Received: October 2, 2024. Revised: November 1, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhanced Detection of Bean Leaf Diseases Using a Stacked CNN Ensemble with Transfer Learning

Naglaa E. Ghannam[1,2]* Ola M. El Zein[2] Doaa R. Fathy[2] H. Mancy[2, 3]

[1]*Department of Computer and Information, College of Engineering, in Wadi Alddawasir, Prince Sattam Bin Abdulaziz University, Wadi Alddawasir, Saudi Arabia*
[2]*Faculty of Science, Al-Azhar University (Girls' branch), Cairo, Egypt*
[3]*Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-kharj, Saudi Arabia*
* Corresponding author's Email: n.said@psau.edu.sa

Abstract: Bean leaf diseases are the major risk aspect for plant growth, and early detection is critical for farmers but challenging due to the complex structure of bean leaf diseases. Bean leaf diseases such as bean rust and angular leaf spots significantly diminish the quality and yield of agricultural products. Accurate detection is crucial for enhancing crop Yield and quality. To tackle this challenge, this paper proposes a novel approach using a deep stacked ensemble learning model, which combines the predictions of several pre-trained Convolutional Neural Network (CNN) models based on the Transfer Learning (TL) technique and utilizes a meta-learner on averaged predictions to detect bean leaf diseases. We have trained three pre-trained CNN models—EfficientNetB3, InceptionV3, and MobileNetV2—on a bean leaf dataset with 1296 leaf images and assessed their efficiency. Finally, we utilized a stacked ensemble learning approach, where the average of the predictions from these models are used as features to train an ensemble model to enhance the detection accuracy of bean leaf diseases. The proposed stacked ensemble method, particularly the combination of EfficientNetB3 and InceptionV3, achieves exceptional results with a classification accuracy of 99.22%, precision of 99.24%, recall of 99.22%, and F1-score of 99.22% on the test data with reduced training time, outperforming other state-of-the-art models.

Keywords: Beans leaf, Transfer learning, Convolutional neural network, Fine-tuning, Deep stacked ensemble learning.

1. Introduction

Agriculture plays a prominent role in the development of any country's economy. Diseases in plant leaves present a notable risk to crop health, impacting both their productivity and overall quality [1]. Plant diseases reduce agricultural production quantity and quality by 20-40%, with leaf diseases alone accounting for 42% of losses in this field [2].

Beans are a valuable crop worldwide, providing millions of people with a staple food source and income. Renowned for their rich protein content and associated health benefits, beans play a key role in promoting nutritional well-being, and 30% of this crop is produced by small-scale farmers in Latin America and Africa [3, 4]. This underscores the significant contribution of beans in supporting the livelihoods and nutritional needs of communities in these regions. On the other hand, bean leaves are susceptible to a variety of diseases, some caused by fungi and others by bacteria [5].

Bean production suffers heavily from diseases like angular leaf spot and bean rust. These pathogens disrupt healthy growth and reduce yields. Farmers often turn to various types of pesticides to combat these diseases. For example, fungicides, biological control methods, and cultural practices like intercropping can manage both angular leaf spot and bean rust [6, 7]. However, overreliance on chemical pesticides poses serious risks to human health and the environment. To minimize reliance on harmful chemicals, save costs, and protect the environment, early detection of plant leaf diseases is crucial.

Unfortunately, identifying disease with the naked eye can be challenging for humans [3]. Despite existing disease control methods, accurately identifying and classifying plant leaf diseases remains crucial to minimizing yield losses in agriculture. This is where automated identification, particularly through readily available smartphone technology, shines. Affordable smartphones are empowering farmers to capture images of diseased leaves and, with the help of dedicated apps, receive instant diagnoses. This early detection enables targeted interventions, minimizing reliance on harsh chemicals and maximizing crop yields [8].

As research progresses, smartphone-based disease identification has the potential to revolutionize agriculture, promoting sustainability and food security [9]. Accurate detection and classification of bean leaf diseases is essential for early-stage problem resolution. This work aims to address farmers' challenges in identifying bean leaf diseases, providing a solution for early-stage disease intervention. By facilitating the early detection and treatment of bean leaf diseases, the initiative strives to enhance both the quality and quantity of crops. Ultimately, this contributes to boosting farmers' profits by promoting healthier crops and more efficient agricultural practices.

This paper proposes a deep stacked ensemble learning model that combines the predictions of several pre-trained CNN models based on the TL technique for detecting bean leaf images. We have trained three pre-trained models namely, EfficientNetB3, InceptionV3, and MobileNetV2 on the bean leaf dataset and assessed their efficiency, and then finally utilized stacked ensemble learning which is based upon utilizing the averages of the component models. The final predictions from these base models are utilized as features to train an ensemble model.

The main contributions of this study are the following tasks:

1. Present a deep stacked ensemble learning method for the multiclass classification of bean leaf diseases which increases the reliability of automated diagnoses of bean leaf diseases by providing robustness against dataset noise and unpredictability also it improves overall prediction accuracy, decreasing the possibility of incorrect classifications.
2. The proposed method allows for a more thorough exploration of the feature space, allowing for the detection of subtle patterns that individual models may find difficult to discover.
3. TL and fine-tuning are being utilized to extract meaningful and informative features from images of bean leaves. Instead of training a deep learning model from scratch, the pre-trained models EfficientNetB3, InceptionV3, and MobileNetV2 were utilized as a starting point to reduce the training time.
4. The efficiency of the pre-trained CNN models (EfficientNetB3, Inception V3, and MobileNetV2) is assessed on a bean leaf dataset, providing insights into their performance and suitability for the task.

The rest of this paper is organized as follows: Related work approaches utilized for diagnosing crop leaf diseases are presented in Section 2. The proposed stacked ensemble learning model architecture based on EfficientNetB3, InceptionV3, and MobileNetV2 is proposed in Section 3. Section 4 shows the experimental results with a comparative analysis of state-of-the-art models. Finally, conclusions based on the study findings and outlined probable areas for future research are introduced in section 5.

2. Related works

In recent times, researchers have put forth various approaches for identifying bean leaf diseases through the utilization of machine learning. These approaches aim to develop automated solutions that can help farmers identify infected leaves early on, preventing significant damage to crops. The following is an overview of the latest published research on the classification of diseases affecting bean leaves.

Muthukannan et al. [10] proposed a framework for classifying crops based on their disease using neural network methods such as Feed Forward Neural Network (FFNN), Learning Vector Quantization (LVQ) and Radial Basis Function Networks (RBF). The overall classification accuracy for FFNN, LVQ, RPF are 90.67%, 56.77% and 71.18% respectively. Their experiments showed that the FFNN method achieved the best accuracy, reaching about 90.67%. Nonetheless, the dataset utilized was insufficiently large to adequately assess the proposed method, comprising only 118 samples of plant leaf images.

Kawasaki et al. [11] proposed a new architecture depending on Convolutional Neural Networks (CNNs) to identify and detect the disease present in the leaves of cucumber crop. The model achieved an accuracy of 94.9% on cucumber leaf dataset with total 800 leaf images, effectively distinguishing between zucchini yellow mosaic virus, melon yellow spot virus, and the non-diseased category. Training CNNs from scratch is utilized in this research can

classify diseases efficiently, but takes more processing time. Hence main weakness of this research is early detection of infection is not possible.

Devaraj et al. [12] introduced an innovative automated system based on computer vision for classifying and early detection of diseases in bean crops. The methodology involves the application of histogram equalization for image enhancement. The enhanced images undergo segmentation through a hybrid clustering approach, combining k-means and the watershed algorithm. Features are then extracted from the segmented areas using Principal Component Analysis (PCA), and the final classification is achieved through Support Vector Machine (SVM) classification. SVM demonstrates robust performance in disease classification. The precision, recall, error rate, and average accuracy achieved were 73.6%, 81.2%, 15%, and 84%, respectively. However, the bean leaf dataset used was too small to properly evaluate the proposed method, as it only contained 400 samples.

Sahu et al. [13] introduced a study comparing the performance of pre-trained models and training from scratch in the agricultural domain, specifically focusing on the training of CNNs. The paper shows the principles behind training a CNN from the beginning and utilizing pre-trained models. The experiment involved bean crop leaf images with total 1296 images, encompassing both infected and healthy samples. A significant improvement in accuracy was observed, with the validation accuracy increasing from approximately 70% to 97.06%. The test accuracy achieved an impressive 96.06%. The challenge of this method utilized leading deep neural networks, which are usually quite slow.

Sahu et al. [14] proposed deep learning models for the classification of diseases affecting bean crops. The study specifically compared CNN based deep learning models for the classification of two common bean leaf diseases, namely angular leaf spot and bean rust. The results of the experiments indicated that GoogleNet outperformed VGG16 in the task of disease classification for bean leaf crops dataset with 1296 leaf imagess with an accuracy of 95.31%. The challenge of this method utilized leading deep neural networks, which are usually quite slow.

Abed et al. [15] presented a real-time framework for assessing the health status of bean leaves by using Deep Neural Networks (DNNs). The U-Net architecture was applied in the study to identify and locate bean leaves within input images. The study evaluates the performance of five distinct deep-learning models: VGG-19, VGG-16, Densenet121, ResNet34, and ResNet50 on the bean leaf dataset with total 1296 images.

