

Developing Common Bacteria Blight Disease Detection Model for Common Bean Using Deep Learning Approach

Assefa Senbato (✉ ayese96@gmail.com)

Bule Hora University <https://orcid.org/0000-0002-7411-9456>

Tsion Ayalew

Bule Hora University

Research Article

Keywords: Common bean, Common bacterial blight disease, Plant disease, Convolutional Neural Network

Posted Date: October 31st, 2023

DOI: <https://doi.org/10.21203/rs.3.rs-3512095/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Developing Common Bacteria Blight Disease Detection Model for Common Bean Using Deep Learning Approach

Assefa Senbato¹[0000-0002-7411-9456] and Tsion Ayalew²

¹ Department of Computer Science, Informatics College, Bule Hora University, Bule Hora, Ethiopia

² Department of Software Engineering, College of Informatics, Bule Hora University, Bule Hora, Ethiopia

assefa.senbeto@bhu.edu.et

Abstract

In many farm locations in Ethiopia, common bacterial blight (CBB), one of the common bean's known diseases, impacts the crop, lowers yield by up to 45 percent of the production, and also has an impact on seed quality. Recently, many researchers have tried to overcome this disease by conducting field surveys, crop management activities, crop rotation, chemical treatment, proper cultural practices, and integrated disease management. However, this method is a very tedious, time-consuming, and costly technique that requires more experts. To overcome these issues, a modern deep learning approach is proposed for the early detection of common bean disease. The three main phases are followed for the proposed approach. Firstly, the collection of healthy and diseased common bean images with the help of domain experts from different agricultural research centers is done. Then the design of a modern convolutional neural network that can detect and classify the input image as diseased or healthy is done. Lastly, the designed model is trained and evaluated. During the classification, the designed model is assessed against the performance of two pre-trained models (VGG16 and InceptionV3) to accurately detect the common bean disease. The performance of the proposed model is evaluated using a dataset that contains total images of 3,135 with two classes of healthy and common bacterial blight disease. We have used data augmentation techniques to generate more images to fit the proposed model. Based on our experimental results, the best performance is achieved using a proposed model with a classification accuracy of 98.2% for early detection of common bacterial blight disease on a common bean leaf. The result of this study can help farmers and domain experts by detecting CBB disease at an early stage for the necessary treatments.

Keywords: Common bean, Common bacterial disease, Plant disease, Leaf, deep learning, Convolutional Neural Network

1 Introduction

Agriculture is the backbone of the Ethiopian Economy. Thus, most Ethiopian people are working in farming activities, which act as the main source of income and food for Ethiopian people[1],[2]. Common bean is one of the most known crops enriched with the content of protein [3], and calories and also helps to prevent malnutrition. In Ethiopia, the most common bean production areas are found in Oromia, Benshangul-Gumuz, SNNPR, Tigray, and Amhara regional states. It acts as the main food and cash crop for Ethiopian farmers. However, this common bean crop is affected by different kinds of diseases, which impact the farmers as they do not get a high amount of production from their work. Common bacterial blight (CBB), common mosaic virus, rust, anthracnose, web blight, and angular leaf spot are the diseases that harm this crop. Common bacterial blight, rust, and anthracnose are the most significant and extensively distributed of these diseases [4], while the rest, are primarily constrained by their distribution. The gram-negative bacterial

pathogen *Xanthomonas axonopodis* pv. *phaseoli* (Xap) and its fuscans variant *Xanthomonas fuscans* (Xff), which causes CBB[2]; is the main obstacle to common bean production globally. CBB is proven to be a major problem in the majority of common bean production areas around the world including Ethiopia. CBB was reported in many parts of Ethiopia and is the reason for the reduction of yield by 22.4%[5] in the eastern part of the country. It[6] enclosed 98.32 thousand ha of land with a productivity of 1.62 t ha⁻¹ of grain yield in southern Ethiopia. However, productivity is even below the national average of 1.69-ton ha⁻¹.

CBB disease which affects the common bean leads to big damage [6] to bean crops and decreases their productivity. Treating the common bean crop disease in its early stages can increase[7], [8] the quality and quantity of the products. Recently many researchers from the agriculture field of study in Ethiopia tried to overcome the impact of this disease on the common bean in many ways like using[2], [8] the most preferred seeds, detecting disease by observation, and also using drugs to prevent disease. This is a very

tedious, time-consuming, and costly technique that needs many experts as well as more effort[7]. Besides, it is not easy to control and handle the crops in a huge area. In different countries, an enormous number of farmers lack the facilities and experience to deal with crop diseases as well as it is difficult to get the accessibility of domain experts easily. This leads to the introduction of advanced technologies that help farmers detect disease at an early stage of production and also give desired decision at the right time for treatment to get quality and high productivity. Also reduces cost, effort, and time and makes the ways we detect a disease from crop easily.

In Ethiopia[5] common bean is the most important crop that acts as the main food that contains many nutrients as well as a cash crop for many of the farmers in Ethiopia. During[4] 2016/17 cropping period time, in Ethiopia, common bean clothed 290,202 hectares of farmland and harvested 4,839,227 quintals of made with a ratio of 16.7 quintals per hectare and it was produced by around 4 million head house grasp farmers in Ethiopia. However, this common bean crop does not result in as much productivity as it is expected from it, because this known crop is affected by many diseases which cause the reduction of productivity in a large percentage (45%). Among those diseases, CBB is the most common disease that affects this crop. This is because the disease is not detected early.

1.1 Statement of the Problem

Since it is the main cash and food crop in Ethiopia, the common bean has a substantial economic impact[9]. However, these crops are constrained by several disease infections like bacterial, viral, and fungal[2], [3], [6]. This will cause big damage to common bean crops and decrease their productivity. Among several diseases of common bean crops, the most significant factor that adversely impacts common bean harvests in all growing regions is common bacterial blight (CBB). Authors[4] assessed a total of 85 bean fields in four common bean productive districts in low and mid-land of southwestern, Ethiopia and identified a high CBB disease rate. Strictly high damage was recorded at Debub Ari, 77.6 and 52.7% respectively. It is a serious bacterial disease of the common bean that causes lesions on the plant's leaves, stems, pods, and seeds. The disease affects seed quality and can degrade yield by up to 45%, maybe more in vulnerable cultivars[2].

Currently, with farmers and agriculture experts, crop disease detection is carried out using human eyes by observing experts identify them. This detection technique is, however, very tedious, and very costly because it needs more expert efforts, is time-consuming, less accurate, and subjective. The decision-making ability of human performance also

depends on his or her bodily state, including weariness and vision, a biased state of mind, workload, and pressure, and work conditions like uncomfortable and unstable weather conditions. In a developing country like Ethiopia, a number of farmers are uneducated and lack the facilities and experience to deal with disease detection as well as have problems in finding and consulting experts. Nowadays in Ethiopia, most researchers, those from agriculture fields have proposed some methods like using the most preferred seeds, detecting disease by observation, and also using drugs(chemicals) to prevent disease. However, this is very complex, requires more time, and needs much expertise(cost)and effort.

In Today's digital era, there are many advanced technologies like computer vision and artificial intelligence [7](AI) that help the farmer detect disease in its early stages of production and also give the desired decision at the right time for treatment to get quality and high productivity. This approach can reduce cost, effort, and time and makes the ways we detect a disease from crops easy[10]. In the realm of machine learning (ML), deep learning (DL) is a sophisticated technology that has increasingly taken the lead in many industries. Currently, it is essential for the early identification and classification of plant diseases[11], [12]. However, there are a lot of plant diseases that are not addressed in the current technologies. Thus, we need to extend the work to address more diseases and plants.

Thus, this work is focused on developing an automated disease detection model that can help farmers to detect disease in its early stage with better accuracy. In literature, many works are conducted to detect different[7], [13], [14] diseases for bean crops.

Specifically, no method used to detect common bacterial blight disease from common beans yet. This approach can reduce expenses, and effort and is less time-consuming than makes the ways we detect a disease from crops easily. The following research questions are addressed in this study: -

1. How we can develop a CNN model that reduces vanish gradient problems and detect common bacterial blight disease of common bean?
2. What is the performance of the models for common bacterial blight disease detection?
3. How a real common bean dataset will be collected to accomplish this study?

1.2 Scope of the Study

This study focuses mainly on developing an automatic model for common beans to detect common bacterial blight disease. The study proposes to look at three major common bean-producing districts, those are Areka Agricultural Research Center, Haramaya University Agricultural Research

Center, and Bule Hora University Kerch Agricultural Research Center. The proposed approach only focuses on the leaf of the common bean because the disease is mainly visible on the leaf portion as well and the leaf is suitable to capture images easily. This study does not focus on the stem, root, and remaining parts of the plant. In this proposed work the disease that affects the common bean excluding common bacterial blight and the districts that are not mentioned above is not the topic of this research. Instead, it aims to develop a model for the classification of common bean disease, which is a common bacterial blight in those districts. This study will not further concentrate on many diseases of common bean due to getting the desired dataset being difficult. And also assessing all districts is impossible, due to time and financial limitations.

In this paper, a modern deep learning approach for the early detection of common bean is proposed. Firstly, the collection of healthy and diseased common bean images with the help of domain experts from different agricultural research centers is done.

2 Preliminary and related work

In this section, we presented the status of the common bacterial blight of the common bean crop, an overview of convolutional neural networks, and related work to have a clear understanding of the domain area through previous work.

2.1 Common Bacterial Blight of Common Bean

CBB[15] is caused by *Xanthomonas axonopodis* pv. *Phaseoli* and its fuscans variant *Xanthomonas fuscans* subsp. It is the major bottleneck in common bean production in the world as well as in Ethiopia[16]. CBB is proven as the major concern in many of the world's common bean production regions including Ethiopia. CBB was reported in many parts of Ethiopia and is the reason for the reduction of yield by 22.4% in the eastern region of the nation. It enclosed 98.32 thousand ha of land with a productivity of 1.62 t ha⁻¹ of grain yield in Southern Ethiopia. However, productivity is even under the national average of 1.69-ton ha⁻¹. According to[17] CBB is a severe bacterial disease that affects the leaves, seeds, stems, and pods of a common bean. In addition to lowering production yields by up to 45% or more in susceptible cultivars, this disease degrades seed quality. The challenges behind CBB disease are that It is hard to manage due to the seed has borne the nature of bacteria and its capability to harvest huge amounts of second inoculum. Currently, it is found in several parts of Ethiopia causing the common bean to any stages and reducing the production of crops. Authors in[18] proved that the CBB caused important yield loss variables from 20 to 100% for the common

bean crop in Ethiopia. Since the CBB is causing economic losses of yields on common bean crops, the necessity of developing and using an effective and appropriate management system is unquestionable for sustainable production and economic gains in the nation.

