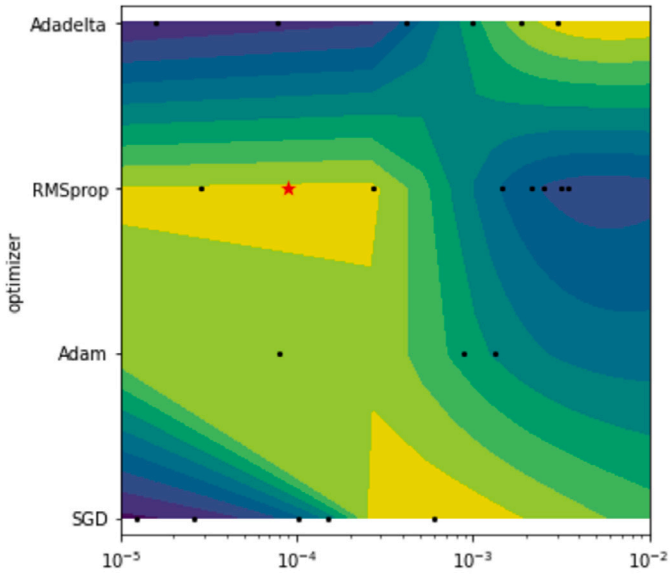


Fig. 4. Convergence analysis.

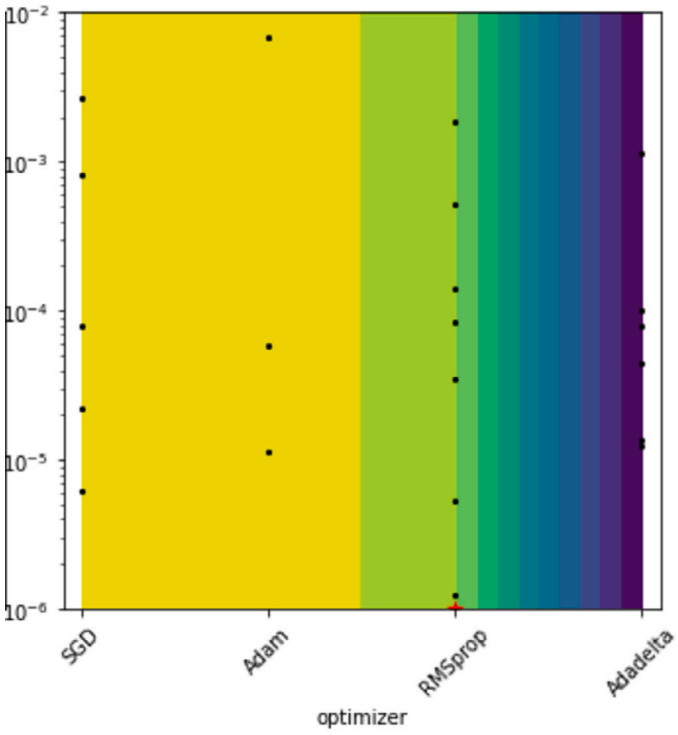
4.2. Optimization of neural network hyperparameters using Bayesian search

In this work, we tuned four neural networks using Bayesian Optimization: EfficientNet, MobileNet, ResNet, and SqueezeNet. The tuned hyperparameters were: learning rate, optimizer, and L2Regularization. Fig. 4 illustrates the convergence plot of the four optimized models. The Y-axis represents the minimum validation loss after n trials. The X-axis represents the number of neural networks trained. It is observed from Fig. 4 that MobileNet had the highest validation loss after the optimization was completed. The validation loss obtained by EfficientNet, ResNet, and SqueezeNet were quite close. However, the lowest validation loss was obtained by ResNet.

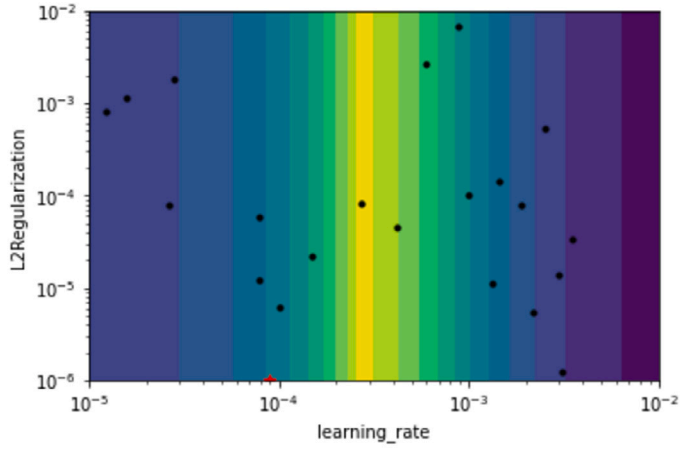
After completing the optimization, three Partial Dependence Plots (PDP) were generated. These PDPs capture the relation between the hyperparameters and the objective function. Two hyperparameters are plotted against each other, and their relationship to the objective function is shown in each plot. The value of the objective function at a specific region can be understood by color. There are mainly two colors in the plot: yellow and blue. Yellow represents the search space regions where the value of the objective function or validation loss is minimum, and blue represents the search space regions where the value of the objective function is more. PDPs for the SqueezeNet architecture are shown in Fig. 5. Hyperparameter values and objective function often follow straightforward relationships, as seen in the PDP with L2 regularization and optimizer. However, sometimes the relationships are complex, as we see in the PDP with the optimizer and learning rate. RMSprop optimizer with a learning rate lower than 10^{-4} gives the least validation error, marked as a red star in Fig. 5a. SGD performed well when the learning rate was around 10^{-3}, and Adadelta worked well when the learning rate was around 10^{-4}. As seen in the PDP with



(a) PDP with optimizer and learning rate



(b) PDP with L2 regularization and optimizer



(c) PDP with L2 regularization and learning rate

Fig. 5. Illustration of the relation between different hyperparameters and their influence on the objective function using Partial Dependence Plots.