Table 1. A literature overview of several proposed methods, including references, crop types, datasets, architectures, and accuracy

Reference Author year	Datasets	Architecture	Accuracy
Muthukannan et al. [10] (2015)	plant leaf images 118	FFNN, LVQ, RBF	90.67%, 56.77%, 71.18%
Kawasaki et al. [11] (2015)	cucumber leaf images 800	CNN	94.9%
Devaraj et al. [12] 2017	beans leaf image 400	hybrid clustering approach (k-means and the watershed), PCA, SVM	84%
Sahu et al. [13] (2020)	beans leaf image 1296	CNNs, Pre-trained networks	56.08%, 96.06%
Sahu et al. [14] (2021)	beans leaf image 1296	GoogleNet,VGG16	95.31%, 93.75%
Abed et al. [15] (2021)	beans leaf image 1296	Densenet121, ResNet34, ResNet50, VGG-16, VGG-19	91.01% using Densenet 121
Elfatimi et al. [16] (2022)	beans leaf image 1296	MobileNets, optimization methods	92.97%
Önler [17] (2023)	beans leaf image 1296	Hybrid model (HOG, transfer learning)	99.24%
Singh et al. [18] (2023)	beans leaf image 1295	MobileNetV2 , EfficientNetB 6, NasNet	91.74%
Suma et al. [19] (2023)	beans leaf image 990	AlexNet	96.8%
Elfatimi et al. [20] (2024)	beans leaf image 1296, 1231, 2527	MobileNets, GradCAM	92.97% 94.53% 93.75%

The results reveal an impressive Classification Accuracy Rate (CAR) of 91.01% using the Densenet121 model for multi-classification tasks.

One drawback of this technique was the extensive execution time due to its numerous parameters.

Elfatimi et al. [16] presented a new method to identify and classify bean leaf diseases into their classes by utilizing MobileNet model and bean leaf images with total 1296 images. This work was based on MobileNet and based on an accurate comparison and evaluation of MobileNet architectures (hyperparameters and optimization methods) that define smaller and more efficient MobileNets models, the classification average accuracy of this work is more than 97% on the training dataset and more than 92% on test data for two unhealthy classes and one healthy class.

Önler in [17] introduced an Artificial Neural Network (ANN) model for the detection of bean leaf diseases. This network was developed by integrating descriptive vectors extracted from bean leaves, utilizing both transfer learning feature extraction and histogram-oriented gradient (HOG) feature extraction methods. The dataset employed in the study comprised images of bean leaf crops with total 1296 images representing classes related to bean rust, angular leaf spot, and healthy leaves. Remarkably, the model demonstrated impressive performance, accuracy rates of 98.33%, 98.40%, and 99.24% in the training, validation, and test datasets, respectively.

In their work [18], Singh et al. proposed the utilization of three pre-trained deep learning models, namely MobileNetV2, EfficientNetB6, and NasNet, for transfer learning on the Beans Leaf image dataset with total 1296. The study also incorporated various optimization techniques to assess the performance variations among different CNN models. The EfficientNetB6 achieving an accuracy rate of 91.74% outperformed the other models. However, it needs different preprocessing techniques to improve the classification accuracy.

Suma et al. [19] introduced AlexNet model to detect the beans leaf disease and classify the beans leaf image with 990 images into infected or healthy. The AlexNet model achieved 96.8% accuracy on the test dataset and 99.7% on the training dataset. Limitations of the model include the absence of explanation, over-optimism in its results, and challenges in generalization.

Elfatimi et al. [20] developed a classification system for bean leaf diseases by utilizing MobileNet models. The effectiveness of the approach is evaluated by testing the model on three different bean leaf image datasets with varying difficulty. The GradCAM technique is applied to the model's prediction to enhance the interpretability of a MobileNet CNN model in its classification of bean leaf images. The proposed approach achieved remarkable accuracy, with over 92% accuracy on all three datasets of bean leaf images.

Much research has been carried out in recent years to develop automated methods of bean leaf disease detection. However, many of these methods use deep neural networks, which are usually quite slow, and most of these methods suffer from limited data size, which affects the model's performance. Also, no study focuses on a stacked ensemble learning model for the multiclass classification of bean leaf diseases. In this work, we aim to improve the performance of the bean leaf disease detection model over state-of-the-art models by using a deep stacked ensemble learning model, which combines the predictions of several pre-trained CNN models based on the TL technique and utilizes a meta-learner on averaged predictions. The main difference between our proposed and state-of-the-art models is that we introduce a stacked ensemble learning model based on TL for the multiclass classification of bean leaf diseases. This model increases the reliability of automated diagnoses by providing robustness against dataset noise and unpredictability. It also improves overall prediction accuracy, decreasing the possibility of incorrect classifications and also, reducing the training time, and overcomes the issue of limited data size by using the TL technique instead of training CNN from scratch.

3. Proposed method

This paper proposes a stacked ensemble learning model that combines the predictions of several pre-trained CNN models based on the TL technique and utilizing a meta-learner on averaged predictions for detecting bean leaf images. Fig. 1 illustrates the detection proposed method employed in this study for bean leaf diseases. This method integrates the advantages of stacked ensemble learning with DCNNs based on TL to enhance the accuracy and reliability of image classification. The stacked ensemble model combines three individual classification models (EfficientNetB3, InceptionV3, and MobileNetV2) to create a more robust overall model. The integration strategy involves several steps: first, we apply data augmentation to the original bean leaf images after preprocessing for the training and validation datasets. Then, each of the three CNN models is independently trained on this augmented data. Finally, their individual predictions are then merged through an average ensemble technique. The final predictions from these base models are utilized as features to train an ensemble model. This ensemble model determines the most effective way to combine the predictions from the

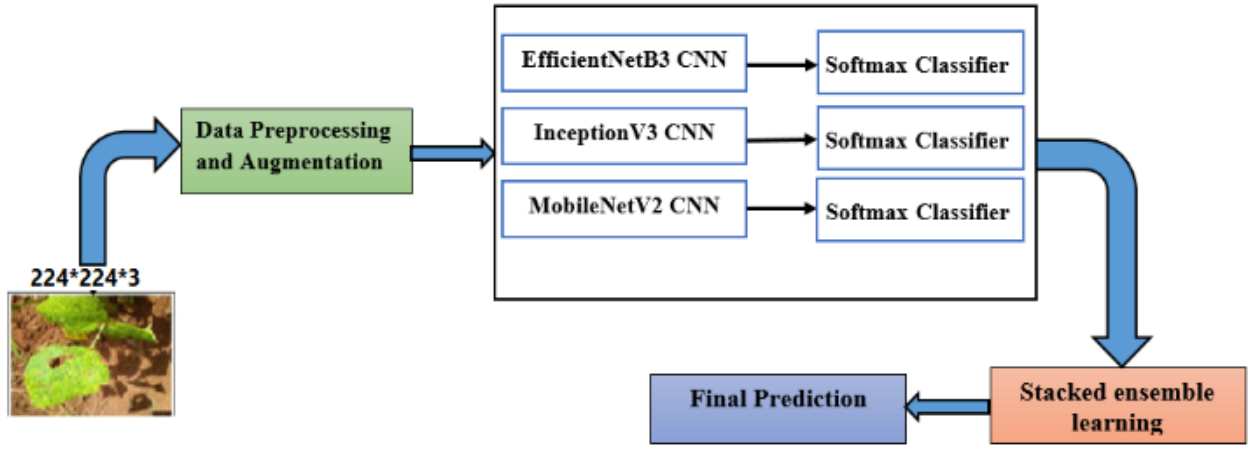
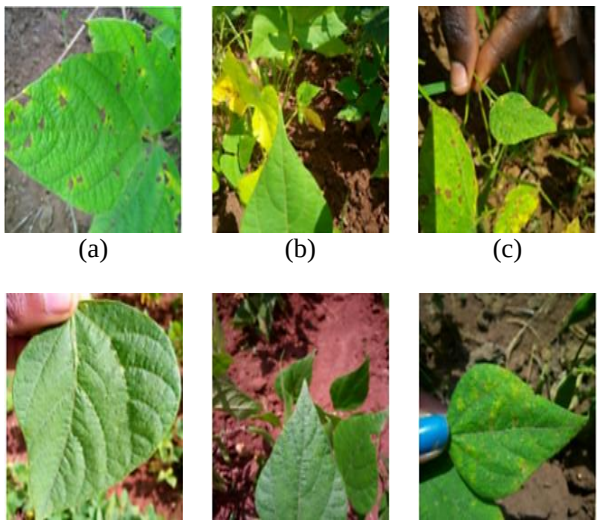


Figure. 1 Proposed model architecture of bean leaf detection

base models, resulting in a final, more accurate prediction. Each step of the proposed architecture is explained in detail in the following subsections.

3.1 Dataset collection and preprocessing

The bean leaf dataset consists of images captured with smartphone cameras, featuring three classes: angular leaf spots, bean rust, and healthy, as shown in Fig. 2. The dataset was gathered by the Makerere AI research lab and annotated by experts at the National Crops Resources Research Institute (NaCRRI) [21]. The images are 500 × 500 pixels in RGB format. The dataset is divided into training, test, and validation subsets for training, validating, and testing the machine learning model. Specifically, the training dataset includes 1,035 images, the test dataset contains 128 images, and the validation dataset has 133 images. Table 2 details the number of images in each class: healthy, bean rust, and angular leaf spot. The model was trained and validated using the training and validation datasets, while the test dataset, which was not used during training, was employed to evaluate the model's performance. In this study, the leaf image is resized to 224 × 224 × 3, which is then used to evaluate the performance of pre-trained models such as EfficientNetB3, InceptionV3, and MobileNetV2. Smaller images require fewer computations during both training and inference, leading to faster processing times. Fig. 2 depicts several bean leaf image samples from the dataset. In this study, the pixel values of input images, post-resize, are normalized within the range of 0 to 1. , each pixel value is multiplied by 1/255 for normalization.This normalization procedure ensures that the CNN model can effectively learn and optimize its parameters during the training process, promoting stability and efficiency in the gradient descent.