- A modern and vanishing gradient problem-reduced CNN model with few parameters was developed to detect and classify common bean disease (CBB) using the images collected from the agriculture research centers.
- Another important contribution of this study is collecting and preparing well-organized and managed real data(dataset) of common bean from the field with the help of domain experts.
- To the end, we trained two pre-trained CNN architectures and performed a comparison with our proposed model.

bean crop in Ethiopia. Since the CBB is causing economic losses of yields on common bean crops, the necessity of developing and using an effective and appropriate management system is unquestionable for sustainable production and economic gains in the nation.

Thus, Treating the common bean disease in its early stages can increase the quality and quantity of crop products. Recently many researchers from the agriculture field of study in Ethiopia tried to overcome the impact of this disease on the common bean using many methods like using[19], [20] most preferred seeds, detecting disease by observation, and also using drugs or chemicals to prevent disease. This is a very tedious, time-consuming, and costly technique that needs many experts as well as more effort[7]. Besides, it is not easy to control and handle the crops in a huge area. In different countries, an enormous number of farmers lack the facilities and experience to deal with crop diseases and it is difficult to get the experts easily. Using deep learning algorithms to detect a disease from leaves helps farmers to detect disease at an early stage of production and also gives desired decision at right time for treatment to get quality and high productivity.

And also reduce cost, effort, time and makes the ways we detect a disease from crop easily. This algorithm is used commonly for image data to detect a disease from leaves for crops[7], [8], [21] like beans which conduct both feature extraction and classification in union with little requirement of human effort, that require to form a learning (computer vision) model that look like the brain of humans.



Fig. 1 Example of common bean crop healthy (left) and diseased (right).

2.2 Convolutional Neural Network

CNN is the most frequently used deep learning network and highly applied algorithm in a range of different fields[22]. CNN has outperformed issues with computer vision. It is the type of feed-forward neural network that is mostly used to analyze visual images by processing data with a grid-like topology. ConvNets is the name of CNN. ConvNets has demonstrated outstanding performance across a wide range of applications, including medical image analysis, speech recognition, object identification, picture classification, and natural language processing[23]. The fundamental advantage of CNN over its forerunners is that it performs automatically identifying the relevant features without human intervention. The CNN's three primary benefits were parameter sharing, sparse interactions, and equivalent representations. cited by authors in [24]. Unlike typical fully connected (FC) networks, CNN fully utilizes 2D input-data structures, including visual signals, by using shared weights and local connections. This method uses a relatively small number of parameters, which speeds up the network and makes training simpler. The number of free parameters in the network dramatically grows as the size of the image increases. As a result, as the model grows, the network's performance degrades and overfitting results[25]. A difficulty with machine learning methods is overfitting, which occurs when the network size is enormous and there isn't enough data to fit the model. The issue reduces the machine learning model's capacity for generalization[26]. CNN resolves this problem by providing layers with neurons arranged in three dimensions (3D): width, height, and depth. Each layer of CNN receives a 3D volume of input data, in this case an image, and

outputs another 3D volume of data using a differentiable function[27]. Height, width, and depth, or $m \times m \times r$, are the three dimensions in which the input x for each layer of a CNN model is arranged, with height (m) being equal to width. The channel number is another name for depth. An RGB image, for example, has a depth (r) value of three. The letter k denotes the available kernels (filters) for each convolutional layer and has three dimensions, like the input images ($n \times n \times q$). But in this instance, n has to be less than m , and q has to be smaller than or equal to r . The local connections are also based on the kernels; they convolve input with k feature maps h_k , each of size $(m-n-1)$, and generate them with identical parameters (weight W^k and bias b^k). In a manner similar to natural language processing (NLP), the convolution layer produces a dot product between its input and the weights as represented in Eq(1), but the inputs are undersized regions of the initial image size. Next, by using the activation function for the convolutional layer or nonlinearity, we arrive at the following equation.

$$h^k = f(W^k * x + b^k) \quad (1)$$

The next step is to down-sample each feature map in the sub-sampling layers. As a result, the network parameters are reduced, the training process is accelerated, and the overfitting problem may then be handled. In all feature maps, the surrounding area of size $p \times p$, where p is the kernel size, is exposed to the function of pooling (such as max or average). The FC levels, which are comparable to the last layers in a typical neural network, give the high-level abstraction after obtaining the mid- and low-level information. The final layer, which generates the classification scores, may use support vector machines (SVMs) or SoftMax as examples. Every score represents the likelihood of a particular class for a given instance[22]

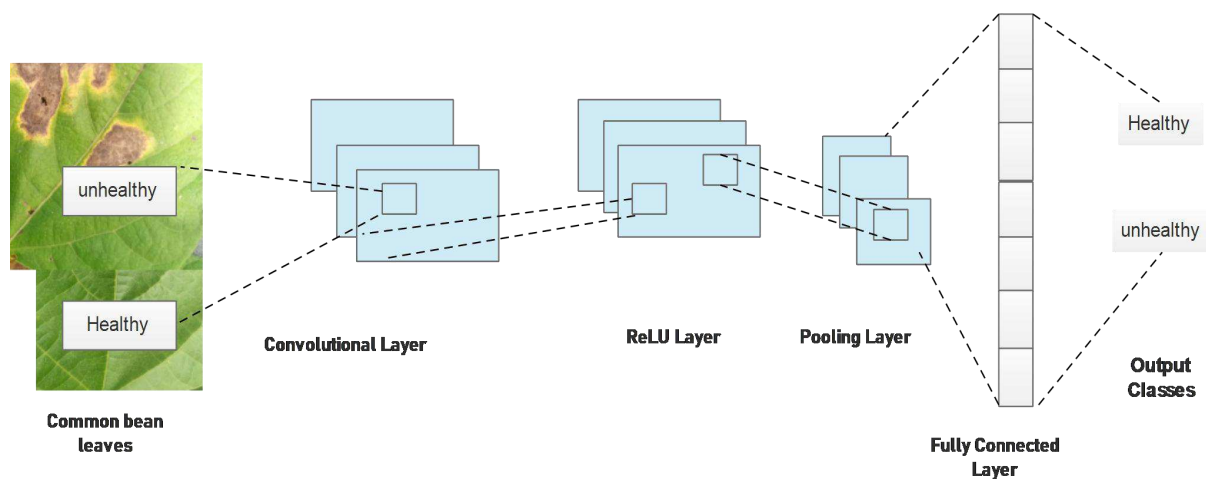


Fig. 2 Representation of CNN Architecture with common bean leaf.

The following are a few benefits of using CNNs in computer vision in contrast to other conventional neural networks:

1. CNN's weight-sharing function reduces the amount of trainable network parameters and therefore aids in improving generalization and preventing overfitting, which is the primary justification for using it.
2. When the feature extraction layers are used, the performance of the model is highly ordered and highly dependent on the extracted features, and the classification layer is simultaneously learned[22].

2.3 Related work

Recently, in the agricultural sector, plant diseases have become a major factor that caused the reduction of productivity and loss of yield. To minimize these losses number of researchers have done several works to detect disease at an early stage by identifying the infected leaf automatically[14], [28], [29]. There are vast works done for the identification of plant diseases using image processing, machine learning approach, and deep learning approach.

Authors in[30] have proposed an integrated deep learning approach for detecting and classifying banana diseases such as a banana bunchy top virus (BBTV), banana *Xanthomonas* wilt(BXW), banana black Sigatoka (BBS) and banana *Fusarium* wilt (BFW). They compared the CNN architecture, which is based on the LeNet-5 architecture with the FSVM, which combines binary and multiclass SVM. Approximately 3,500 images of both healthy and unhealthy banana plants were included in the datasets utilized in this investigation. The result of the study shows the performance of their proposed approach achieves an overall accuracy of 99% in detecting the disease of bananas. In the end, they concluded that deep learning algorithms can handle both structured and unstructured data and they were ultimately shown

to be the most effective option for detecting banana disease.

A deep CNN algorithm in[27] developed a CNN model to identify bacterial wilt disease from the onset based on healthy and unhealthy leaf images of the crop, which is the most common crop in Ethiopia. They have used CNN architecture that is modified from VGG19 and performed a comparison with transfer learning models like VGG16 and InceptionV3. In this study, a dataset of around 4896 images of onset crops was gathered from the agriculture field by using a digital camera. The authors used an expert to annotate manually the dataset into two classes; namely healthy and unhealthy. Finally, three CNN architecture models are trained such as VGG16, InceptionV3, and the proposed model (modified VGG19) in the study, which has been performed with 98.5% accuracy for the detection of onset crop disease.

A novel 14-layered deep CNN [31] is proposed to detect plant disease using leaf images. They have used three image augmentation methods to generate datasets such as Neural style transfer, neural convolutional generative adversarial network, and simple image manipulation. They used around 147,500 images of 58 different diseased and healthy plant leaf classes. Based on their finding result, the proposed DCNN model achieved a better accuracy of 99.9655% than transfer learning algorithms. However, the authors did not use the actual images taken in the fields and sometimes data argumentation leads to high accuracy and causes overfitting issues unless we perform necessary experiments to avoid overfitting.

The author in[32] presented deep learning techniques using modified AlexNet CNN architecture for the detection and classification of Tomato leaves. Open-source data from the Kaggle repository was used in this study to train and evaluate the model. The total

number of images in the datasets used by the authors after augmentation and transformation is 30,880 images. A total of 18,345 images were used for training data and 4585 images of test data were collected for identification and classification of tomato disease with ten (10) classes such as Late blight, Mosaic Virus, Septoria leaf spot, Spider mites, Target spot, Bacterial spot, Leaves mold(fungus), Early blight(scar), Yellow Leaf Curl Virus and healthy. To end, the proposed model detected and classified 10 different classes of tomato disease with a better accuracy of 98%.

As mentioned[33][34] in a number of studies CNNs are highly suitable for the automated detection and diagnosis of various plant diseases through the investigation of simple leaves of plants. Authors in[35] presented a specialized CNN model that depends on the specific CNN architecture for the identification of plant disease through simple plant leaves. In this paper, the authors used an openly available dataset of 87,848 total images taken in both real conditions and laboratory conditions in cultivation fields. The authors detect disease for 25 plants with 58 classes of disease. Several amount of CNN architecture models are trained such as GoogLeNet, AlexNet, VGG, and Over feat to detect disease on a plant leaf. Finally, the VGG has achieved better performance in the identification of plant diseases.