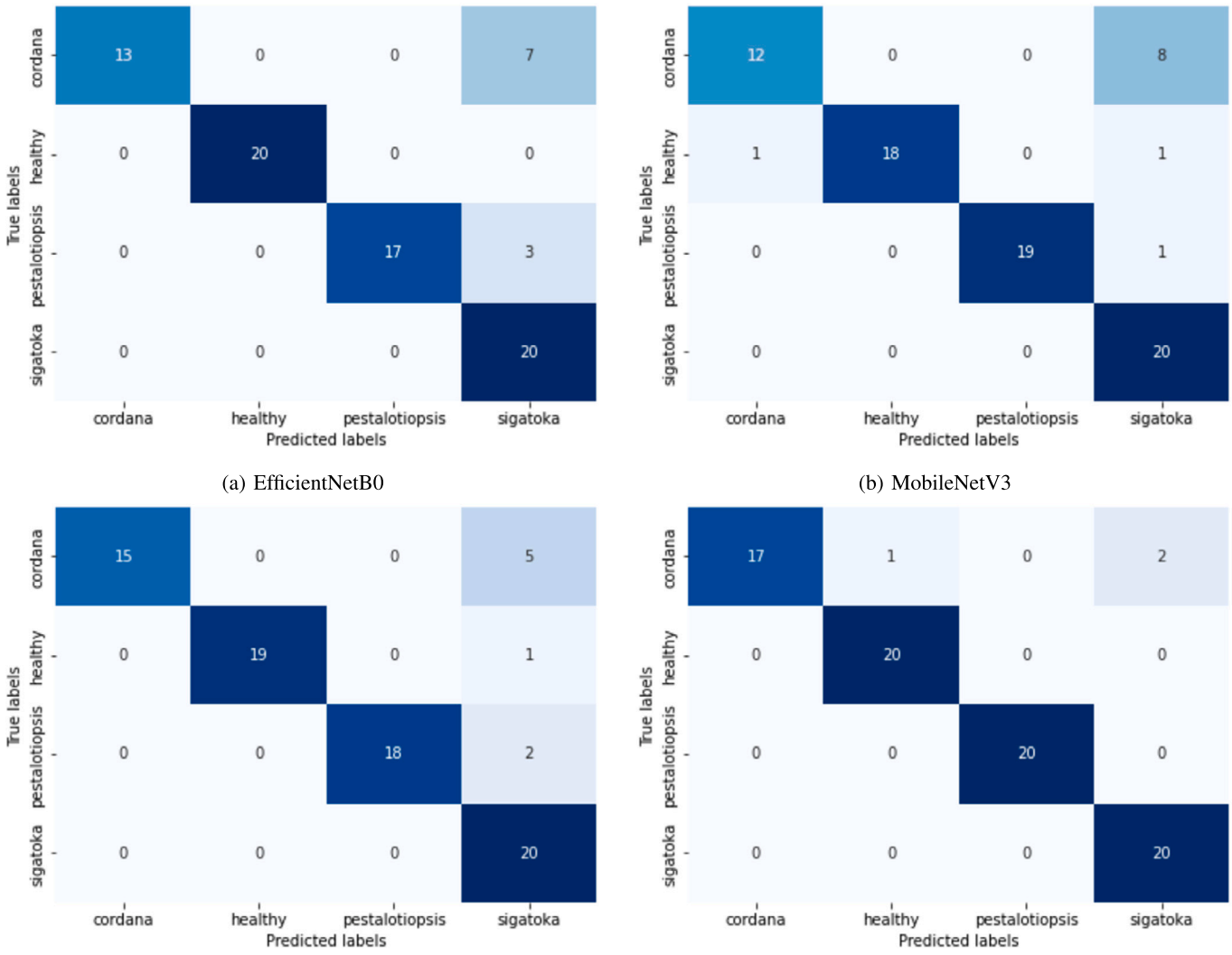
L2 regularization and the optimizer, the objective function is not significantly affected by L2 regularization. SGD and Adam performed well for any values of L2 regularization. The Adadelta optimizer performed the worst. The best value of the objective function was obtained with the RMSProp optimizer when the L2 regularization was close to 10^{-6} In the PDP with L2 regularization and learning rate, it is evident that the learning rate completely dictates the objective function, in contrast to the L2 regularization. The region where the learning rate is 10^{-3} and 10^{-4} seems the most promising. The best value of the objective function is found when the L2 regularization value is close to 10^{-6} and a learning rate value is close to 10^{-4}.

Table 3 provides a summary of the optimization process. In the table, the observed value of the objective function is displayed along with the hyperparameters for the model: learning rate, optimizer, and L2Regularization. Each row in the table represents an objective function value observed in a particular iteration and the associated hyperparameters. It is observed from the table that the objective function reached its minimum value during iteration 22 of the optimization process. The objective function value at that iteration was 0.005, and the hyperparameters used to build the model were: learning rate 8.9e-05, RMSprop optimizer, and L2Regularization 1e-06.

4.3. Evaluation of the optimized models

4.3.1. Evaluation results

Fig. 6 shows the confusion matrix for each of the models. In EfficientNetB0, the sigatoka and the healthy leaf images were classified with 100% accuracy. Three pestalotiopsis and seven cordana leaf images were misclassified as sigatoka. Only 12 images out of 20 Cordana images were properly classified by MobileNetV3. 18 healthy and 19 pestalotiopsis images were properly predicted. All 20 sigatoka images were accurately classified. This model performed the worst of all, mostly due to poor cordana classification. ResNet-101 is the second best model based on performance on the test set. Eight images out of 80



(a) EfficientNetB0 (b) MobileNetV3

(c) ResNet-101 (d) SqueezeNet

Fig. 6. Confusion matrices.

Table 3
Bayesian optimization of the SqueezeNet architecture.