(a) (b) (c) (d) (e) (f)

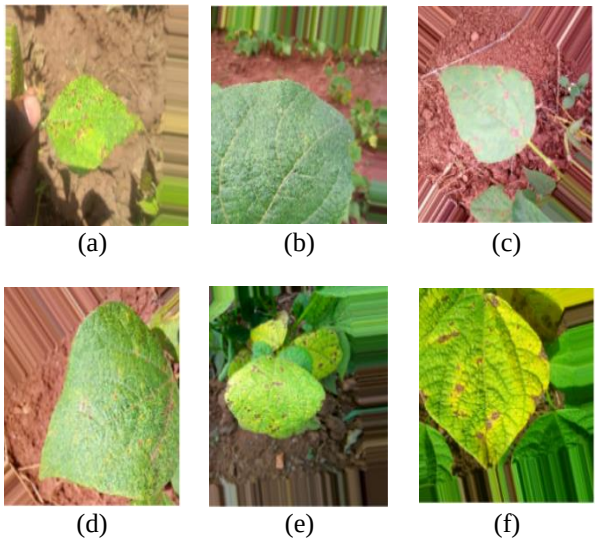
Figure. 2 Examples of bean leaf images: (a) angular leaf spot, (b) healthy, (c) bean rust, (d) healthy, (e) healthy, and (f) bean rust

Table 2. Depiction of the utilized datase

Category	Training	Validation	Testing	Total
Angular Leaf Spot	345	44	43	432
Bean Rust	348	45	43	436
Healthy	342	44	42	428
Total	1035	133	128	1296

3.2 Data augmentation

After preprocessing and splitting the data, data augmentation [22] is used during the training process to increase the dataset size and reduce the risk of overfitting, which is particularly useful when the training dataset is too small.



(a) (b) (c)

(d) (e) (f)

Figure. 3 Some samples of bean leaf images by utilizing different augmentation techniques: (a) angular leaf spot, (b) healthy, (c) angular leaf spot, (d) bean rust, (e) angular leaf spot, and (f) angular leaf spot

This strategy includes geometric transformations such as rotations, shifts, zooms, shears, and flips. The images are randomly rotated up to 40 degrees and shifted vertically or horizontally by up to 0.2. Shear, zoom, and horizontal flip are all set to 0.2. The final step involves scaling the image pixel values from integers (0-255) to floats (0-1). During data augmentation, each image in the training and validation datasets undergoes a randomly selected augmentation process before being fed into the artificial neural network, as depicted in Fig. 3. The test dataset, representing real-world, unseen data, does not undergo data augmentation to ensure an unbiased evaluation of the model's performance.

3.3 Bean leaf diseases detection using deep stacked ensemble learning model

This section detailes the pre-trained CNN models (EfficientNetB3, InceptionV3, and MobileNetV2) utilized for the stacked ensemble learning model and the proposed stacked ensemble learning model.

3.3.1. EfficientNetB3

EfficientNet [56] is a CNN architecture available in various versions, from EfficientNetB0 to EfficientNetB7. To achieve maximum model accuracy, EfficientNet models employ compound scaling, which balances the scaling of the convolutional network's size across width, depth, and resolution [23].

Compound scaling uses a compound coefficient to uniformly scale these dimensions, enabling efficient and balanced model growth [24]. For our classification task, we utilized EfficientNetB3, as larger networks with increased width, depth, or resolution generally achieve higher accuracy. The EfficientNetB3 model features a depth of 210 layers and consists of 11.1 million parameters. This deeper network captures intricate and rich features, ensuring better generalization to new tasks. Additionally, the wider network of EfficientNetB3 optimally extracts features and patterns, enhancing its performance in classification tasks. Fig. 4 shows the architecture of the proposed pre-trained EfficientNetB3 network architecture for the detection of bean leaf diseases.

3.3.2. InceptionV3

InceptionV3 [25] is a deep learning model based on CNNs designed for image classification. It was designed to facilitate deeper networks while keeping the parameter count manageable, containing fewer than 25 million parameters. The model's architecture comprises symmetric and asymmetric building blocks, such as fully connected layers, convolutions, max pooling, and dropouts [26]. It also makes extensive use of batch normalization for the activation inputs. The model calculates the loss using the SoftMax function. Fig. 5 illustrates the architecture of the proposed pre-trained InceptionV3 model for the detection of bean leaf diseases.

3.3.3. MobileNetV2

The MobileNetV2 model [27] is CNN that comprises 53 layers and 88 depthwise separable convolutions. MobileNetV2 is an extension of MobileNetV1, designed specifically for mobile devices. Running neural networks on mobile devices enhances model availability and provides benefits such as increased security and reduced energy consumption. MobileNets utilize depthwise separable convolutional layers as their fundamental building blocks. MobileNetV2 introduces two key features to improve performance: 1. **Linear bottlenecks between layers**: These help to preserve the essential information by compressing the data efficiently before passing it to the next layer. 2. **Shortcut connections between bottlenecks**: Similar to traditional residual connections, these shortcuts facilitate faster training and improved accuracy by enabling direct pathways for gradient flow. These innovations enable MobileNetV2 to maintain efficiency while achieving higher accuracy and speed in training. Fig. 6 illustrate the architecture of MobileNetV2 for detection bean leaf diseases. All models (EfficientNetB3, InceptionV3, and MobileNetV2) undergo fine-tuning for the detection of bean leaf diseases.