3 Research Materials and Methods

This section presented data description, data preprocessing, data augmentation, Overview of the proposed model, Components of the proposed model,

Authors in[36] presented an automatic discriminative method based on CNNs for categorizing 12 different cultivars of common beans that belong to three species. Authors have used advanced loss functions, such as Large Margin Cosine Loss and Additive Angular Margin Loss. They collected images with a digital camera from Arak University and evaluated the models by classifying them into three species (I, II, III) based on images from the leaf backside and foreside. Finally, the mean classification accuracies of 95.86, 91.37 and 86.87% were achieved at levels I, II, and III, respectively.

A deep learning-based approach in[37] is proposed for the identification of diseases of Maize crops using images leaf. In this study, a total of around 5939 images of maize crops were collected from experimental fields of All India Coordinated Research Project on Maize centers of ICAR-Indian Institute of Maize Research. The authors considered three major diseases of maize crops namely Turcicum Leaf Blight, Banded Leaf and Sheath Blight, and Maydis Leaf Blight. So, the dataset images have three disease classes and one healthy class. They have trained three CNN architectures based on the InceptionV3 architecture. On a different test dataset, their proposed model got an overall classification accuracy of 95.99 %.

Feature extraction, and classification techniques of the proposed model in this study. In addition, we discussed the pre-trained model, and hyperparameter of the proposed model to improve the effectiveness of the proposed approach.

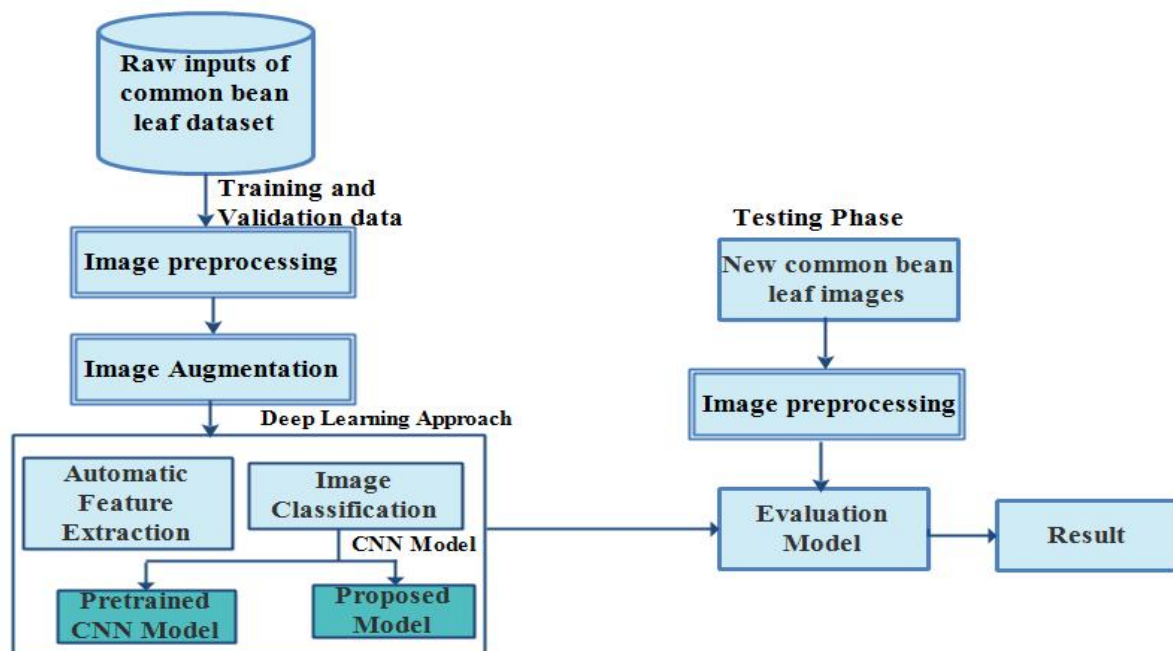


Fig. 3 Block Diagram of Detection of CBB disease

3.1 Data Description and Data Preprocessing

Data description

To get the right solution for the proposed problem, having an image of a common bean leaf is essential to detect the disease of common bean crops at an early age to make the right decision with the necessary treatment. However, to make the entire experiment more reliable, there is a scarcity of real images to fit the CNN models with a large dataset. Thus, in this research paper mainly the common bean data was collected from three agricultural Research Centers in Ethiopia. These three agricultural Research Centers are Areka Agricultural Research Center (Wolieta, SNNPR, Ethiopia), Bule Hora Kercha Agricultural Research Center (Oromia, Ethiopia), and Haramaya University, Agricultural Research Center (Oromia, Ethiopia). We have also downloaded some images from the internet to enlarge the dataset to examine the ability of the developed model to accurately detect the healthiness and disease from common bean leaves.

The images are taken with a digital camera from the agricultural center with the help of agricultural experts when everything is normal, without adjusting the light sensors or the images' respective positions. The deep neural networks take those images as input and manage the necessary conditions through pre-processing and automatic feature selection without difficulties. The collected images are annotated by agricultural experts from Areka Agricultural Research Center, Southern Agricultural Research Institute, and Haramaya Agricultural Center. Every image has been verified by an Associate Plant Pathology Researcher (Misgana Mitiku) at SAR. With the help of the domain experts, we have classified the images into two classes namely, Healthy and Bacterial blight disease. Even if there is another disease like Bean rust, Angular spot on common bean leaves rarely, In this study, we have concentrated on the bacterial blight disease since Ethiopia's common bean crop is primarily affected by it. The total of original images collected for this study was 3,135 images. This dataset consists of both healthy and unhealthy images.



Fig. 4 Sample of Collected Images

Data Preprocessing

Data preparation(pre-processing) is a common task that researchers deal with at the first step in deep learning techniques to prepare the raw data (images collected from agricultural centers) in the format that the network can accept as input. But in the deep learning approach since algorithms can take raw images, there is no explicit preparation of the data, and extract the feature from it automatically. The gathered image data come in various sizes. Nonetheless, the collected images' dimensions must correspond to the size of an image input layer used by CNN methods. Thus, we have to perform normalization(resize) images with required pixels to enhance desired features or to reduce factor(bias) that affects our classification models. For instance, we have resized collected images to 150 x 150 pixels for the developed model, background noise was avoided from all images, some images were cropped manually, sequential renaming, data balancing for two classes,

and other important preprocessing tasks were managed before training our model.

Data augmentation

Today, many of CNN's algorithms in image classification are so effective, it requires a large amount of data that may be difficult to provide. The most common problem with CNN's algorithm is the lack of enough amount of training datasets. The method of dealing with this problem is called data augmentation[38]. It is a process of increasing the amount of training data in a dataset by generating more data from the existing training data. In addition, it helps to learn more complex features from the data, significantly improve the neural network accuracy, and conquer the overfitting issue[39]. Increasing the number of images through data augmentation makes it generally less vulnerable to adversarial attacks and more robust desired models for training. In this study, we have used ImageDataGenerator techniques from Keras libraries to generate enough images for the

proposed model using original images from agriculture centers. We have performed data augmentation using different parameters of ImageDataGenerator such as rotation range, width shift range, height_shift_range, zoom range and others before feeding the dataset into our models. Finally, amount of data was increased from our original images and improved the performance of proposed model.

Table 1. Augmentation Parameters

Augmentation Parameters	Augmentation Factor
Rotation Range	False
Width Shift Range	0.2
Height -Shift -Range	0.2
Zoom Range	0.2
Horizontal Flip	True
Vertical Flip	True

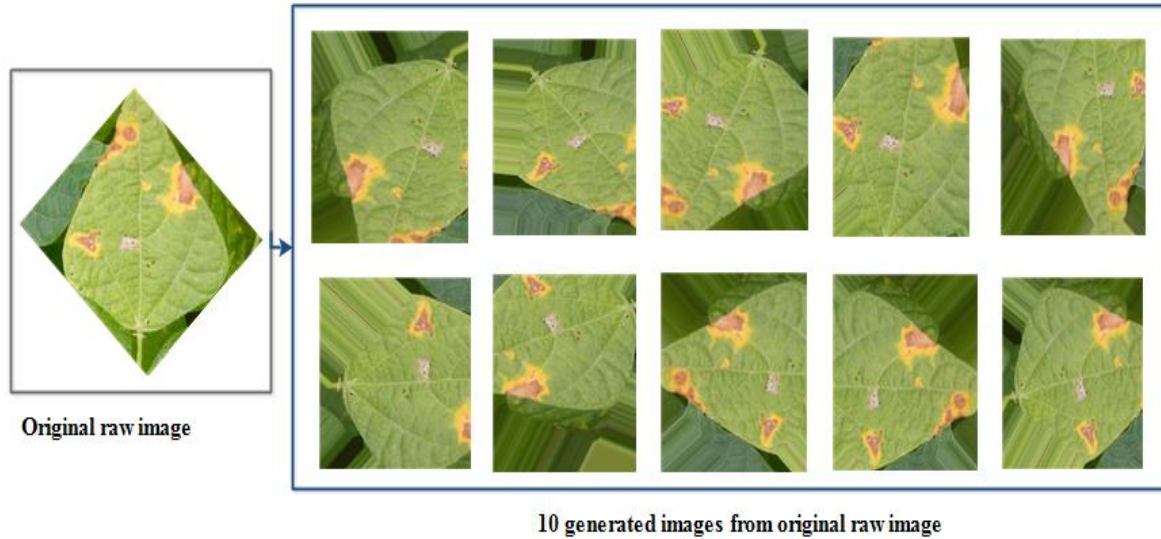


Fig. 5 Sample of Augmented Images

The total amount of generated images using augmented techniques is 75,240 for the training. Thus, we experimented using the augmented images for the proposed model. Here we presented a sample of augmented images with the original images in Fig. 5.

3.2 Proposed Model Architecture

The main aim of this study is to detect CBB disease from common bean images using CNN architecture. For this study, we used new CNN architecture, which is designed with an association of densely connected convolutional networks(DenseNet)[40] to classify and detect CBB disease on the common bean leaves. And also, performed a comparison with other pre-trained CNN algorithms. Optimized DenseNet-121 was competitive. Some main reasons behind using this algorithm: It reduces the number of parameters, enhances feature propagation (feature reuse), and lowers or mitigates the vanishing gradient issue. Thus, those points make the proposed model more efficient for training.

In the deep learning approach, the information (input data) passes through a lot of hidden layers between

the input layer and the output layer. Due to this long path between input and output layers, the information can vanish before reaching its destination. A vanishing gradient problem was addressed in the development of DenseNet, an image classification technique, which was designed to enhance model performance[41]. It has acquired state of art with feature reuse and fewer parameters. With some modification, unlike ResNet, DenseNet concatenates features using (.) attributes from all previous layers instead of combining the features using additive(+) attributes from the previous layers[42]. The DenseNet overcame the problem of vanishing gradient by providing high accuracy when contrasting it with other CNN architecture by connecting every layer directly to each other.