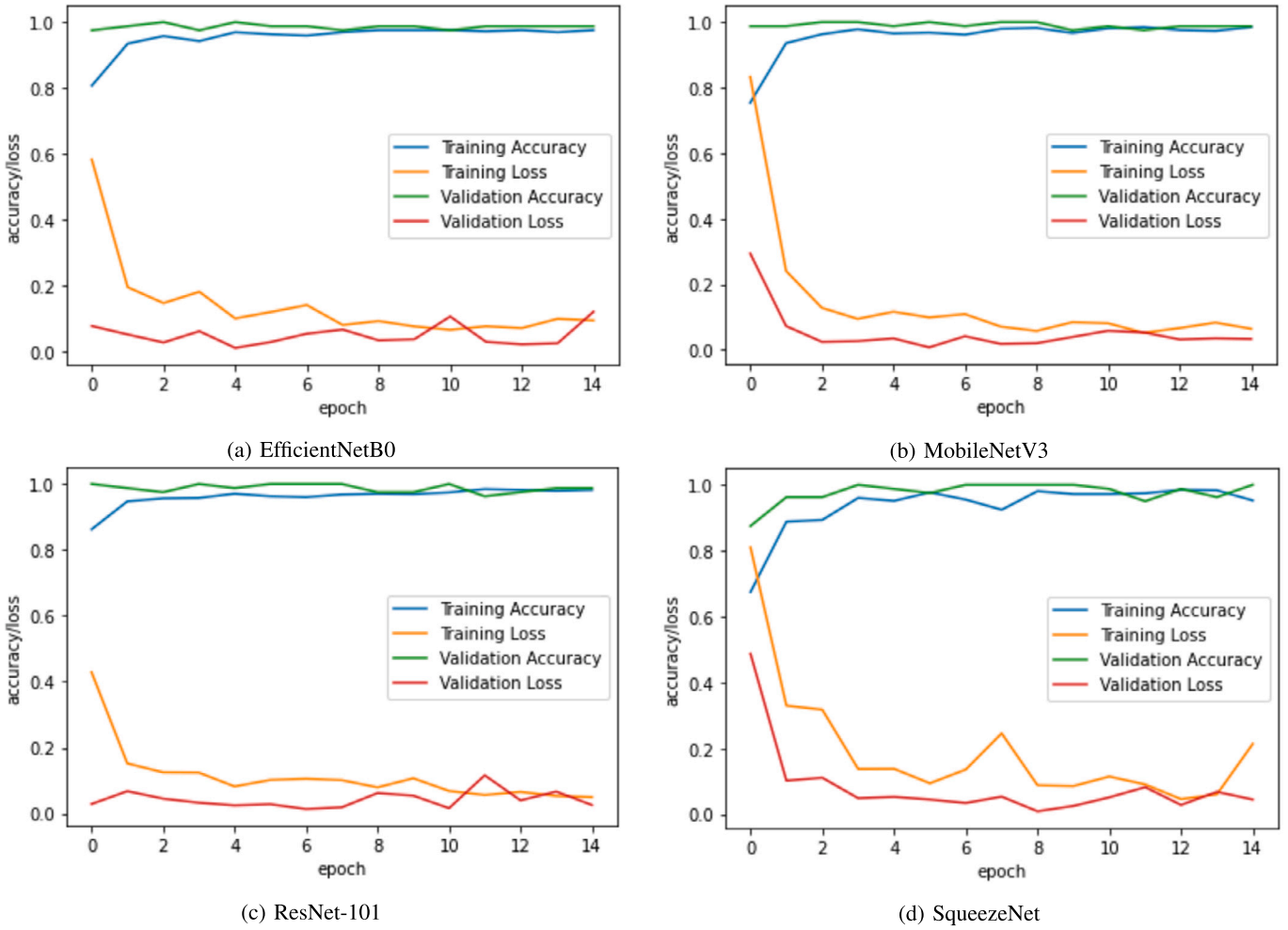
Iteration No	Objective	Learning Rate	Optimizer	L2Regularization
1	0.489	0.001	Adadelta	0.0001
2	0.011	7.8e-05	Adam	5.9e-05
3	1.182	0.003	RMSprop	1.2e-06
4	0.645	0.0013	Adam	1.1e-05
5	0.017	2.8e-05	RMSprop	0.0018
6	0.346	0.0001	SGD	6.2-06
7	1.359	1.2e-05	SGD	0.0008
8	1.002	0.0004	Adadelta	4.5e-05
9	0.902	0.0021	RMSprop	5.3e-06
10	0.097	0.0030	Adadelta	1.3e-05
11	1.426	1.5e-05	Adadelta	0.0011
12	0.031	0.0006	SGD	0.0027
13	1.036	0.0034	RMSprop	3.4e-05
14	1.569	7.8e-05	Adadelta	1.2e-05
15	0.052	0.0002	RMSprop	8.3e-05
16	0.139	0.0001	SGD	2.2e-05
17	0.482	0.0008	Adam	0.0067
18	0.916	2.6e-05	SGD	7.8e-05
19	1.102	0.0025	RMSprop	0.0005
20	1.001	0.0014	RMSprop	0.0001
21	0.179	0.0018	Adadelta	7.8e-05
22	**0.005**	**8.9e-05**	**RMSprop**	**1e-06**

Table 4
Assessment of the effectiveness of four optimized models for classification.

Model	ACC	PRE	REC	SPE	F1	MCC
EfficientNetB0	87.50	91.66	87.50	95.83	87.67	84.74
MobileNetV3	86.25	89.74	86.25	95.41	86.22	83.01
ResNet-101	90.00	92.86	90.00	96.66	90.30	87.53
SqueezeNet	**96.25**	**96.54**	**96.25**	**98.75**	**96.17**	**95.14**

were misclassified. Among these, 5 belong to the cordana class. Using SqueezeNet, all images belonging to the healthy, pestalotiopsis, and sigatoka classes were correctly predicted. 3 Cordana images were misclassified. Overall, all models predicted the sigatoka class with 100% accuracy. sigatoka also has the highest false positive rate. All models had trouble classifying the cordana leaf images.

Table 4 shows the classification performance of four optimized models: EfficientNetB0, MobileNetV3, ResNet-101, and SqueezeNet. The performance metrics are accuracy (ACC), precision (PRE), recall (REC), specificity (SPE), F1 score (F1), and Matthews Correlation Coefficient (MCC). Among the four models, SqueezeNet has the highest values for all performance metrics, with an accuracy of 96.25%, precision of 96.54%, recall of 96.25%, specificity of 98.75%, F1 score of 96.17%, and MCC of 95.14%. MobileNetV3 has the lowest values for all met-



(a) EfficientNetB0 (b) MobileNetV3

(c) ResNet-101 (d) SqueezeNet

Fig. 7. Loss and accuracy analysis of the best models from each architecture.

rics, with an accuracy of 87.5%, precision of 91.66%, recall of 87.5%, specificity of 95.83%, F1 score of 87.67%, and MCC of 84.74%.