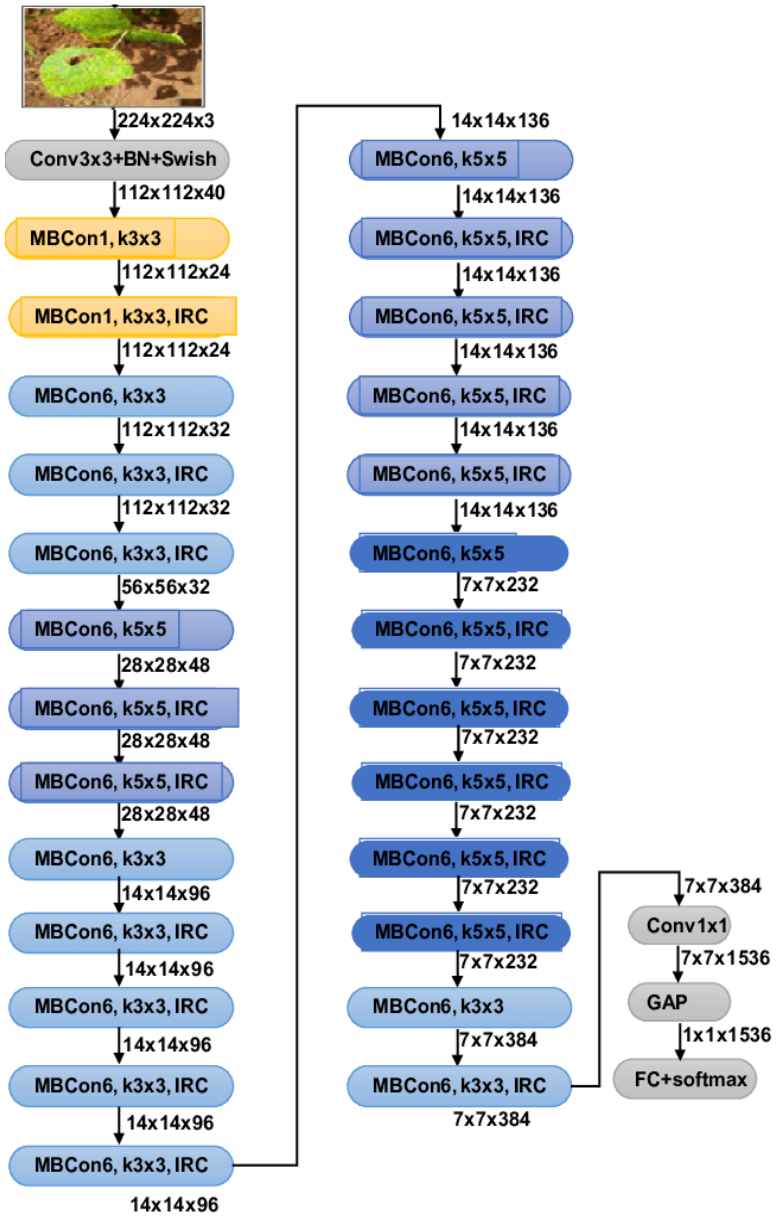


Figure. 4 Architectural design of fine-tuned EfficientNetB3 Model

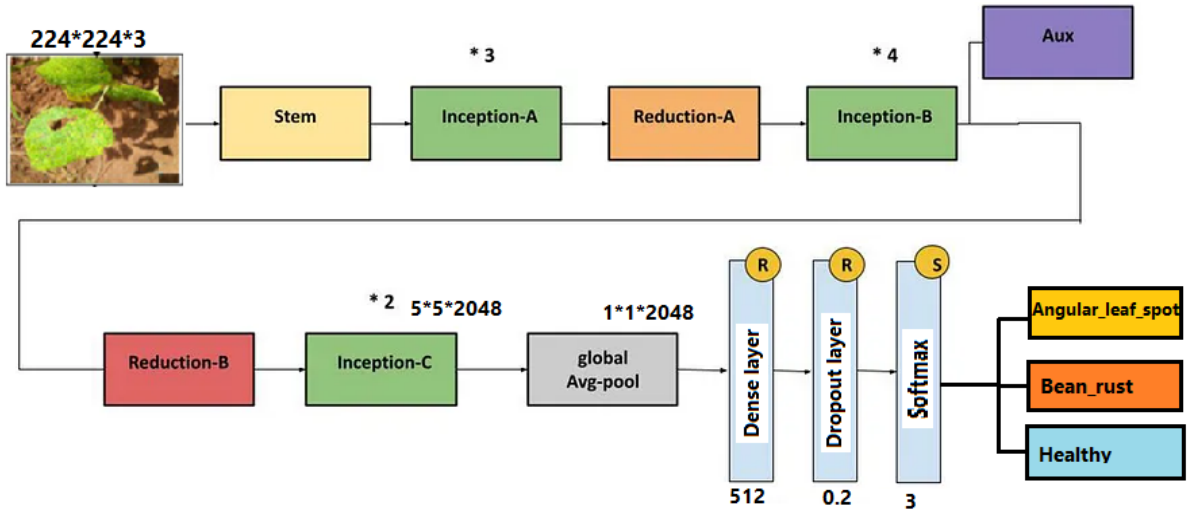


Figure. 5 Architectural design of fine-tuned InceptionV3 Model

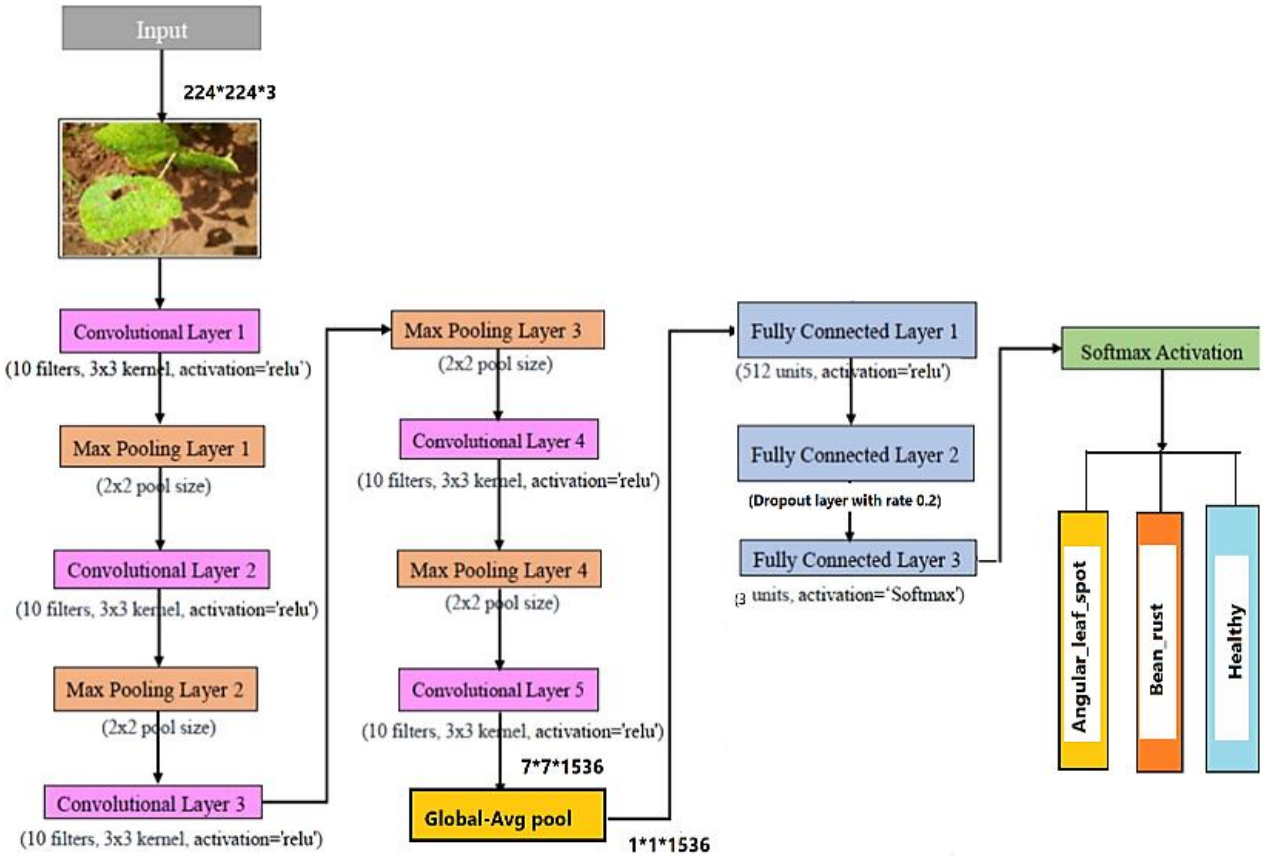


Figure. 6 Architectural design of fine-tuned MobileNetV2 Model

During fine-tuning, the following alterations are applied to the models for retraining the models utilizing the dataset [21]. Bean leaf images underwent preprocessing and normalization before being utilized for networks training. Then, data augmentation techniques were applied to enhance dataset processing efficiency. All layers within the networks were trainable, enabling them to extract features from the images effectively. For these models, the top layers are excluded to use their pre-trained weights and the acquired feature representations. The redesigned new classifier part of each model utilizes a global average pooling layer instead of a flattening layer after the feature extractor to reduce the number of learning parameters, followed by adding two dense layers with sizes of 512 and 3 neurons, respectively, as shown in Figs. 4-6. The output layer consists of three neurons for classifying images into three classes: angular leaf spots, bean rust, and healthy. Each dense layer, except the output layer, is followed by a dropout layer with a rate of 0.2. Dropout is employed during training to prevent overfitting by reducing the model's capacity. The dense layers utilize a Rectified Linear Unit (ReLU) activation function, while the output layer employs a softmax activation function for multi-class classification. Also, Fine-tuning each model includes retraining all layers of each model, except freezing the first ten layers of each model. The mathematical computation of the softmax activation function is as follows [28]:

$$Softmax\ (x_i) = \frac{e^{x_i}}{\sum_{j=0}^{n} e^{x_i}} \quad (1)$$

Where x_i denotes the input vector and n represents the number of classes.

3.3.4. Proposed stacked ensemble learning model

To improve the performance of the overall classification model, selecting a high-performance classification model as the base model is crucial. The classifier's predictive ability is closely linked to its capacity to extract high-quality features, making the choice of a high-performance CNN essential for feature extraction. Deep neural networks, known for their high capacity and flexibility, often exhibit high variance and low bias. Averaging the outputs of independent models can significantly reduce this variance. In this study, we address this using the average ensemble method by averaging the softmax probability values of all models, accommodating the varying output magnitudes from different models.

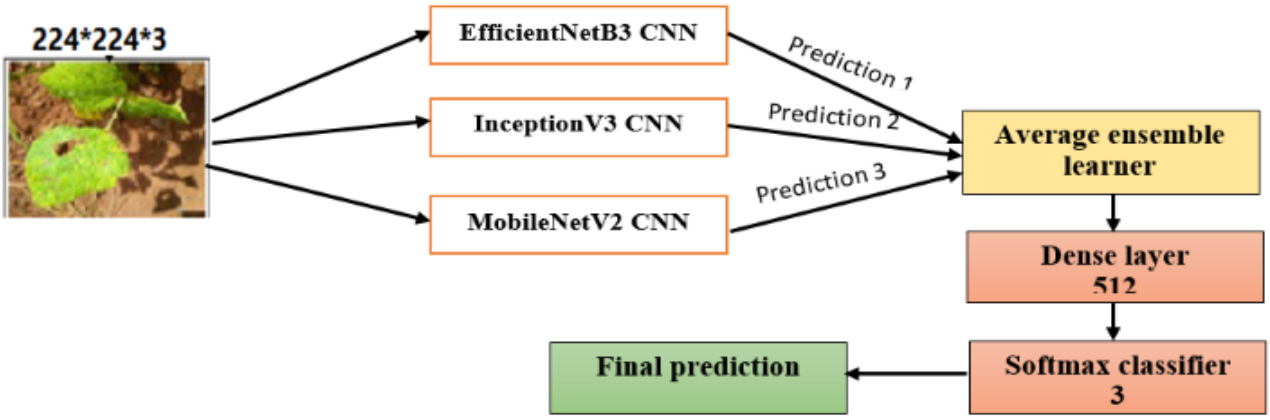


Figure. 7 The Architecture of the proposed deep stacked ensemble learning

The stacked ensemble learning model utilized in this work was created using multiple pre-trained CNN models, including EfficientNetB3, InceptionV3, and MobileNetV2, which shows more than 95% accuracy. The three individual classification models are combined to create a more robust overall model. The combination strategy involves several steps: first, we apply data augmentation to the original bean leaf images after preprocessing for the training and validation datasets. Then, each of the three pre-trained CNN models is trained independently on this augmented data. Finally, the softmax output probabilities of the three models are averaged to pass as input into a new classifier to produce the final output. Average prediction is calculated using the following Eq. (2):

$$\text{Average ensemble}=\text{argmax}\frac{1}{n}\sum_{j=1}^{n} p_{ij} \quad (2)$$

Where p_j the output propability of i^{th} class label of the j^{th} model and *n is* the total number of the single models.

The new classifier consist of two dense layers with 512, 3 neuronus, respectively. The output layer consists of three neurons for classifying images into three classes: angular leaf spots, bean rust, and healthy. The stacked average ensemble method leverages the diversity of base classifiers to construct a more robust and reliable overall model. This approach involves calculating the average of predictions from each individual model for a given input image. By averaging these predictions, the method reduces the impact of individual model variations and errors, thereby enhancing the reliability and accuracy of the final prediction. This stacked ensemble approach enhances the model's classification performance by leveraging the strengths of each individual model. The architecture of the proposed deep-stacked ensemble learning model is shown in Fig. 7.