In the traditional feed-forward CNN, each layer receives a feature(output) from the previous layer except the first layer which takes in the input and produces a feature map(output) that is then passed into the next layer. Thus, L layers have L direct connection between layers, or one between each layer as well as the layer after that.

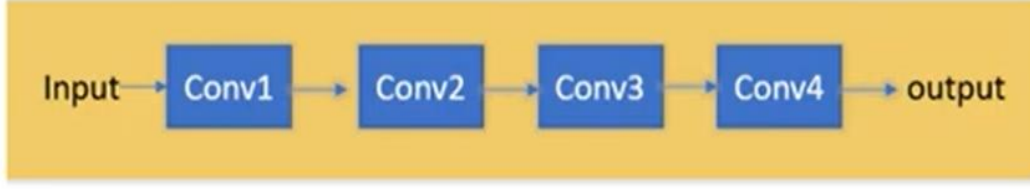


Fig. 6 Traditional Convolutional Neural Network

However, as the number of layer increase, the problem of vanishing gradient will happen. This is because the information path from the input to output layers increases. The last layer of the network's final feature map becomes convoluted and less useful, which lowers the network's training performance. DenseNet overcame this vanish gradient problem by modifying the standard CNN architecture and simplifying the connectivity between layers. This

architecture (DenseNet), collects features from all layers, and every layer is connected directly with each other. Thus, L layers have $L(L+1)/2$ direct connections. In DenseNet, for backward propagation after calculating the lost function(error) that lost function sends gradient information to every layer. Even to an earlier layer of the network, each layer has direct access to the gradient from the lost function.

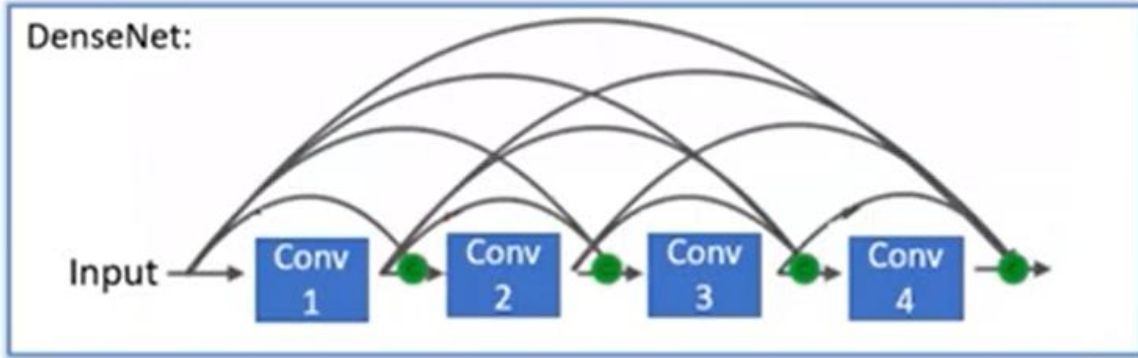


Fig. 7 Densely Connected Convolutional Networks

Generally, classical CNN architectures calculate the output layers (n^{th}) using a non-linear transformation represented by $H_n(\cdot)$ to the previous layer X_{n-1} .

$$X_n = H_n(X_{n-1}) \quad (2)$$

In DenseNet, the feature maps of all the previous layers are concatenated and used as inputs, not summed. Likewise, DenseNet requires fewer parameters and this allows for feature reuse. It offers a simple communication model for improving information flow between layers. So, the n^{th} layer receives the feature maps of all previous layers, $X_0, \dots, X_{n-1}$, as input. The equation is modified into:

$$X_n = H_n([X_0, X_1, X_2, \dots, X_{n-1}]) \quad (3)$$

Where $(X_0, X_1, X_2, \dots, X_{n-1})$ is a single tensor of the concatenation of the feature maps. $H_n(\cdot)$ represented a non-linear transformation. The output produced in all previous layers n $(0, 1, 2, \dots, n-1)$.

The multiple inputs of H_n are concatenated into a single tensor to simplify implementation. $H_n(\cdot)$ function consists of three main operations: activation function(ReLU), batch normalization(BN), and pooling and convolution(Conv). In addition, the growth rate k helps to increase the number of features and generalize the n^{th} layer with the following equation: $K_n = k_0 + K(n-1)$.

In this study, we designed the proposed model based on the DenseNet-121 architecture and optimized it with the necessary parameters. The proposed model has 1 basic convolution, 4 dense blocks, 3 transition layers, batch normalization, ReLU (activation function), and dropout are included in the both dense block and transition layer to avoid the overfitting issue and also the model has a single fully connected layer and the SoftMax activation function.

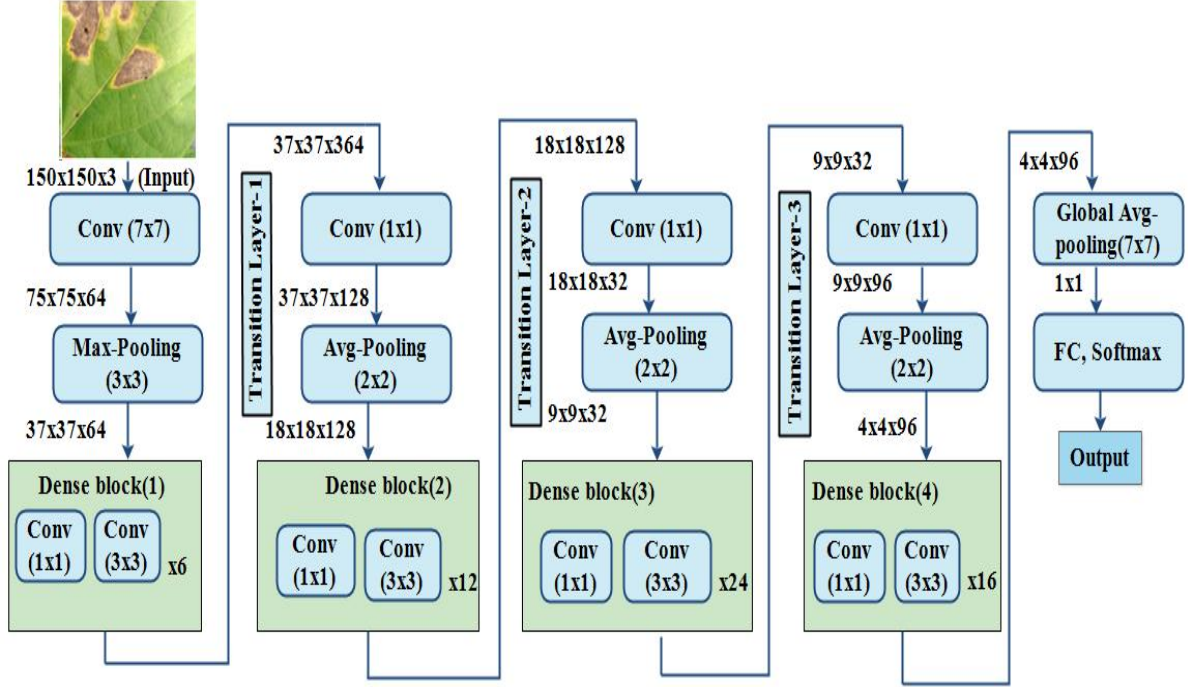


Fig. 8 Proposed Model Architecture

We can compute the spatial size of the output in each layer by using an initial input image size(W_1), Stride(S), Field Size(F), and Padding(P)[43].

where W_1 is the size of the input image, F is the filter size, S is the stride, and P is padding from Eq(4).

The initial input size of an image (W_1) is $150 \times 150 \times 3$ and changed after max-pooling operations, the initial size of the Stride(S) and Filter Size(F) are 2 and 7×7 respectively and these sizes are changed after pooling and dense block operations, and the initial size of zero padding(P) is 2 in the network for our proposed model. The proposed model's first convolutional layer uses 64 kernels with a size of 7×7 and a stride of 2 pixels to filter the $150 \times 150 \times 3$ input image. Given that this convolution layer output size is $(150 - 7 + 2(2)/2 + 1 = 75$, and since this layer has a depth of $K = 64$, the output volume of this layer is $75 \times 75 \times 64$. The total number of neurons is determined by the product of the output volume for the basic convolutional layer is 360,000. A region of size 7×7 in the input volume is connected to each of the $75 \times 75 \times 64$ neurons in this network. To determine how many parameters there are for the output volume. For instance, the basic convolution layer's input size is $75 \times 75 \times 64 = 360,000$ neurons, and each has $7 \times 7 \times 3 = 147$ weights and 1 bias. Then the number of parameters for the basic convolution layer is $360,000 \times 147 = 52,920,00$ parameters in the model's basic convolution layer. The concept of parameter sharing is used to take advantage of the traditional neural network and reduce the number of parameters in the proposed model.

We can get an exact output size of all layers using the following equation in the proposed model.

$$OutPut\ Size(W_2) = \frac{(W_1 - F + 2P)}{S} + 1 \quad (4)$$

The basic convolutional layer of the proposed model is followed by one max-pooling. With a filter of size 3×3 and stride 2, this max-pooling layer decreases the output of the basic convolutional layer to $37 \times 37 \times 64$. This layer just downsamples along the spatial dimension of the input volume and lacks learnable features, hence there are no parameters in these levels. Dense block enables the DenseNet architecture to down-sampling the layers to reduce the size of feature maps. Thus, DenseNet is divided into dense blocks to gain higher computation speeds. There are four dense blocks in the proposed model. Every dense block has a different number of layers(repetitions) featuring two convolutional; Conv 1×1 sized kernel as the bottleneck layer and conv 3×3 to perform the convolutional operation.

The layers between two dense blocks are called the transition layer which performs down-sampling of the size of the feature-map through convolution and average-pooling operations. In the proposed model there are three transition layers. Every transition layer has the same size convolution(1×1) and average-pooling(2×2) filter size operations to perform down-sampling of the feature maps. The global average-pooling layer with filter size 7×7 accepts all the feature maps of the network to perform image classification. The output layer is the last layer of the model and it has 1 fully connected layer (FC) with a SoftMax activation function.