Fig. 7 shows the change of loss and accuracy with the number of epochs for the best models from all four architectures. With increasing epochs, accuracy increases while loss decreases until they reach saturation for each model. MobileNetV3 achieves the smoothest learning curve, and the SqueezeNet model has the roughest learning curve.

The Receiver Operating Characteristic (ROC) curves for EfficientNetB0, MobileNetV3, ResNet-101 and SqueezeNet models are shown in Fig. 8. In case of EfficientNetB0 model, the area under the ROC curve for detecting Cordana, Healthy, Pestalotiopsis, and Sigatoka respectively were 91.67%, 100%, 100%, and 99.25% (8a). For MobileNetV3 model, the area under the ROC curve for detecting Cordana, Healthy, Pestalotiopsis, and Sigatoka respectively were 92.45%, 100%, 100%, and 98.25% (8b). For the ResNet-101 model, the area under the ROC curve for detecting Cordana, Healthy, Pestalotiopsis, and Sigatoka respectively was 94.75%, 100%, 100%, and 99.33% (8c). In the case of the SqueezeNet model, the area under the ROC curve for detecting Cordana, Healthy, Pestalotiopsis, and Sigatoka respectively was 89.83%, 97.83%, 99.83%, and 98.92% (8d).

4.3.2. Complexity comparison

Comparison of complexities of different architectures is shown in Table 5. While all four models require similar training time, SqueezeNet has a very small model size and number of parameters which is suitable for deployment scenarios. Due to having a relatively small number of parameters, it requires the least inference time of 17.84 s for the whole test set. MobileNetV3 and EfficientNetB0 have slightly larger inference times and they have a smaller number of Floating Point Operations (FLOPs) than SqueezeNet. On the other hand, ResNet-101 has quite a large model size and number of parameters. Due to the large number of parameters, ResNet-101 also requires more inference time than the other three models.

4.4. BananaSqueezeNet: best performing model identified using Bayesian search

The classification performance of the BananaSqueezeNet model on various classes is presented in Table 6. The table includes the model's accuracy, precision, recall, specificity, F1-score, and Matthews Correlation Coefficient (MCC) for each class. The BananaSqueezeNet performs very well, achieving an overall accuracy of 96.25% and a high F1 score of 96.17%. In particular, the model performs very well on the healthy and Pestalotiopsis classes, achieving an accuracy of 98.75% and 100%, respectively. The model also performs well on the Sigatoka class, achieving an accuracy of 97.5%. In the Cordana class, the model achieves an accuracy of 96.25%, slightly lower than the other classes. Overall, these results suggest that the BananaSqueezeNet model is highly effective at classifying banana leaf diseases.

4.5. Comparing BananaSqueezeNet model with other state-of-the-art models

We have compared the proposed BananaSqueezeNet model with four other state-of-the-art models found in the literature. These models are: ResNet-50 [44], Inception-V3 [21], and VGG-16 [47]. The

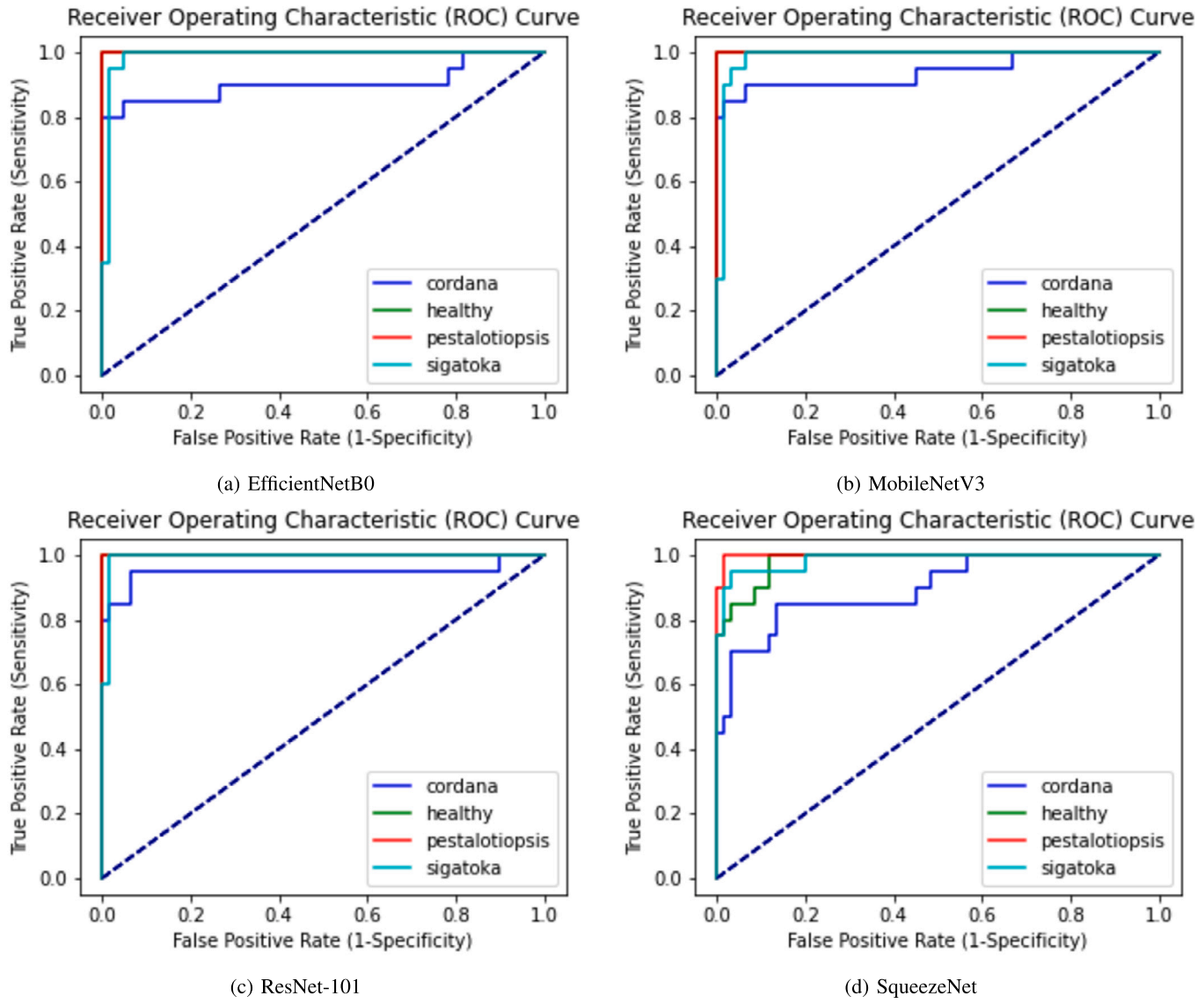


Fig. 8. Receiver operating characteristic (ROC) curve.