4. Experimental results and analysis

Detecting bean leaf diseases is crucial for preventing their spread in farming environments. Therefore, conducting various experiments to evaluate the effects of multiple deep-learning models and stacked ensemble learning models on the bean leaf dataset [21] is inherently valuable. These experiments offer numerous benefits, including improved disease management and agricultural productivity. For this purpose, we carried out several experiments in this section to showcase the effectiveness of the proposed models on the dataset [21]. This section presents the results obtained from several experiments, the overall experimental analysis for bean leaf disease detection utilizing different pre-trained CNN (EfficientNetB3, InceptionV3, and MobileNetV2) models, and stacked ensemble learning models. Three stacked ensemble models are presented in this work. The first stacked ensemble model 1 combines the three pre-trained CNN models (EfficientNetB3, InceptionV3, and MobileNetV2), the second stacked ensemble model 2 combines EfficientNetB3 and MobileNetV2, while the third stacked ensemble model 3 combines EfficientNetB3 and InceptionV3. A comparative analysis of these models introduces and compares the results obtained from these models with recent state-of-the-art approaches. Finally, the most effective performing model is obtained.

All codes of the proposed models were written in python and trained on a kaggle where a kaggle search project is created to supply everybody with free NVIDIA Tesla-P100 GPU resources for their deep learning projects and research. Each user is presently specified 16GB of RAM, and it will be up to 29GB.

4.1 Training and validation accuracy

In the training step, all pre-trained (EfficientNetB3, InceptionV3, and MobileNetV2 models used in this study undergo fine-tuning for the detection and classification of bean leaf images. Fine-tuning each model includes retraining all layers of each model, except freezing the first ten layers of each model to update their weights with the utilized dataset [21]. The weights parameters of these layers are fine-tuned utilizing optimizer Adam with categorical cross-entropy as the loss function. The categorical cross-entropy function measures the performance of a classification model whose output (class score) is a probability value between 0 and 1. Categorical cross-entropy is calculated as [28]:

$$L(y,p) = -\sum_{j=0}^{n} y_j \, log \, (p_j) \quad (3)$$

Where y_j is the actual value and p_j is the predicted value.

The early stopping technique is used in this work to stop the training process after 15 epochs if the validation accuracy does not improve to prevent the proposed model from overfitting. The hyperparameters for these networks can be indicated in Table 3.

Table 3. Training hyperparameters used while fine-tuning the deep learning model

Hyperparameters of Hybrid model	
Learning rate	0.0001
Optimizer	Adam
Batch Size	32
Max Epochs	50
Early stopping	15 epochs

Table 4. The accuracy rates of training, and validation for all the proposed models

Accuracy / Model	Tr-Acc%	Val-Acc%	Training time
EfficientNetB3	99.6	97.7	0:11:34
InceptionV3	98.4	98.4	0:09:32
MobileNetV2	99.8	96.9	0:13:01
Stacked ensemble model 1	99.2	99.2	0:08:16
Stacked ensemble model 2	99.4	99.2	0:06:44
Stacked ensemble model 3	99.8	99.2	0:06:52

Table 4 shows the accuracy rates for training and validation datasets for three pre-trained models (EfficientNetB3, InceptionV3, and MobileNetV2) alongside the proposed stacked ensemble models. The EfficientNetB3 model attained an accuracy of 99.6% and 97.7% for training and validation datcompleting training in 11 minutes and 34 seconds. In comparison, the InceptionV3 model achieved accuracies of 98.4% and 98.4% for training and validation data with a training time of 9 minutes and 32 seconds. The MobileNetV2 model achieved an accuracy of 99.8% and 96.9% for training and validation data within a training time of 13 minutes and 1 second. The stacked ensemble models showed notable performance improvements. Stacked ensemble model 1, which combines EfficientNetB3, InceptionV3, and MobileNetV2, achieved an accuracy of 99.2% on training data and 99.2% on validation data, with a training time of 8 minutes and 16 seconds. Stacked ensemble model 2, combining EfficientNetB3 and MobileNetV2, reached an accuracy of 99.4% on training data and 99.2% on validation data, with a reduced training time of 6 minutes and 44 seconds. Another stacked ensemble model 3 combining EfficientNetB3 and InceptionV3 also achieved high accuracy rates of 99.8% on training data and 99.2% on validation data, with a training time of 6 minutes and 52 seconds. The training and validation accuracy curves with the number of training epochs are displayed in Fig. 8 for all the proposed models to show their performance in classifying bean leaf diseases.

As shown in Table 4 and Fig. 8, the stacked ensemble models, especially those combining EfficientNetB3 and InceptionV3 or EfficientNetB3 and MobileNetV2, achieve superior validation accuracy compared to individual models, indicating better generalization and robustness.The individual models, while strong, particularly MobileNetV2, show signs of overfitting. Stacked ensemble models generally require less training time compared to the average of individual models, with Ensemble Model 2 (EfficientNetB3, MobileNetV2) being the most efficient. The ensemble models balance training and validation accuracy more effectively than individual models. This balance is crucial for real-world applications where unseen data must be accurately predicted. In conclusion, stacked ensemble models not only enhance performance by improving validation accuracy and mitigating overfitting but also do so efficiently with reduced training times. This makes them a preferable choice for robust and reliable classification tasks.

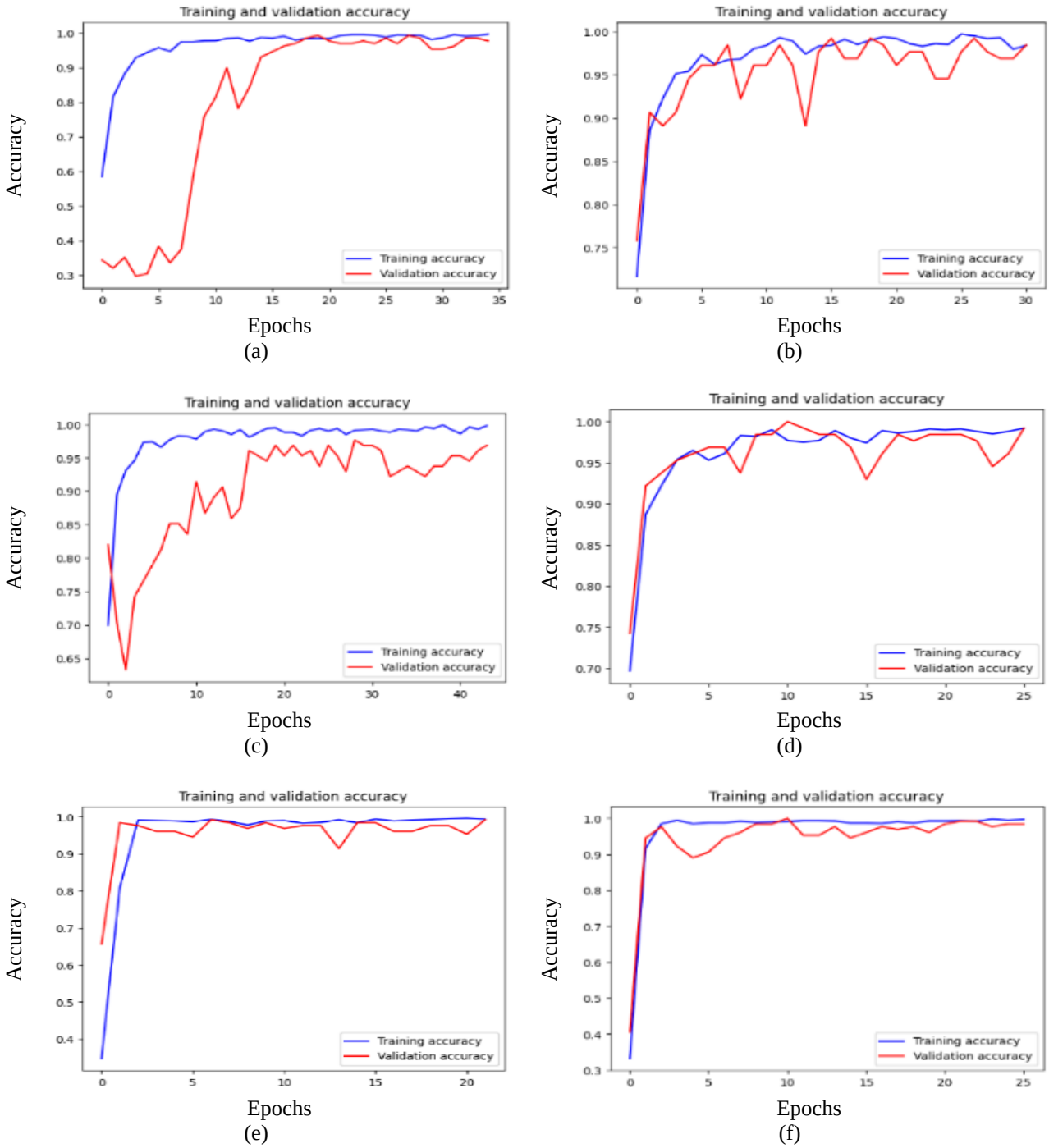


Figure. 8 Comparison of training and validation accuracy for: (a) EfficientNetB3 model, (b) InceptionV3 model, (c) MobileNetV2 model, (d) stacked ensemble model 1, (e) stacked ensemble model 2, and (f) stacked ensemble model 3

4.2 Evaluation metrics of model performance

Different performance metrics were utilized in this work to assess the performance of the proposed models. We measured the classification performance of the proposed models by utilizing several metrics like Accuracy, Recall, Precision, and F1-score. [29, 30].