Table 2 The details of the proposed model with parameters

Layers		Input Size	Filter Size		Window Size for Pooling	Stride	Padding	Output Size
Images(Input)		150x150x3	-		-	-	-	150x150x3
Convolution		150x150x3	7x7		-	2	2	75x75x64
Max-Pooling		75x75x64	-		3x3	2	0	37x37x64
Dense block(1)		37x37x64	1x1 3x3	x6	-	-	0	37x37x64
Transition Layer(1)	Convolution	37x37x64	1x1		-		-	37x37x128
	Average Pooling	37x37x128	-		2x2	2	0	18x18x128
Dense block(2)		18x18x128	1x1 3x3	X12	-	-	0	18x18x128
Transition Layer(2)	Convolution	18x18x128	1x1		-		-	18x18x32
	Average Pooling	18x18x32	-		2x2	2	0	9x9x32
Dense block(3)		9x9x32	1x1 3x3	x24	-	-	0	9x9x32
Transition Layer(3)	Convolution	9x9x32	1x1		-		-	9x9x96
	Average Pooling	9x9x96	-		2x2	2	0	4x4x96
Dense block(4)		4x4x96	1x1 3x3	X16	-	-	0	4x4x96
Global Average Pooling		4x4x96	-		7x7	-	0	1x1
Full Connected Layer		1000 neurons						
SoftMax		2 classes						

3.3 Feature Extraction for Proposed Model

Before the classification step of an image can begin, we must undertake an important activity called feature extraction. The CNN method extracts significant features that are required to identify the images. One feature parameter, a color feature, is employed for the detection of CBB. This feature is taken into consideration since, according to the current system, a human eye can tell by a difference in visual color whether a crop is infected with the

disease or not. One of the important tasks of deep learning is managing feature extraction automatically. Thus, this proposed model automatically extracts features from the common bean leaf. The algorithms learn the features to extract from the input image during training CNNs. In our case, as we can understand from Fig. 9, the Global Average Pool layer performs feature extraction and sends one column feature to the output layer

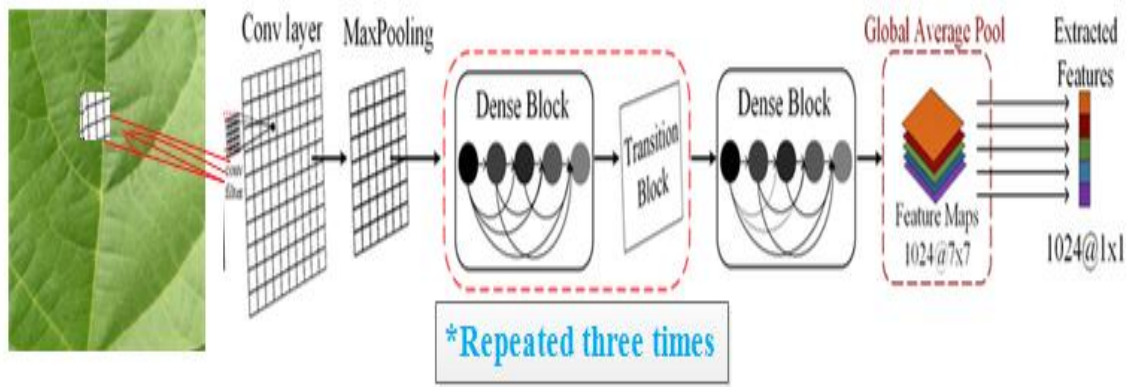


Fig. 9 Feature Extraction for Proposed

3.4 Image Classification for Proposed Model

As we have indicated in the proposed model diagram in **Fig. 8** total we have one fully connected layer including the output layer. Classification is carried out in layers that are fully connected. The main purpose of these layers is to classify the given image using the features that the Global Average pool layer collected into a single vector. The global-average

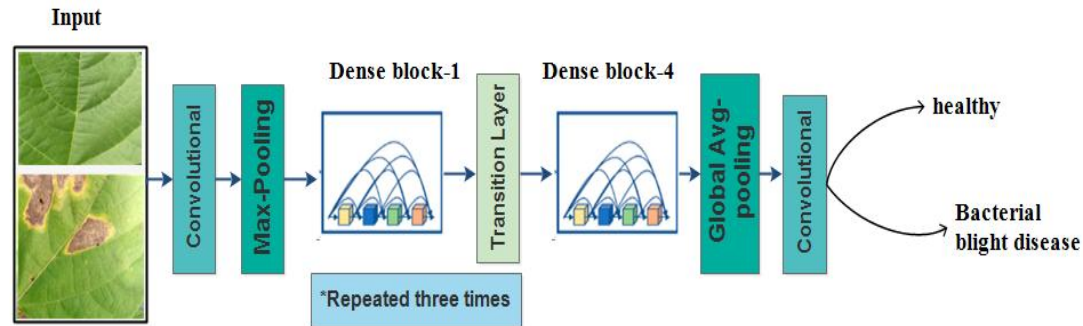


Fig. 10 Classification for Proposed Model

Every dense block has a different number of layers(repetitions) featuring two convolutional; Conv 1x1 sized kernel as the bottleneck layer and conv3x3 to perform the convolutional operation. The convolution layer in the dense block consists of the batch norm, ReLU, Conv3x3, and dropout in sequential order. Also, every transition layer has the same size convolution(1x1) and average-pooling(2x2) filter size operations to carry out the feature map downsampling. The transition layer consists of the batch norm, ReLU, Conv1x1, dropout, and average-pool 2x2 in sequential order.

pool layer accepts the output of the dense block (4). But before being fed into the fully connected layer, the outputs are combined and flattened into a single vector value (FC). Each element of the vector represents the likelihood that a particular feature (in the input images, color) belongs to a particular class. The **Fig. 10** illustrates how the input values flow into the FC layer of the network.

3.5 Classification Using Transfer Learning Method

Transfer learning is mostly an applied approach to extend knowledge already learned to solve a certain problem in a different field and to apply for to solve a new problem in the other related field. As part of a large-scale image classification task with thousands of classes, many CNN architectures are often trained using an extremely large number of images[44]. Due to its extensive training on millions of images and thousands of classes, this architecture offers a higher capability for object generalization. As such, the learned feature with CNN architecture is applied to many more real-world scenarios, even though the

challenges are different from those of the original effort[22]. These models are known as pre-trained models, and transfer learning is the process of both training the models and using them. These architectures must be trained from scratch, which takes a lot of time and effort. In this study, in addition to our proposed model, pre-trained models are assessed for the detection of CBB disease using

the common bean leaves dataset collected for this study. Due to the small size of our common bean leaves dataset, we have opted to employ two pre-trained models, InceptionV3 and VGG16, which were learned on a massive scale of the ImageNet dataset and transferred into our current study, rather than training each model from scratch. Finally, we have performed a comparison with the proposed model.

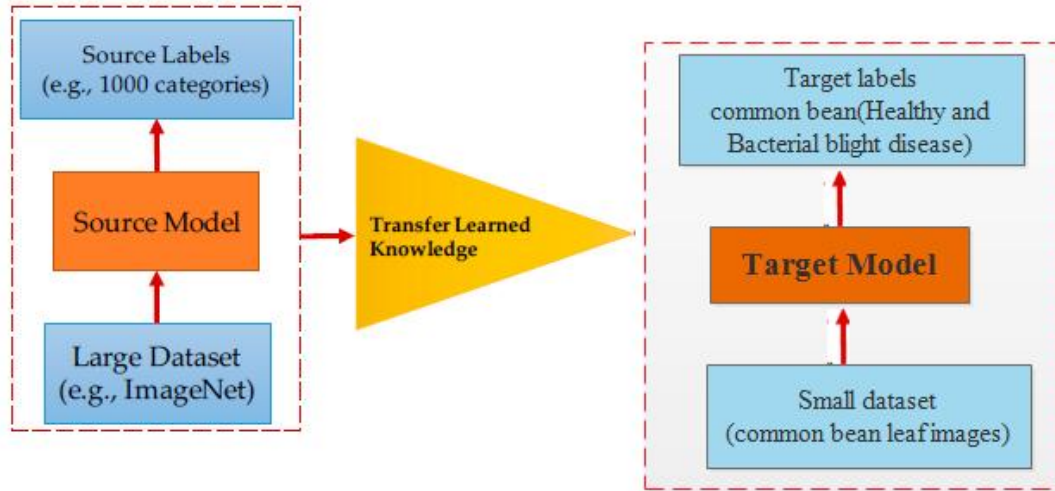


Fig. 11 Concept of transfer learning for this study

4 Experimental Result

In this section, several comprehensive experiments are conducted to implement and demonstrate the effectiveness of the proposed approach for common bean image classification. Firstly, a brief demonstration of the common bean dataset, data pre-processing, and data augmentation is done. Next, two different classification scenarios are undertaken during the experiment-based color features to image simply as diseased or healthy. In the first scenario using the proposed model, the image classification was done and the performance of the model was evaluated. In the second incident, two well-known, previously trained models mentioned in section 3.5 the image classification were conducted, and the performance of the model was evaluated. Finally, an extensive comparison between the proposed model and pre-trained CNNs architecture was undertaken and the proposed model performed better for the common bean image classification.

4.1 Implementing the Proposed Model Architecture

Before we start the implementation of the proposed approach, installing the necessary development environment has been undertaken. Firstly, we

installed the latest version of Anaconda3 and created the 'TensorFlow' environment variable in the Anaconda Navigator and then installed TensorFlow version 2.9 and Keras version in the created environment variable with all necessary libraries. The implementation of the proposed model of classification common bean leaves disease was executed with a Python programming language and used a local machine (Dell Desktop) with 16GB of RAM, 1TB hard drives, and Intel(R) Core (TM) i7-9700,3GHz CPU on the window 10 operating system.

In this study, our proposed CNN model is based on the DenseNet121 design using limited resources and the model performed better for the detection of CBB disease. As we discussed in section 3.2, the model has 1 basic convolution, 1 Max-pooling, 4 dense blocks, 3 transition layers, Average pooling(Global), and 1 fully connected layer with a SoftMax activation function. The accepts image input size of 150x150 color images and gives an output of 2 classes for this study. To train our model: augmentation techniques are applied to the training data from our original dataset, and 75,240 total amount of augmented images is used. We performed experiments using a different ratio of training, validation data, and activation function, and by using different learning rates.