Table 5
Assessment of the complexity of four optimized models.

Model	Training Time	Inference Time	Model Size	Number of Parameters	MACs	FLOPS
EfficientNetB0	54 m 27 s	18.502 s	20.5 MB	4.01 M	400.39 M	0.41 G
MobileNetV3	60 m 40 s	18.38 s	16.6 MB	4.21 M	227.08 M	0.23 G
ResNet-101	58 m 3 s	23.54 s	171 MB	42.51 M	7.85 G	7.88 G
SqueezeNet	57 m 20 s	17.84 s	4.78 MB	737.48 k	743.36 M	0.75 G

Table 6
Assessment of the effectiveness of the BananaSqueezeNet model for classification.

Class	ACC	PRE	REC	SPE	F1	MCC
Healthy	98.75	95.23	100	98.33	97.56	96.77
Pestalotiopsis	100	100	100	100	100	100
Sigatoka	97.50	90.90	100	96.67	95.23	93.74
Cordana	96.25	100	85.00	100	91.89	89.90
Overall	**96.25**	**96.53**	**96.25**	**98.75**	**96.17**	**95.13**

Table 7
Assessment of the effectiveness of some larger models.

Model	ACC	PRE	REC	SPE	F1	MCC
ResNet-50	86.25	89.26	86.25	95.03	85.46	82.97
Inception-V3	90.00	91.96	90.00	96.49	89.30	87.62
VGG-16	95.00	95.45	95.00	98.27	94.84	93.57
BananaSqueezeNet	**96.25**	**96.54**	**96.25**	**98.75**	**96.17**	**95.14**

experimental results are shown in Table 7. It is observed from the table that the BananaSqueezeNet model significantly outperforms ResNet-50, and Inception-V3. The VGG-16 model performs slightly better than those two, but the BananaSqueezeNet model still outperforms it.

Table 8
Assessment of the generalizability of the BananaSqueezeNet model.

Class	ACC	PRE	REC	SPE	F1	MCC
Banana Fruit Scarring Beetle	100	100	100	100	100	100
Black Sigatoka	99.03	90.38	100	99.03	94.94	94.56
Bacterial Soft Rot	96.30	100	82.24	100	90.25	0.88.64
Pseudo Stem Weevil	96.10	93.17	100	96.10	96.46	92.43
Yellow Sigatoka	99.03	100	80.76	100	89.36	89.41
Banana Aphids	100	100	100	100	100	100
Panama Disease	99.80	100	90.00	100	94.74	94.77
Overall	**95.13**	**97.65**	**93.29**	**95.14**	**95.11**	**92.69**

4.6. Generalizability of the BananaSqueezeNet model

In order to test the generalizability of the BananaSqueezeNet model, we trained and tested the model on the PSFD-Musa dataset [25]. The dataset contains 5,170 images of seven common banana leaf, fruit, and stem diseases: banana fruit scarring beetle, black sigatoka, bacterial soft rot, pseudo stem weevil, yellow sigatoka, banana aphids, and panama disease. Table 8 presents the experimental results of the BananaSqueezeNet model on the PSFD-Musa dataset. The BananaSqueezeNet model predicted the banana fruit scarring beetle and the banana aphids disease of banana leaves with 100% accuracy. The model also predicted black sigatoka, yellow sigatoka, and panama disease with 99% accuracy. Bacterial soft rot and pseudo stem weevil had the lowest prediction accuracy, with 96.3% and 96.1%, respectively. The BananaSqueezeNet model achieved an overall accuracy of 95.13%. To the best of our knowledge, this is the highest reported accuracy on this dataset.

4.7. Qualitative analysis

The learning process of neural networks is often considered a black box, since no simple explanation exists for how they work. Class Activation Map (CAM) helps in this regard by giving us insights on what each layer looks for in an image. As a result, we are able to figure out whether our network is focusing on the region of interest, or if it seems to be focusing on the wrong regions, we are able to adjust our network configuration. In Fig. 9, we have shown how our model predicts different diseases from an image. For cordana disease, we see that our network has correctly concentrated on the central part of the leaf where the disease has actually occurred. In case of pestalotiopsis, there are multiple areas of interest in the leaf, with the disease mostly spreading at the bottom. The CAM for this image shows the larger heatmap at the bottom, and also identifies the slight variation in the middle. Similarly, we can see that our network can accurately detect the sigatoka disease pattern since the heatmap is generated correctly over the affected leaves.

4.8. App for banana growers and stakeholders

We have developed a web application to bring our model into the real world. Users can upload a photograph of a banana leaf and check if a disease is present in that leaf. The application processes the uploaded image and makes the prediction using a Flask backend. The prediction begins when a user captures an image of a banana leaf or selects an image to upload and submits it through the web application's user interface. The Flask backend then receives the image and passes it through our model, which has been trained to identify banana leaf diseases from images. Once the prediction has been made, the Flask backend returns the prediction to the user via the web application user interface. The user can then see the predicted disease and use this information to seek attention or treatment for their plants, if necessary. Fig. 10 shows that the application can accurately predict the disease in the uploaded image. Overall, the web application provides a convenient and efficient way for users to quickly receive a prediction of the disease present in an image, as the model is very fast and lightweight.

5. Discussion

In this study, we proposed a lightweight and fast CNN named BananaSqueezeNet for identification of diseases of banana leaves, fruits and stems. We initially compared the BananaSqueezeNet model with two lightweight CNN models that include EfficientNetB0 and MobileNetV3 and a comparatively larger model, ResNet-101. As most of these models are very small, they can be easily deployed in embedded devices. Hence, we can use these models to classify three leaf diseases: Sigatoka, Cordana leaf spot, and Pestalotiopsis leaf blight disease in banana leaves in real-time using a smartphone camera. The lightweight models used in our experiments only take up about 4.78 to 16.6 MB of disk space when trained, and ResNet-101 takes 171 MB. whereas heavyweight architectures like the VGG16 took about 512 MB of disk space when trained [48]. This performance shows a distinct path toward developing a smartphone-assisted disease diagnosis device for use on a global scale.