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

$$Recall = \frac{TP}{(TP+FN)} \tag{5}$$

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{6}$$

$$Accuracy = \frac{(TN+TP)}{(TP+TN+FP+FN)} \tag{7}$$

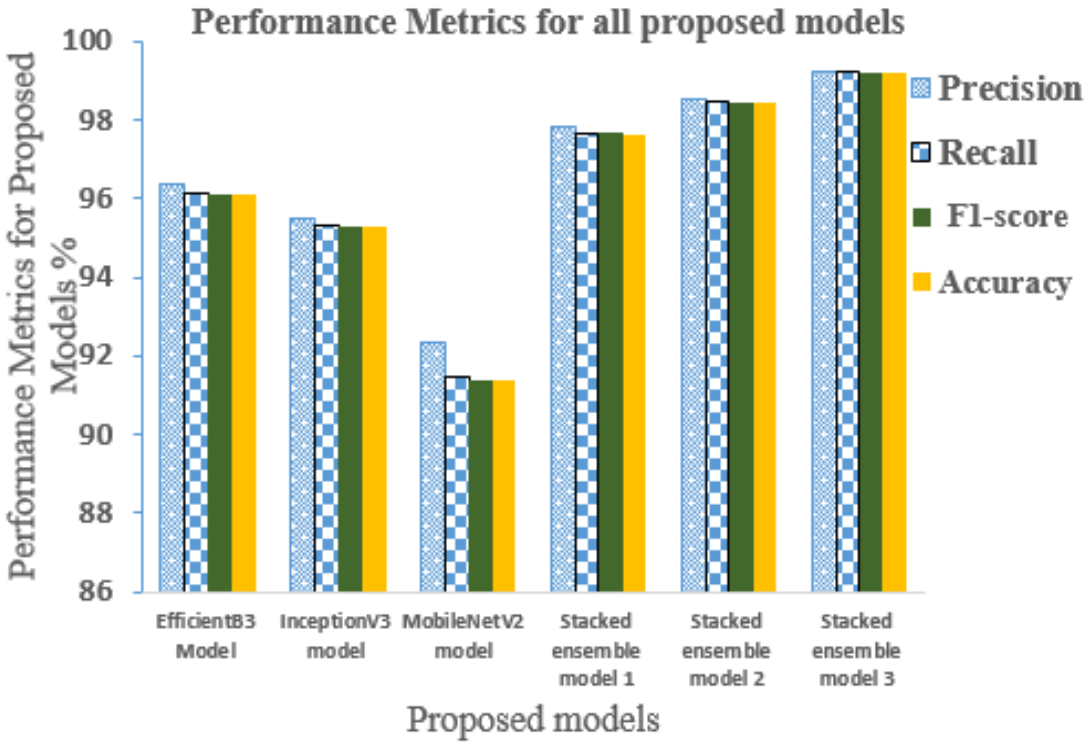


Figure. 9 Comparison of performance metrics for all proposed models

Table 5. Performance metrics for the proposed model

Models \Metrices	Precision %	Recall %	F1-score%	Accuracy %
EfficientB3 model	96.38	96.12	96.09	96.09
Inception V3 model	95.51	95.33	95.31	95.31
MobileNetV2 model	92.34	91.47	91.40	91.40
Stacked ensembel model 1	97.82	97.67	97.67	97.65
Stacked ensembel model 2	98.51	98.45	98.45	98.44
Stacked ensembel model 3	99.24	99.22	99.22	99.22

Where TP signifies the count of True Positive samples, denoting instances accurately predicted as belonging to the positive category. TN stands for True Negative samples, representing instances correctly predicted as belonging to the negative category. FP corresponds to False Positive samples, indicating instances inaccurately predicted as positive. Lastly, FN represents the count of False Negative samples, highlighting instances incorrectly predicted as negative when they are positive.

Table 5 presents the performance metrics for all pre-trained CNN (EfficientNetB3, InceptionV3, and MobileNetV2) and the stacked ensemble proposed models. Fig. 9 shows the comparative performance of Precision, Recall, F1-score, and testing accuracy for all proposed models. According to Table 5 and Fig. 9, the stacked ensemble model 3, which combines EfficientNetB3 and InceptionV3, outperformed (EfficientNetB3, InceptionV3, and MobileNetV2) in all metrics, achieving the highest percentages for Percision (99.24%), Recall (99.22%), and F1-score (99.22%) along with the highest testing accuracy of 99.22%. The testing accuracy for EfficientNetB3, InceptionV3, MobileNetV2, and the stacked ensemble proposed model 1, model 2, and models 3 are 96.09%, 95.31%, 91.40%, 97.65%, 98.44%, and 99.22%, respectively. The stacked ensemble model 3, which utilizes a meta-learner trained on the averaged predictions of EfficientNetB3 and InceptionV3, demonstrates superior performance compared to the other stacked proposed models and individual models, indicating that these models together provide the most comprehensive feature extraction and classification capability. The results show that the overall accuracy of the proposed stacked ensemble model increases significantly, reaching its peak accuracy when combining multiple predictions from two or three robust networks rather than relying solely on each network individually, where the stacked ensemble method reduces the impact of individual model variations and errors, thereby enhancing the reliability and accuracy of the final

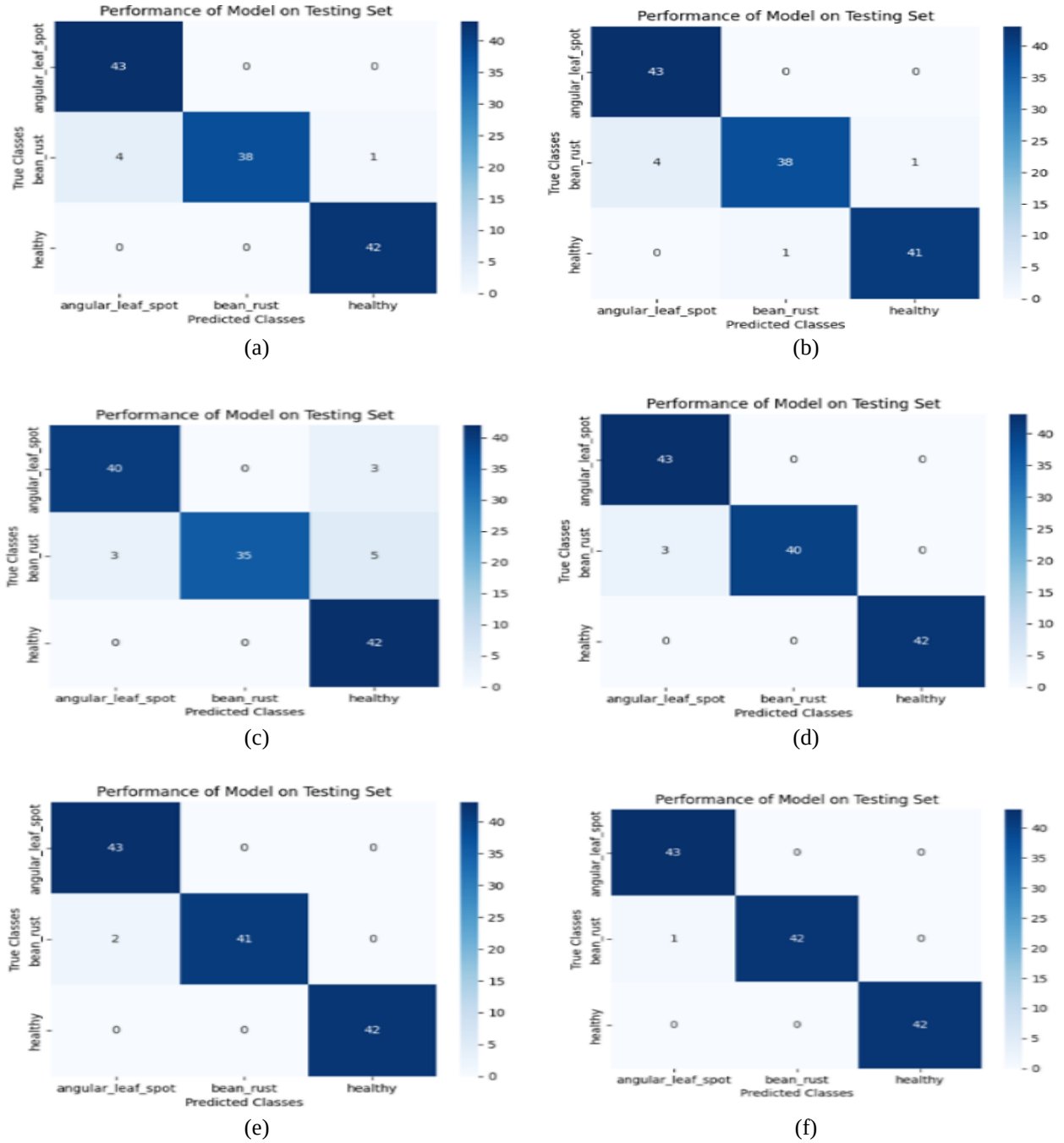


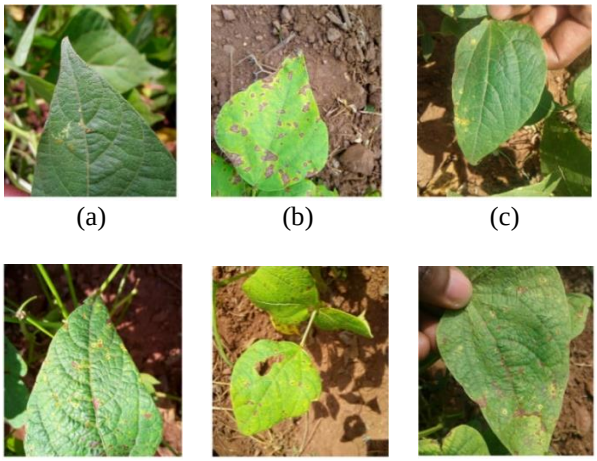
Figure. 10 The confusion matrix for: (a) EfficientNetB3 model, (b) InceptionV3 model, (c) MobileNetV2 model, (d) stacked ensemble model1, (e) stacked ensemble model 2, and (f) stacked ensemble model 3

prediction. This stacked ensemble model improves the model's classification performance by leveraging the strengths of each individual model.