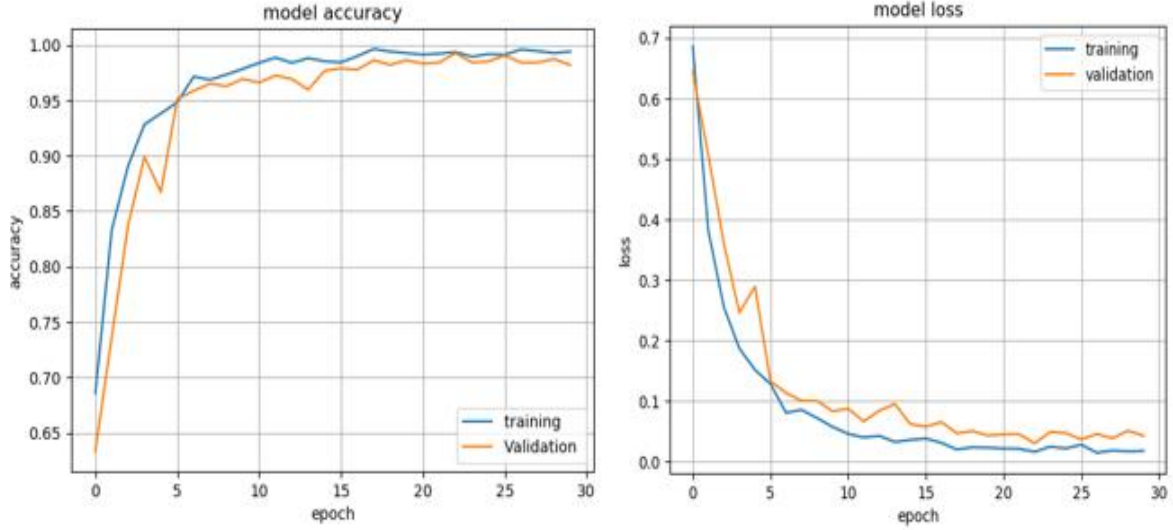


Fig. 12 Plots of the proposed model on train and validation datasets.

Our model performed classification of the given common bean images with an outstanding result of training accuracy of 99.4% and validation accuracy of 98.2% by using batch size 32, a learning rate of 0.001, CCE loss function, output layer activation function of SoftMax, and dataset ratio of 80:20 training and testing data respectively.

Fig. 12 illustrates the training and validation accuracy and loss curve of the proposed model. The training accuracy line's value increases to approximately 68% at the beginning of the training, but the validation accuracy line's value is less than 60%. Up till epoch 10, the values of both accuracies gradually rise, and at that point, training and validation begin to synchronize. At epochs 17 up to 20, both lines are kept on the same value and then both accuracies pass 97% and keep growing very slowly. Slowly the curve of both training and validation loss decreases linearly from 0.7 up to 0.01 starting from the first epoch to the end. Starting from the value of loss 0.1, and the validation loss became less than the training loss up to epoch 30. Which indicates our model is free from overfitting problems.

In conclusion, we can say that training and validation accuracy both continued to rise in the same manner while training and validation loss continued to decline. Similar to how the training loss and validation loss curves are almost linear, so are the training accuracy and validation accuracy curves. Because the training accuracy and training loss are rising and falling, respectively, as well as for the validation accuracy and loss, the plots show that the model is not overfitting. The differences in training accuracy, validation accuracy, and loss are very slight. As a result, we conclude that the model has more generalization power.

Table 3 Summary of Training and Validation accuracies and losses

Metrics	Training	Validation
Accuracy	99.4%	98.2%
Loss	0.017	0.042

For the inference, we assessed our model using new testing images as follows:

```
plt.figure(figsize=(15, 15))
for images, labels in test_generator:
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i])

        predicted_class, confidence = predict(model, images[i])
        actual_class = class_names[int(labels[i])]

        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence: {confidence}%")

    plt.axis("off")
    break
```

Fig. 13 The Sample Code of Inference for Proposed Model

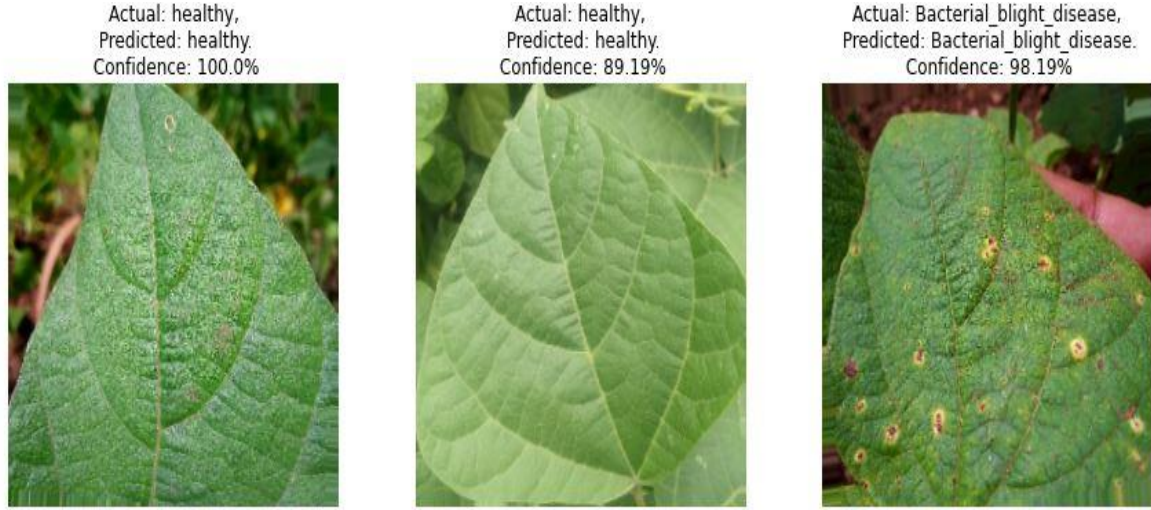


Fig. 14 Sample of truly Classified result of Proposed Model.

4.2 Performance Evaluation for Proposed Model

Since this study's type of problem is a classification problem, we have employed several classification Metrics for evaluating performance like Confusion Matrix, Classification Accuracy, and Classification Report. True Negative, False Positive, False Negative, and True Positive are the four values in the confusion matrix. In each of the Confusion Matrix illustrations, the blocks display the correctly predicted negative in True Negative, incorrectly forecasted positive in False Positive, incorrectly predicted negative in False Negative, and correctly predicted positive in True Positive, respectively. Inaccurately predicted values are represented by white blocks, and correctly predicted values are represented by blue blocks. Our model's performance is then evaluated by computing its Accuracy, Precision, Recall, Specificity, and F1 Score using these values.

In the confusion matrix, the columns correspond to the anticipated labels and the rows to the true labels. The Confusion matrix is shown in the Fig. 15.

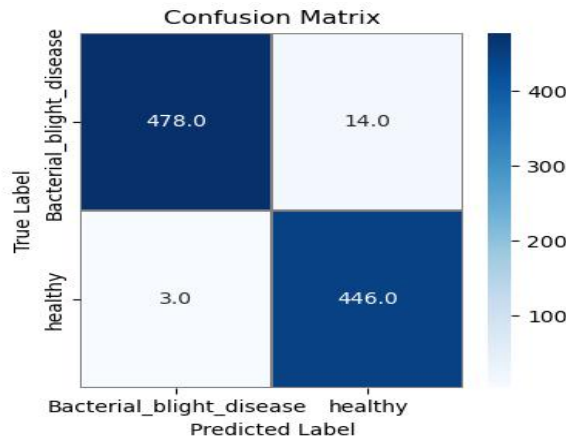


Fig. 15 Confusion Matrix of the model.

In addition to overall accuracy, we also performed the accuracy, precision, recall, specificity, and F1 score for our model. Accuracy is determined by the number of correct predictions made by the model through the whole number of predictions made in classification problems. Our model scored an accuracy of 98.1%, indicating that almost the same with overall accuracies. Scored a Precision of 98.1%, which is an evaluation of the percentage of common bean leaf that was predicted as having CBB disease, actually had a CBB. Scored Recall of 98.1%, which is an evaluation of the percentage of common bean leaves that was predicted to have CBB disease among those common bean leaves that truly had a CBB. Specificity is the exact opposite of Recall. It is a measure of the proportion of common bean leaves that did not have a CBB disease, which was predicted by our model as a healthy class. Scored F1 Score of 98.2%, is the single score that denotes both recall and precision (mean of the Recall and Precision that applied to test for Accuracy).

In order to calculate performance metrics and evaluate the result of our model the following four equations are used.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1\ Score = 2 * \frac{Recall*Precision}{Recall+Precision} \quad (8)$$

Table 4 Result from the confusion matrix for each class of common bean leaf

class	Precision	recall	F1 Score
CBB	0.99	0.972	0.982
Healthy	0.972	0.99	0.982
Accuracy			0.981
Macro avg	0.981	0.981	0.982
Weighted avg	0.981	0.981	0.982

4.3 Implementing VGG16 model

pre-trained (VGG16) model was also applied in addition to the proposed model for the detection of common bean disease. This model has 13 convolution

layers, 5 Max-pooling, and 3 FC layers with a SoftMax activation function. It is categorized by its simplicity by using only 3×3 convolution layers which are stacked on each other, and then the depth layer increases. The VGG16 has 16 weight layers in the network. The default input size for this model is 224 × 224 RGB images and produces 1000 classes of ImageNet dataset[22]. The input sent to this model is shadowed by the ReLU activation function and passes through stacked convolution layers with a 3×3 amenable field. This model uses 1 stride and 1 spatial padding for entirely of the 3×3 convolutions. Max-pooling of window size 2×2 with 1 stride is there next to very 3 consecutive convolution layers to perform down-sampling operation along the spatial dimension of the input volume of the result of the convolution layers. To end, this model has 3 FC layers for the classification, which are followed by a stack of convolutional layers.

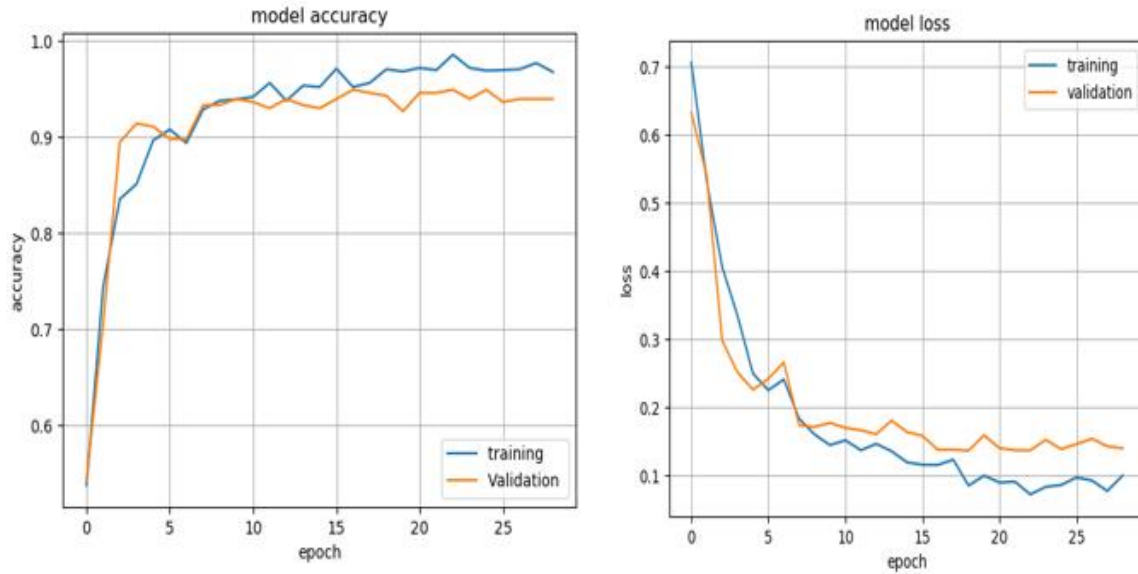


Fig. 16 Plots of VGG16 model on train and validation datasets.