We also compared the BananaSqueezeNet model with state-of-the-art models for computer vision that includes ResNet-50, Inception-V3, VGG-16. The BananaSqueezeNet model outperforms all the other models in terms of performance. It is also the smallest of all of these models, making it more suitable for deployment in mobile devices.

We also used a deep learning technique called transfer learning. When trained on a large enough dataset, the earlier layers of a deep convolutional neural network detect more straightforward features like edges, shapes, and simple patterns. These features are standard in almost any computer vision task, and thus, this knowledge can be reused in other datasets. Instead of training from scratch, we started with models pre-trained on the ImageNet dataset and then fine-tuned them for our dataset. We used the Bayesian optimization technique to find the best possible hyperparameters for our models.

Of the four models that were trained and optimized, BananaSqueezeNet achieved the highest accuracy of 96.25% despite having the smallest model size. The other three models ResNet, EfficientNetNet and MobileNet, achieved 90%, 87.5% and 86.25% accuracy, respectively. The SqueezeNet model is the smallest, with a size of only 4.78 MB and ResNet is of 171 MB.

We also tested the generalizability of our work by evaluating its performance on a larger dataset of banana leaf, fruit, and stem diseases, titled PSFD-MUSA [25]. The BananaSqueezeNet model performed remarkably on this dataset, confirming its generalizability.

However, a few limitations are still present in this paper. The dataset that we used was quite small. It would also have been better to include images of banana leaf of different countries, which would have increased the diversity of the data. Overall, the present approaches are reasonably well to diagnose leaf blight of banana (Pestalotiopsis microspora), a newly reported disease of banana in Bangladesh [24], and separate this disease from Sigatoka, and Cordana leaf spots; and is expected to improve in the future with more training data.

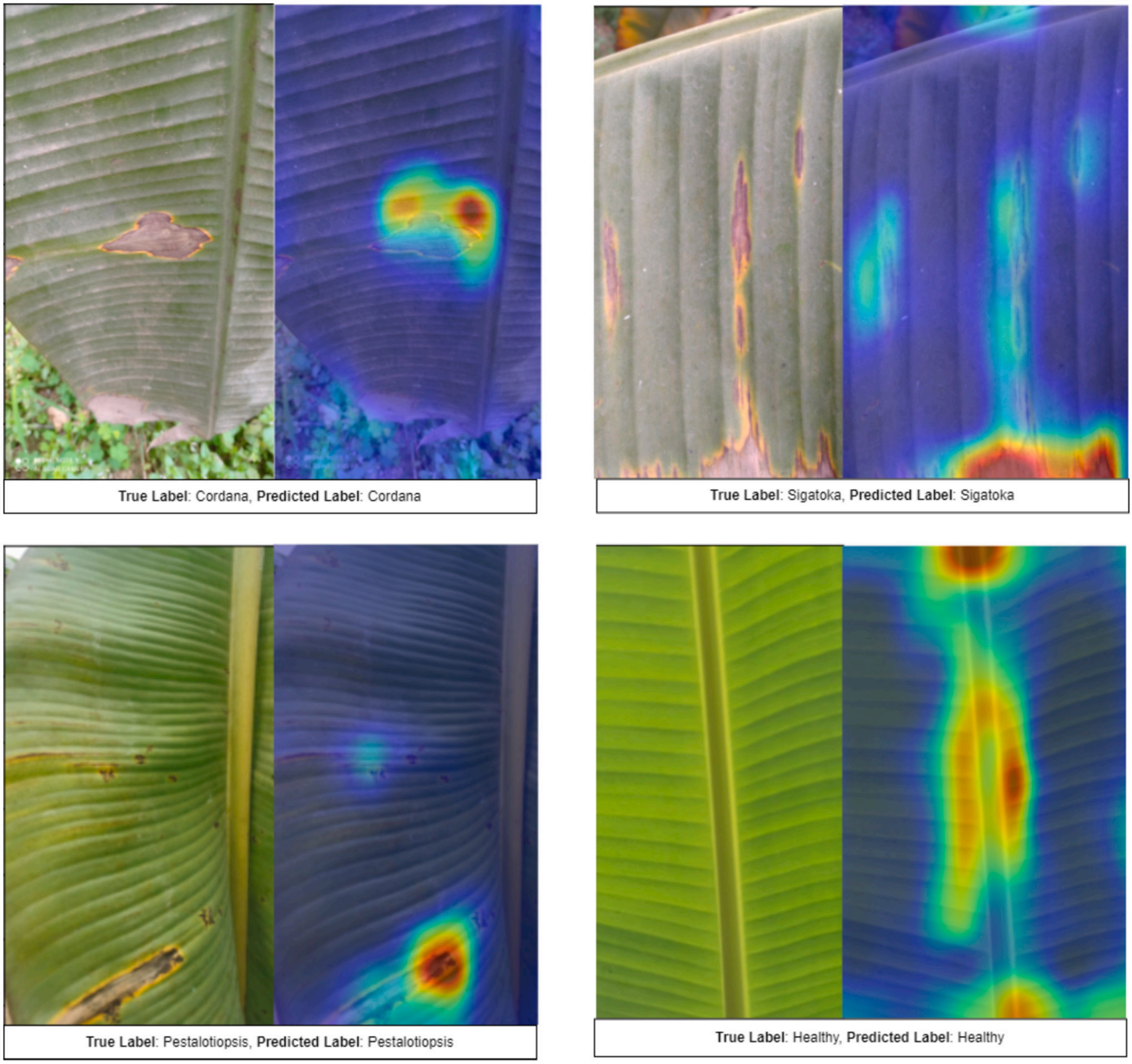


Fig. 9. Class activation map generated using the BananaSqueezeNet architecture for different banana leaf samples.