The confusion matrix performs as a tabular representation to visually evaluate the performance of a prediction model. It keeps counts of predictions made by the model, distinguishing between correct and incorrect classifications in each cell. Fig. 10 displays the confusion matrices of EfficientNetB3, InceptionV3, MobileNetV2, and the stacked ensemble proposed model utilized in this work.

In Fig. 10, the confusion matrix reveals that the EfficientNetB3 model (Fig. 10(a)) correctly predicted all images of angular leaf spots and healthy classes. For the "bean rust" class, it identified 38 images correctly and misclassified 5. Conversely, the

InceptionV3 model (Fig. 10(b)) correctly identified all "angular leaf spots" images, while in the "bean rust" class, 38 images were correctly classified and misclassified 5. For the "healthy" class, it classified 41 images correctly and misclassified one as bean rust.



(a) (b) (c)

(d) (e) (f)

Figure. 11 Some correct and wrong predictions made on test data: (a) True Label: healthy Predicted Label: healthy, (b) True Label: angular leaf spot Predicted Label: angular leaf spot, (c) True Label: bean rust Predicted Label: bean rust, (d) True Label: bean rust Predicted Label: healthy, (e) True Label: bean rust Predicted Label: angular leaf spot, and (f) True Label: bean rust Predicted Label: angular leaf spot

The MobileNetV2 model (Fig. 10(c)) correctly identified 40 “angular leaf spots” images with three misclassifications as healthy, while the bean rust class correctly identified 35 images and misclassified 8. In the “healthy” class, it identified all images correctly. The stacked ensemble model 1, which combines (EfficientNetB3, InceptionV3, and MobileNetV2) (Fig. 10(d)) demonstrated robust performance by correctly identifying all images of angular leaf spots, bean rust, and healthy classes except for three image misclassifications as angular leaf spots in the “bean rust” class, while The stacked ensemble model 2, which combines (EfficientNetB3 and MobileNetV2) (Fig. 10(e)) demonstrated robust performance by correctly identifying all images of angular leaf spots, bean rust, and healthy classes except for two image misclassifications as angular leaf spots in the “bean rust” class, The stacked ensemble model 3, which combines (EfficientNetB3 and InceptionV3) (Fig. 10(f)) demonstrated robust performance by correctly identifying all images of angular leaf spots, bean rust, and healthy classes except one image misclassifications as angular leaf spots in the “bean rust” class. The results from the confusion matrices in Fig. 10 highlight the substantial benefits of using stacked ensemble models over individual models. The ensemble approach, especially Stacked Ensemble Model 3, demonstrates superior performance by correctly classifying the majority of images across all classes with minimal errors. This underscores the effectiveness of combining EfficientNetB3 and InceptionV3, as their complementary strengths lead to enhanced model accuracy and robustness. The ensemble method reduces the impact of individual model errors, resulting in a more reliable and accurate classification system. Fig. 11 shows some correct and wrong predictions made on test data.

4.3 Comparative analysis

In this section, we thoroughly evaluate the effectiveness of our proposed stacked ensemble model for detecting bean leaf diseases, performing a comprehensive comparison with state-of-the-art models. The proposed stacked ensemble models, which combines the predictions of multible pre-trained models (EfficientNetB3, InceptionV3, and MobileNetV2), underwent rigorous training and evaluation on the bean leaf dataset [21]. The resulting classification report highlights the model’s strong performance, demonstrating high accuracy, precision, recall, and F1 score, validating its effectiveness in bean leaf disease detection. However, to further validate its performance, we compare it against all models introduced in this study and established state-of-the-art models. This comparative analysis primarily focuses on precision Recall, f1-score, and accuracy metrics. Table 6 and Fig. 12 present a comparison between the proposed models and recent existing models executed on the [21] dataset. Notably, The EfficientNetB3 model achieved a testing accuracy of 96.09%. While it performed well, it struggled with certain misclassifications, particularly in the bean rust class. The InceptionV3 model achieved a testing accuracy of 95.31%. Similar to the EfficientNetB3 model, it had a few misclassifications, with one misclassification in the healthy class, while the MobileNetV2 model achieved a testing accuracy of 91.40%. This model showed the highest number of misclassifications, indicating a lower performance compared to EfficientNetB3 and InceptionV3. In Stacked Ensemble Models, the Stacked Ensemble Model 1 (EfficientNetB3, InceptionV3, and MobileNetV2) achieved a testing accuracy of 97.65%. This model demonstrated strong performance, significantly reducing the number of misclassifications across all classes. The Stacked Ensemble Model 2 (EfficientNetB3 and MobileNetV2) achieved a testing accuracy of 98.44%. This model further improved accuracy, indicating that even with fewer models, a well-chosen combination can outperform a larger ensemble. The stacked ensemble model 3, which combines EfficientNetB3 and InceptionV3, outperformed (EfficientNetB3, InceptionV3, and MobileNetV2), stacked ensemble model 1, stacked ensemble model

Table 6. Comparison of the proposed stacked ensemble model with existing models

Models	Precision%	Recall%	F1-score%	Accuracy%
Sahu et al. [13] (2020)	-	-	-	96.06
Sahu et al. [14] (2021)	-	-	-	95.31
Abed et al. [15] (2021)	-	-	-	90.01
Elfatimi et al. [16] (2022)	92.98	93.02	92.94	92.97
Önler [17] (2023)	-	-	-	99.2
Elfatimi et al. [20] (2024)	92.98	93.02	92.94	92.97
EfficientNetB3 proposed model	96.38	96.12	96.09	96.09
InceptionV3 proposed model	95.51	95.33	95.31	95.31
MobileNetV2 proposed model	92.34	91.47	91.40	91.4
Stacked ensemble proposed model 1	97.82	97.67	97.67	97.65
Stacked ensemble proposed model 2	98.51	98.45	98.45	98.44
Stacked ensemble proposed model 3	99.24	99.22	99.22	99.22

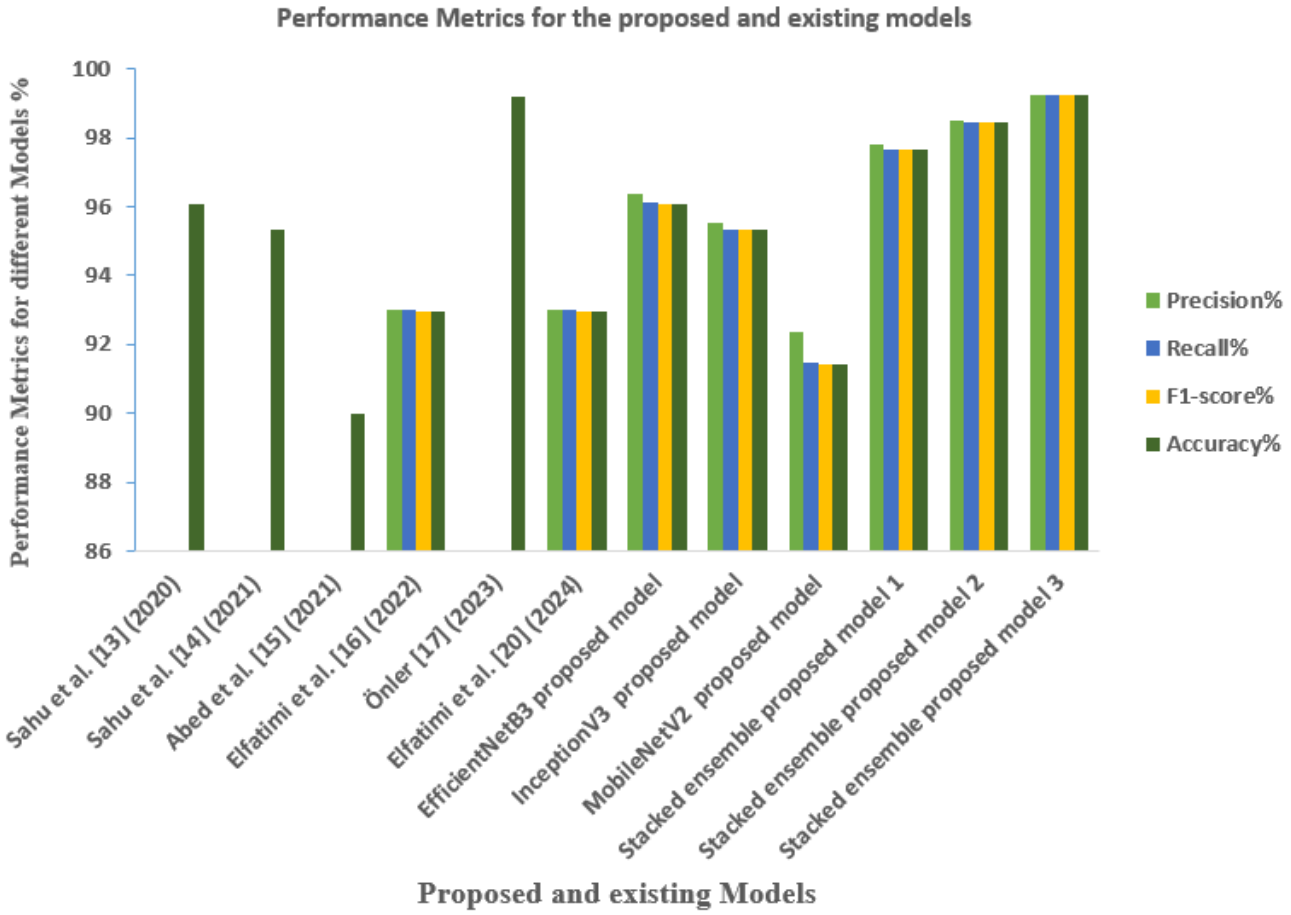


Figure 12. Comparison of precision, recall, F1 score, specificity, sensitivity and testing accuracy for different models

2, and most existing models in all metrics, achieving the highest percentages for Percision (99.24%), Recall (99.22%), and F1-score (99.22%) along with the highest testing accuracy of 99.22%. This model showed the best performance, with the least number of misclassifications, highlighting the complementary strengths of EfficientNetB3 and InceptionV3. The performance improvement can be attributed to the ensemble's ability to mitigate the weaknesses of individual models and enhance overall robustness. This underscores the significance of the ensemble approach in improving model performance for complex classification tasks like bean leaf disease detection.