We used only the convolution basis of the approach to fine-tune the model for the two classes' output in the common bean dataset. In our implementation, this model accepted the resized RGB images of the input size 150×150 with 138 million parameters. Overall, the VGG16 model is large and requires a sizable dataset. Since our model reduces the spatial dimension of the images, we train 7,342,593 parameters out of a total of 14,977,857 parameters. The remaining 7,635,264 parameters are not trainable. Thus, only some parts of the algorithm were trained. We have performed several experiments to get optimal values by using different hyperparameters using the common bean dataset. We updated some of the weights and increased the amounts of our data using augmentation techniques using different

parameters to prevent the Overfitting problem that arises from the VGG16 weights being trained with a huge amount of images, whereas our original data contained only 3135 images. Finally, this model was trained with a total of 75,240 images. Then we gained an accuracy of 96.7% and 94% for training and testing respectively as a result of our experiment using the optimal value of different hyperparameters.

The curves in **Fig. 16** illustrate the classification accuracy of training and validation, training loss, and validation of VGG16(pre-trained) with trained with epochs of 30 using the common bean dataset. In our implementation, VGG16 also performs well by updating some hyperparameters from the original pre-trained model for the classification of common bean crop leaves as diseased and healthy. As we can

understand from the curves of the plot the training accuracy in the first epoch is around 53.8% and slowly increases and reaches 96.7% at the 30 epochs.

Training loss also decreases slowly as the training accuracy increases with validation accuracy. Finally, as we can understand from the above curves in the plots the training accuracy curve is almost relative to the validation accuracy curve. Likewise, the training loss curve is also relative to the validation loss. This indicates that the VGG16 model is not overfitting using the common bean dataset. On another hand, the training loss is decreasing and the training accuracy is increasing.

Training loss also decreases slowly as the training accuracy increases with validation accuracy. Finally, as we can understand from the above curves in the plots the training accuracy curve is almost relative to the validation accuracy curve. Likewise, the training loss curve is also relative to the validation loss. This indicates that the VGG16 model is not overfitting using the common bean dataset. On another hand, the training loss is decreasing and the training accuracy is increasing. Finally, the VGG16 model performed classification of the given common bean image with a result of training accuracy of 96.7% and validation accuracy of 94% by using batch size 32, a learning

rate of 0.001, BCE loss function, output layer activation function of Sigmoid, and dataset ratio of 80:20 for training and testing data respectively.

4.4 Implementing InceptionV3 model

InceptionV3 is another pre-trained model we used for the common bean disease detection with the above two CNN architectures. It is another efficient CNN architecture for image or object detection proposed by Google and named another name as GoogLeNet [46]. This architecture is introduced with a large number of layers and neurons in each layer of the algorithms with less computational power. The default input size of this model is $299 \times 299 \times 3$ images and produces an output of 100 classes. This model has a total layer of 42, but it is computationally faster than VGG16 as we can understand them while we conduct our experiments. This model accepted the resized RGB images of input size 150×150 and we transferred learning with only the convolutional base of the algorithms to the model for the two classes output in the common bean dataset. Finally, this model was trained by using a total of 75,240 images with the same amount of data as the proposed model. Then we gained accuracy of 90% and 88% for training and validation respectively as a result of our experiment using the optimal value of different hyperparameters.

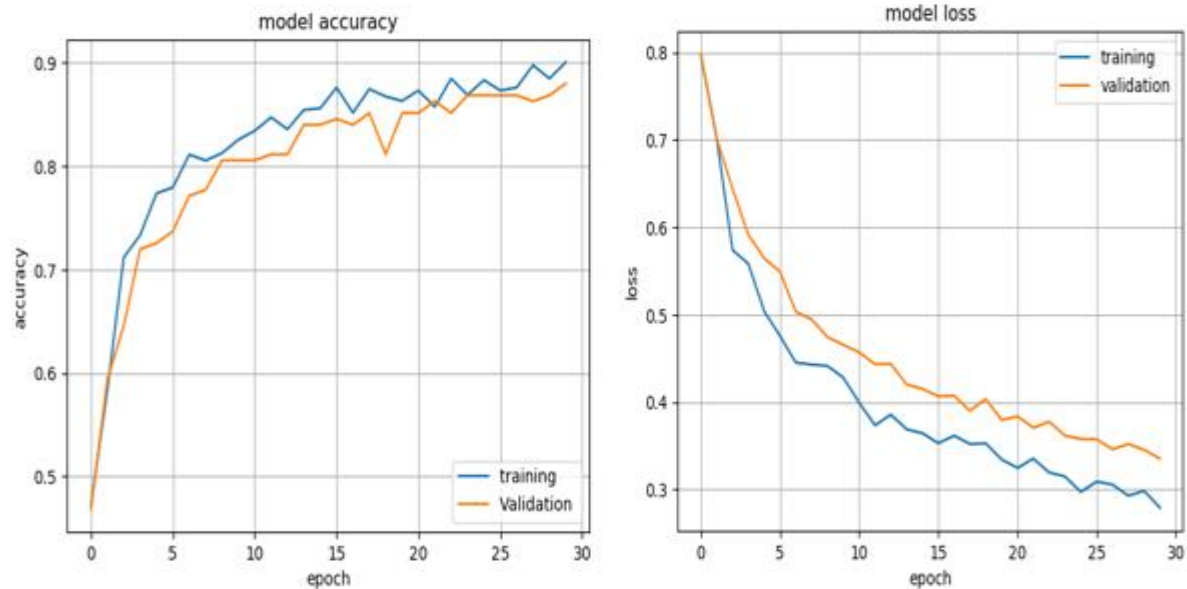


Fig. 17 Plots of the proposed model on train and validation datasets.

The curves in **Fig. 17** illustrate the classification accuracy and loss from the experiments for the InceptionV3(pre-trained) with epochs of 30 by using the common bean dataset. In our implementation, InceptionV3 also performs well in the classification of common bean crop leaves as diseased and healthy. As we can understand from the curves of the plot both accuracies increase from epoch 10 up to 25 and

training accuracy in the first epoch is around 47.7% and slowly increases and reaches 90% at the 30 epochs. Training loss also decreases slowly as the training accuracy increases with validation accuracy. Finally, as we can understand from the above curves in the plots the training accuracy curve is almost relative to the validation accuracy curve. Likewise, the training loss curve is also relative to the validation

loss. This indicates that the InceptionV3 model is not overfitting using the common bean dataset. On the other hand, the training loss is decreasing and the training accuracy is increasing.

Finally, the InceptionV3 Pre-trained model performed classification of the given common bean image with a result of training accuracy of 90% and test accuracy of 88% by using batch size 32, a learning rate of 0.001, BCE loss function, output layer activation function of SoftMax, and dataset ratio of 80:20 training and testing data.

4.5 Comparison of the proposed model with Transfer Learning

We have performed experiments using three different CNN architectures: the proposed model and the two pre-trained models using the collected common bean leaf dataset. During our experiments, we used the augmented images (dataset) from the original dataset that was collected from the fields with the help of experts from the domain. 75,240 total number of images used with 150x150 input size for the all model we trained for this study. In order to get a better performance classification model for common bean crops, we applied different scenarios like using different ratios of data split, different learning rates, different batch sizes, different activation functions, different epochs, and different input sizes. Then we performed classification using the optimal value of hyperparameters. All of the models are evaluated by new test dataset (unseen) data during the training of the model. Then our models performed well for the

common bean leaf dataset. In order to measure the performance of our models we used classification problem metrics.

In our experiment, the proposed model gave 99.4% accuracy of training and a validation accuracy of 98.2%. Likewise, the model gave a training loss of 0.017 and a validation loss of 0.042. This model was performed well for each classification metric to detect CBB disease on the common bean leaf crop. Based on our experimental result, this model scores high accuracy of classification for early detection of CBB disease on common bean leaf images to take the right action to treat crops in the early stage. In this study, next to the proposed model, the VGG16 pre-trained model performed well with 96.7% accuracy of training and a validation accuracy of 94%. In the same way, the model gave a training loss of 0.0995 and a validation loss of 0.139. This model also scores high accuracy of classification for early detection of CBB disease on common bean leaf images next to the proposed model. The last model(pre-trained) we used to detect CBB disease on common bean leaves was InceptionV3. The InceptionV3 pre-trained model also performed well but slightly less than the proposed model and VGG16 pre-trained with a training accuracy of 90% and validation accuracy of 88.1%. In the same way, InceptionV3 gave a high training loss of 0.279 and validation loss of 0.335 compared to the above two models. Lastly, we performed a comparative analysis of the proposed model with the two pre-trained models, and the proposed model has gained high classification accuracy.