Our work aims to create a convenient and straightforward approach of diagnosing banana leaf diseases. Farmers or other stakeholders can take pictures of banana leaves from the field using the app and identify the disease within seconds. Selvaraj et al. [26] used a different approach where they detected banana plant and their major diseases from aerial images using different machine learning methods. The authors in [27] discuss a deep transfer learning based method to detect banana pests and disease symptoms from banana field images. They detected five major banana diseases. Similarly in [28] the authors proposed methods for diagnosing only two banana leaf diseases: black and yellow sigatoka. In our work, we showed how we can detect 9 banana leaf, fruit, and stem diseases that include pestalotiopsis, sigatoka (black sigatoka, yellow sigatoka), cordana, banana fruit scarring bettle, bacterial soft rot, pseudo stem weevil, banana aphids, and panama disease.

Finally, our goal is not to replace the laboratory technique of pathogen identification or to replace any established diagnosis approaches, but rather to supplement them. Laboratory techniques are always more reliable for diagnosing the plant disease than visual inspection systems. Even at the early stage of a disease, diagnosis is very challenging through visual observation. Therefore, alternate approaches such as CNNs would provide additional support to diagnose plant disease using smartphones. Keeping the current global 3.5 billion smartphone users [49] in our mind and a rapid rise of smartphone usage, we strongly believe that our lightweight models could be an additional support to prevent yield loss. Our effort to distinguish leaf diseases of bananas could be a great example of plant disease diagnosis in real-time using a smartphone camera.

6. Conclusion

In this paper, we demonstrated how different diseases that affect banana leaf, fruit, and stem can be detected from images. We collected a dataset of banana leaf images from the BSMRAU experimental field and different farmers' fields in Bangladesh. We proposed a lightweight CNN architecture named BananaSqueezeNet which can quickly identify leaf diseases from banana leaf images with high accuracy. Since the BananaSqueezeNet model is very lightweight, it can be easily deployed on mobile devices. We hope that the BananaSqueezeNet model

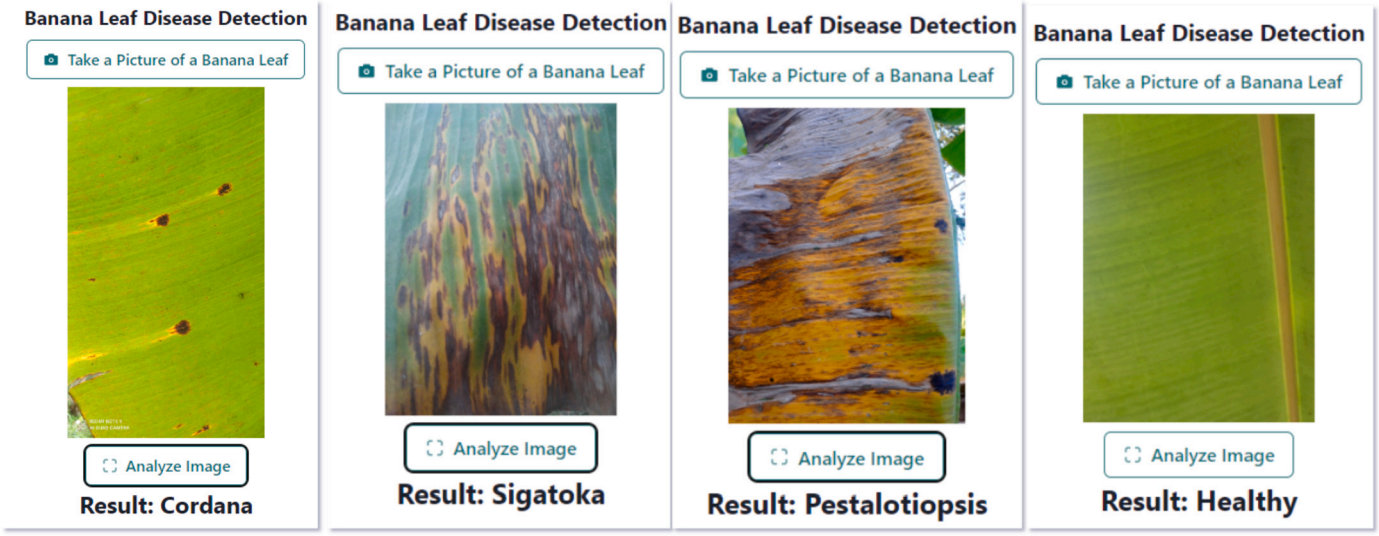


Fig. 10. Screenshots taken from the web-app classifying banana diseases.