5. Conclusion

This study introduces a robust stacked ensemble learning model for bean leaf disease detection by leveraging the strengths of multiple pre-trained CNN models based on TL and utilizing a meta-learner on

averaged predictions. This method reduces the impact of individual model variations and errors by averaging these predictions, thereby it appears as a more efficient and effective way to improve model performance, especially when the dataset is small and you need faster training. The experiments have shown that the stacked ensemble model 3, which combines EfficientNetB3 and InceptionV3, demonstrates superior performance in detecting bean leaf diseases, surpassing both the individual TL models and most state-of-the-art models in all metrics, achieving the highest percentages for Percision (99.24%), Recall (99.22%), and F1-score (99.22%) along with the highest testing accuracy of 99.22% and with reduced training time.

Future research can explore more sophisticated meta-learning techniques, including gradient boosting machines or neural network-based Meta learners, to further improve the ensemble's performance. Additionally, expanding the dataset and including more diverse disease types could enhance the model's generalizability. We also expand this research to use a hybrid model between stacked ensemble learning and semantic ontology to make accurate detection for bean leaf diseases.

Conflicts of Interest

The authors state no conflict of interest regarding the publication of this paper.

Author Contributions

Conceptualization, Ola M. El Zein; methodology, Ola M. El Zein; software, Ola M. El Zein; validation, Naglaa E. Ghannam, Ola M. El Zein, and Doaa R. Fathy; formal analysis and investigation, Ola M. El Zein; resources, Naglaa E. Ghannam, Ola M. El Zein, Doaa R. Fathy; data curation, Ola M. El Zein; writing—original draft preparation, Ola M. El Zein; writing—review and editing, Naglaa E. Ghannam, Ola M. El Zein, Doaa R. Fathy; visualization, Naglaa E. Ghannam and Ola M. El Zein; supervision, Naglaa E. Ghannam and H. Mancy; project administration, Naglaa E. Ghannam and H. Mancy; funding acquisition.

Acknowledgments

This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

References

[1] S. Lee, C. Chan, S. J. Mayo and P. Remagnino, "How deep learning extract and learns leaf features for the plant classification", *Pattern Recognition*, Vol. 71, pp. 1-13, 2017. https://doi.org/10.1016/j.patcog. 2017.05.015.

[2] K. Yin, J. L. Qiu, "Genome editing for plant disease resistance: applications and perspectives", *Phil.Trans. R. Soc. B*, Vol. 374, 2019, doi: 10.1098/rstb.2018.0322

[3] S. P. Singh, "Production and utilization", *Common bean improvement in the twenty-first century. Springer, Dordrecht, the Netherlands*, Vol. 7, pp. 1-24, 1999, doi: 10.1007/978-94-015-9211-6_1.

[4] W.J. Broughton, G. Hernández, M. Blair, S. Beebe, P. Gepts P and J. Vanderleyden, "Beans (Phaseolus spp.); model food legumes", *Plant and Soil*, Vol. 252, pp. 55-128, 2003, doi: 10.1023/A:1024146710611

[5] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learningbased detector for real-time tomato plant diseases and pests recognition", *Sensors*, Vol. 17, No. 9, p. 2022, 2017, doi: 10.3390/s17092022.

[6] D. Mawejje, P. Pamela, and M. Ugen, "Severity of angular leaf spot and rust diseases on common beans in central Uganda", *Uganda Journal of Agricultural Sciences*, Vol. 15, No. 1, pp. 63-72, 2014.

[7] M. M. Nay, T. L. P. O. Souza, B. Raatz, C. M. Mukankusi, M.C. Gonçalves-Vidigal, A. F. B. Abreu, L. C. Melo, and M. A. Pastor-Corrale, "A Review of Angular Leaf Spot Resistance in Common Bean", *Crop Science*, Vol. 59, pp. 1376-1391, 2019, doi: 10.2135/cropsci2018.09.0596.

[8] A. Sagar and J. Dheeba, "on using transfer learning for plant disease detection", *BioRxiv*, pp. 1-8, 2020, doi: 10.13140/RG.2.2.12224. 15360/1.

[9] A. Kamilaris and F.X. Prenafeta-Boldú, "Deep learning in agriculture: A survey", *Comput. Electron. Agric.*, Vol. 147, pp. 70-90, 2018, doi: 10.1016/j.compag.2018.02.016.

[10] K. Muthukannan, P. Latha, R.P. Selvi, and P. Nisha, "Classification of diseased plant leaves using neural network algorithms", J. *Eng. Appl. Sci*, Vol. 10, No. 4, 2015.

[11] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks", *International Symposium on Visual Computing*, pp. 638-645, 2015, doi: 10.1007/978-3-319-27863-6_59.

[12] Devaraj P, M. P. Arakeri and B.P. Vijaya Kumar, "Early detection of leaf diseases in Beans crop using Image Processing and Mobile

Computing techniques", *Advances in Computational Sciences and Technology*, Vol. 10, No.10, pp. 2927-2945, 2017.

[13] P. Sahu, A. Chug, A. P. Singh, D. Singh and R. P. Singh, "Implementation of CNNs for Crop Diseases Classification: A Comparison of Pre-trained Model and Training from Scratch", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 20, No. 10, pp. 206-215, 2020.

[14] P. Sahu, A. Chug, A. P. Singh, D. Singh and R. P. Singh, "Deep Learning Models for Beans Crop Diseases: Classification and Visualization Techniques", *International Journal of Modern Agriculture*, Vol. 10, No.1, pp. 796-812, 2021.

[15] S. H. Abed, A. S. Al-Waisy, H. J. Mohammed and Sh. Al-Fahdaw, "A modern deep learning framework in robot vision for automated bean leaves diseases detection", *International Journal of Intelligent Robotics and Applications, springer*, Vol. 5, pp. 235-251, 2021, doi: doi.org/10.1007/s 41315-021-00174-3.

[16] E. Elfatimi, R. Eryigit and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models", *IEEE Access*, Vol.10, pp. 9471-9482, 2022, doi: 10.1109/ACCESS.2022 .3142817

[17] E. Önler, "Feature fusion based artificial neural network model for disease detection of bean leaves", *Electronic Research Archive*, Vol. 31, No. 5, pp. 2409-2427, 2023, doi: 10.3934/era.2023122.

[18] V. Singh, A. Chug and A. P. Singh, "Classification of Beans Leaf Diseases using Fine Tuned CNN Model", In: *Proc. of International Conference on Machine Learning and Data Engineering, Procedia Computer Science, Elsevier*, Vol. 218, pp. 348-356, 2023, doi: 10.1016/j.procs.2023.01.017.

[19] S. A. Suma, A. Haque, N. Vasker, M. Hasan, J. A. Ovi and M. Islam, "Beans Disease Detection Using Convolutional Neural Network", In: *Proc. of 2023 4th International Conference on Big Data Analytics and Practices (IBDAP)*, Bangkok, Thailand, pp. 1-5, 2023, doi:10.1109/IBDAP58581.2023. 10271983.

[20] E. Elfatimi, R. Eryig and H. A. Shehu, "Impact of datasets on the effectiveness of MobileNet for beans leaf disease detection", *Neural Computing and Applications, springer*, Vol. 36, pp. 1773-1789, 2024, doi: 10.1007/s00521-023-09187-4.

[21] Makerere AI Lab, "Bean disease dataset", 2020. Available from: https://github.com/AI-Lab-Makerere/ibean.

[22] A. Mikołajczyk, M. Grochowski, "Data augmentation for improving deep learning in image classification problem", In: *Proc. of 2018 International interdisciplinary PhD workshop (IIPhDW)*, pp. 117-122, 2018, doi: 10.1109/iiphdw. 2018.8388338.

[23] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention", *IEEE Access*, Vol. 9, pp. 14078-14094, 2021, doi: 10.1109/ ACCESS.2021.3051085.

[24] I. Papoutsis, N. I. Bountos, A. Zavras, D. Michail, and C. Tryfonopoulos, "Benchmarking and scaling of deep learning models for land cover image classification", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 195, pp. 250-268, 2023, doi: 10.1016/j.isprsjprs.2022.11.012.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826, 2015, doi: 10.48550/arXiv.1512. 00567.

[26] M. Nikhitha, S. R. Sri, and B. U. Maheswari, "Fruit recognition and grade of disease detection using inception v3 model", In: *Proc. of International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1040-1043, 2019, doi: 10.1109/ICECA.2019.8822095.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", In: *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510-4520, 2018, doi: 10.48550/arXiv.1801.04381.

[28] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, " Deep Learning", *Cambridge, MIT press*, Vol. 1, No. 2, pp. 1-800, 2016.

[29] J. Han, M. Kamber and J. Pei, *Data mining: concepts and techniques*, Morgan Kaufmann, 4th edition, 2012.

[30] Y. Kim. "Convolutional Neural Networks for Sentence Classification", In: *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751, 2014, doi: 10.48550/arXiv.1408.5882.