will be used by farmers and stakeholders worldwide for early diagnosis of banana leaf diseases and take precautionary steps. Future research will investigate the efficacy of the BananaSqueezeNet architecture for identifying diseases of other plants' leaves. Farmers should also be instructed on what measures to take based on the severity of the disease in future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] B. Singh, J.P. Singh, A. Kaur, N. Singh, Bioactive compounds in banana and their associated health benefits – a review, Food Chem. 206 (Sep. 2016) 1–11.
[2] D. Mohapatra, S. Mishra, N. Sutar, Banana and its by-product utilisation: an overview, J. Sci. Ind. Res. 69 (05 2010) 323–329.
[3] W.H. Organization, et al., Fruit and Vegetables for Health: Report of the Joint Fao, 2005.
[4] Faostat, https://www.fao.org/faostat/en/#data, 2019.
[5] P. Crous, X. Mourichon, Mycosphaerella eumusae and its anamorph pseudocercospora eumusae spp. nov.: causal agent of eumusae leaf spot disease of banana, South Afr. J. Sci. 54 (06 2002).
[6] A. Surridge, A. Viljoen, P. Crous, F. Wehner, Identification of the pathogen associated with sigatoka disease of banana in South Africa, Australas. Plant Pathol. 32 (03 2003) 27–31.
[7] S.-H. Lin, S.-L. Huang, Q.-Q. Li, C.-J. Hu, G. Fu, L.-P. Qin, Y.-F. Ma, L. Xie, Z.-L. Cen, W.-H. Yan, Characterization of exserohilum rostratum, a new causal agent of banana leaf spot disease in China, Australas. Plant Pathol. 40 (3) (Mar. 2011) 246–259.
[8] M.H. Restrepo, J.Z. Groenewald, P.W. Crous, Neocordana gen. nov., the causal organism of cordana leaf spot on banana, Phytotaxa 205 (4) (Apr. 2015) 229.
[9] S. Huang, B. Yan, J. Wei, W. Yan, Z. Cen, T. Yang, First report of plantain zonate leaf spot caused by pestalotiopsis menezesiana in China, Australas. Plant Dis. Notes 2 (1) (2007) 61–62.
[10] M.-H. Wong, P.W. Crous, J. Henderson, J.Z. Groenewald, A. Drenth, Phyllosticta species associated with freckle disease of banana, Fungal Divers. 56 (1) (2012) 173–187.
[11] S. Han, Y. Wang, M. Wang, S. Li, R. Ruan, T. Qiao, T. Zhu, First report of pestalotiopsis microspora causing leaf blight disease of machilus nanmu in China, Plant Dis. 103 (11) (2019) 2963.
[12] G. Wang, Y. Sun, J. Wang, Automatic image-based plant disease severity estimation using deep learning, Comput. Intell. Neurosci. 2017 (2017).
[13] N. Yesmin, F. Jenny, H.M. Abdullah, M.M. Hossain, M.A. Kader, P.S. Solomon, M.A. Bhuiyan, A review on South Asian wheat blast: the present status and future perspective, Plant Pathol. 69 (9) (2020) 1618–1629.
[14] S. Sanga, V. Mero, D. Machuve, D. Mwanganda, Mobile-based deep learning models for banana diseases detection, arXiv preprint, arXiv:2004.03718, 2020.
[15] J. Amara, B. Bouaziz, A. Algergawy, et al., A deep learning-based approach for banana leaf diseases classification, in: BTW (Workshops), vol. 266, 2017, pp. 79–88.
[16] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, J. Legg, D.P. Hughes, A mobile-based deep learning model for cassava disease diagnosis, Front. Plant Sci. (2019) 272.
[17] J. Kamdar, M. Jasani, J. Jasani, J. Praba, J.J. Georrge, 11 artificial intelligence for plant disease detection: past, present, and future, in: Internet of Things and Machine Learning in Agriculture, De Gruyter, 2021, pp. 223–238.
[18] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[19] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, Ieee, 2005, pp. 886–893.
[20] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.
[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[22] S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016) 1419.
[23] D.L. Hernández-Rabadán, F. Ramos-Quintana, J. Guerrero Juk, Integrating soms and a bayesian classifier for segmenting diseased plants in uncontrolled environments, Sci. World J. 2014 (2014).
[24] M.A.B. Bhuiyan, S.M.N. Islam, M.A.I. Bukhari, M.A. Kader, M.Z.H. Chowdhury, M.Z. Alam, H.M. Abdullah, F. Jenny, First report of pestalotiopsis microspora causing leaf blight of banana in Bangladesh, Plant Dis. 106 (5) (2022) 1518.
[25] E. Medhi, N. Deb, Psfd-musa: a dataset of banana plant, stem, fruit, leaf, and disease, Data Brief 43 (2022) 108427.
[26] M.G. Selvaraj, A. Vergara, F. Montenegro, H.A. Ruiz, N. Safari, D. Raymaekers, W. Ocimati, J. Ntamwira, L. Tits, A.B. Omondi, et al., Detection of banana plants and their major diseases through aerial images and machine learning methods: a case study in dr Congo and republic of Benin, ISPRS J. Photogramm. Remote Sens. 169 (2020) 110–124.
[27] M.G. Selvaraj, A. Vergara, H. Ruiz, N. Safari, S. Elayabalan, W. Ocimati, G. Blomme, Ai-powered banana diseases and pest detection, Plant Methods 15 (2019) 1–11.
[28] J. Sujithra, M. Ferni Ukrit, Performance analysis of d-neural networks for leaf disease classification-banana and sugarcane, Int. J. Syst. Assur. Eng. Manag. (2022) 1–9.
[29] K. Seetharaman, T. Mahendran, Leaf disease detection in banana plant using Gabor extraction and region-based convolution neural network (rcnn), J. Inst. Eng. (India), Ser. A 103 (2) (2022) 501–507.
[30] S. Gopinath, K. Sakthivel, S. Lalitha, A plant disease image using convolutional recurrent neural network procedure intended for big data plant classification, J. Intell. Fuzzy Syst. 43 (4) (2022) 4173–4186.
[31] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, Adv. Neural Inf. Process. Syst. 25 (2012).

[32] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: a novel bandit-based approach to hyperparameter optimization, J. Mach. Learn. Res. 18 (1) (2017) 6765–6816.
[33] Y. Ozaki, Y. Tanigaki, S. Watanabe, M. Nomura, M. Onishi, Multiobjective tree-structured parzen estimator, J. Artif. Intell. Res. 73 (2022) 1209–1250.
[34] S.E. Arman, S.A. Deowan, Igwo-ss: improved grey wolf optimization based on synaptic saliency for fast neural architecture search in computer vision, IEEE Access 10 (2022) 67851–67869.
[35] S.E. Arman, S. Rahman, S.A. Deowan, Covidxception-net: a bayesian optimization-based deep learning approach to diagnose Covid-19 from x-ray images, SN Comput. Sci. 3 (2) (2022) 115.
[36] P. Doke, D. Shrivastava, C. Pan, Q. Zhou, Y.-D. Zhang, Using cnn with bayesian optimization to identify cerebral micro-bleeds, Mach. Vis. Appl. 31 (2020) 1–14.
[37] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, Squeezenet: alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size, arXiv preprint, arXiv:1602.07360, 2016.
[38] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint, arXiv:1704.04861, 2017.
[39] M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, Jun 2019, pp. 6105–6114.
[40] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[41] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, S.J. Oh, Rethinking spatial dimensions of vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11936–11945.
[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[43] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[45] E.S. Olivas, J.D.M. Guerrero, M. Martinez-Sober, J.R. Magdalena-Benedito, L. Serrano, et al., Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques, IGI Global, 2009.
[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014.
[48] L. Fu, Y. Feng, J. Wu, Z. Liu, F. Gao, Y. Majeed, A. Al-Mallahi, Q. Zhang, R. Li, Y. Cui, Fast and accurate detection of kiwifruit in orchard using improved yolov3-tiny model, Precis. Agric. 22 (3) (2021) 754–776.
[49] R. Keçili, F. Ghorbani-Bidkorbeh, A. Altıntaş, C.M. Hussain, Future of smartphone-based analysis, in: Smartphone-Based Detection Devices, Elsevier, 2021, pp. 417–430.