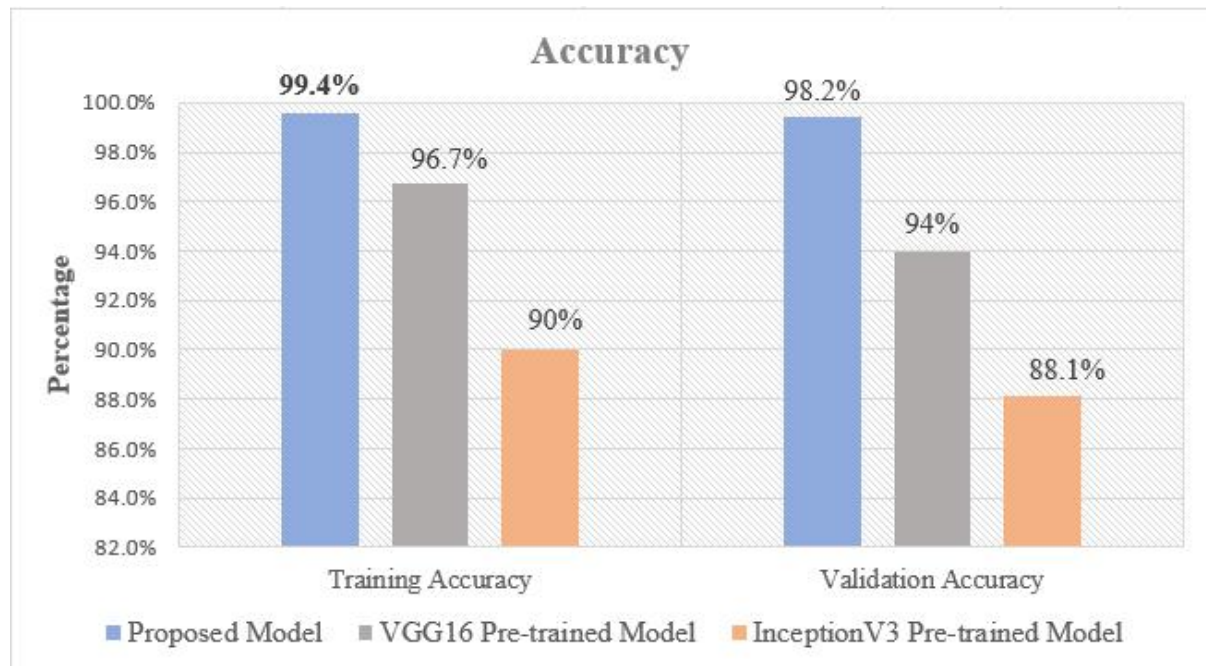


Fig. 18 Comparison Accuracies of Three Models

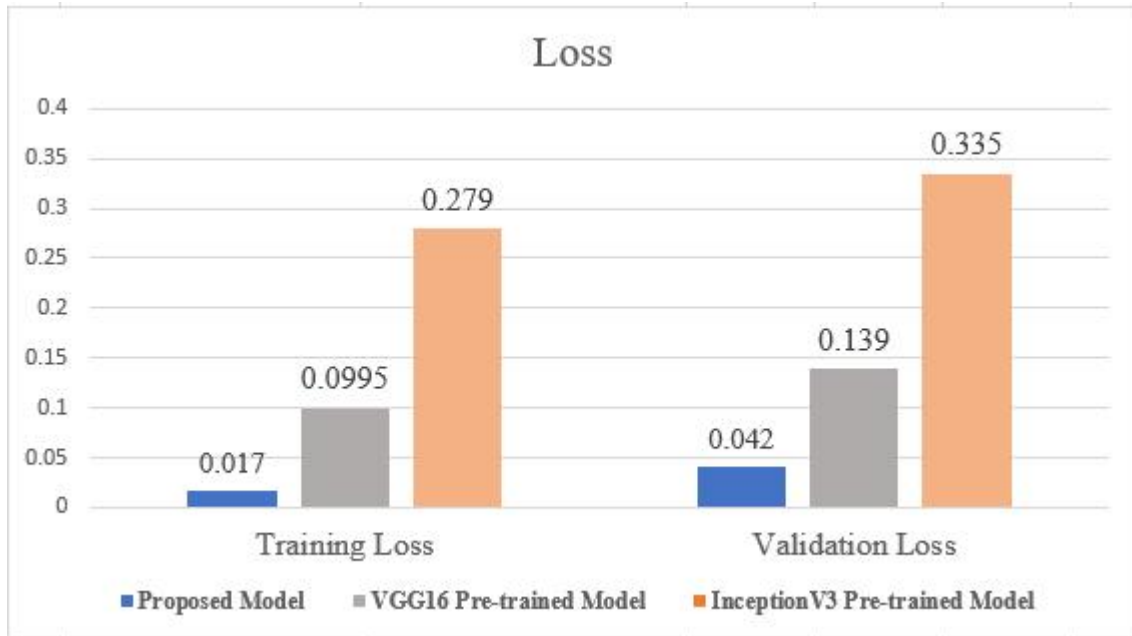


Fig. 19 Comparison Losses of Three Models

As we illustrate in Fig. 18, our models performed well in our experiments. The test accuracy of the proposed model is 99.2%, the VGG16 pre-trained model is 94%, and the test accuracy of the InceptionV3 pre-trained model is 88%. The test result shows that the proposed model has classified the input image as diseased or healthy with better performance than the VGG16 and InceptionV3 pre-trained models.

The loss of three models are presented in Fig. 19 above and the value of our model is less than the two pre-trained models. The validation loss of our model is 0.042, the validation loss of VGG16 is 0.139, and the validation loss of InceptionV3 is 0.335 which is high. Therefore, the proposed model performed better in our experiment.

5 Discussion and Analysis

The secret behind the proposed model to give a better result for the classification of common bean leaf images is that the proposed model reduces the vanishing gradient problem. Reduces the number of parameters with feature reuse by collecting features from all layers, and every layer is connected directly with each other during training. In addition, we have used different hyperparameters and fine-tuned them during the experiment to find an optimal value. In traditional CNN architecture, the information (input data) passes through a lot of hidden layers between the input layer and the output layer. Due to this long path between input and output layers, the information can vanish before reaching its destination. The proposed modern CNN architecture avoids the vanishing gradient problem by collecting features from all layers and connecting directly each layer. .

Using an optimal value of parameters after several experiments, all models performed better on the detection of CBB disease from common bean leaves. The proposed model scores a training accuracy of 99.4% and validation accuracy of 98.2%, the VGG16 pre-trained model scores a training accuracy of 96.7% and validation accuracy of 94%, and the Inception pre-trained model scores a training accuracy of 90% and validation accuracy of 88%. Likewise, the proposed model obtained a training loss of 1.7% and a validation loss of 4.2%, the VGG16 pre-trained model obtained a train loss of 9.95% and a validation loss of 13.9% and the Inception pre-trained model obtained a train loss of 27.9% and test loss of 33.5%.

As we discussed in the related work, several types of research are conducted related to plant disease detection. Even though the state-of-art, dataset, and the model they used are different from this study. For instance, the paper[45] wheat leaf Septoria disease classification using image processing and machine learning using wheat leaf dataset, and the performance of the model is 98.7%. However, the authors used conventional machine learning. Another paper presented by [46] used a cotton leaf dataset for the detection of cotton leaf disease using deep learning-based image processing, they achieved around 96.4% accuracy in classification. In this study, we have used modern CNN architectures that are intended to avoid the vanishing gradient problem and improve the performance of the model. In addition, the paper conducted by[7] used bean leaf for the detection of bean leaf disease using modern deep learning framework-based robot vision. They performed comparisons for different models and obtained an accuracy of 98.31%. The current study

increased training accuracy to ~99.4% using the proposed model. A deep learning approach to the identification of CBB disease from common bean leaves has not been studied in Ethiopia, based on the literature review. Thus, we hope as initial this research will motivate other researchers to extend the importance of the deep learning approach for the detection of various diseases even though this research considered only CBB disease

Finally, as a contribution, this study enhanced the common bean crop disease detection method to take action on the crops at an early stage to increase the production of crops in the country.

6 Conclusion and Future Work

Common bean is the world's most essential food legume which is enriched with protein, minerals, calories, energy, and Vitamins. It's an important food legume crop in Ethiopia that gives food and profit protection to the millions of smallholder farmers. However, nowadays due to the common bean suffering from different kinds of diseases, farmers are not getting high-quality production from the crop. Among those diseases, CBB is mostly and widely distributed disease special in Ethiopia, which reduces the production and quality of common bean crops. In the country, to reduce the risks of disease, farmers are using preferred seeds, detecting disease by observation, and also using drugs to prevent disease. But those mechanisms are a very tedious, time-consuming, and costly technique that needs many experts as well as more effort and also not essential. Besides, the shortage of diagnostics tools in developing countries like Ethiopia has an overwhelming impact on their development sectors like agriculture to improve the quality of their life. Thus, for the development of the agriculture sector, there is a vital need for technological solutions to detect the disease at an early stage for effective production. To detect the common bean disease at an early stage we have developed a deep learning approach-based model using CNN algorithms and taken the right action at the right time for the treatment of the crop earlier. We have developed a model based on the DenseNet121 CNN's architecture and proposed a model to identify and classify CBB

disease of common beans using leaf images collected from fields.

During the implementation of this study, we used images that were directly collected from the agriculture research centers with the help of domain experts to annotate the images. According to the result, we obtained from this study during the experimentation, the proposed model detected CBB disease on common bean leaf effectively with better performance with limited resources of hardware on our hands. A proposed model achieved a classification accuracy of 98.2% with training accuracy of 99.2%. Therefore, the result of this study can help the farmers and domain experts to detect CBB disease in the common bean leaf at the early stage for the necessary treatments.

In future work, we intend to extend our proposed approach to detect and classify another disease of common bean to boost the performance of our model and also aim to develop a mobile app with local languages for the farmers that takes an image from the field and instantly gives real-time results for the better treatment of crops with the necessary advice of domain experts. In addition, even though our proposed model performed well for the data collected for this study; we also aim to test with a huge number of images on computational resources like GPU.

Acknowledgements We owe our deepest appreciation to Associate Prof. Misgana Mitiku Plant Pathology Researcher at Southern Agricultural Research Institute for providing us with expert advice about common bacterial blight disease of common beans and supporting us in annotating images of common beans that are affected by common bacterial blight disease.

Data Availability The data used in this study was collected by authors from different agricultural research centers in Ethiopia.

Declarations

Funding No funding was received for conducting this study.

Conflict of Interest The Authors declare they have no conflict of interest.

References

1. D. Ersulo and A. Dana, "Evaluation of Common Bean Varieties for Yield and Yield Component in Segen Area Peoples Zone SNNPRS, Ethiopia," *Issn*, vol. 72, pp. 2224–6088, 2018, [Online]. Available: www.iiste.org
2. B. T and B. KK, "Common Bacterial Blight (*Xanthomonas axonopodis* pv. *phaseoli*) of Beans with Special Focus on Ethiopian Condition," *J. Plant Pathol. Microbiol.*, vol. 08, no. 03, 2017, doi: 10.4172/2157-7471.1000403.
3. B. Mwaipopo *et al.*, "Viruses infecting common bean (*Phaseolus vulgaris* L.) in Tanzania: A review on molecular characterization, detection and disease management options," vol. 12, no. 18, pp. 1486–1500, 2017, doi: 10.5897/AJAR2017.12236.
4. T. Degu, W. Yaregal, and T. Gudisa, "Status of Common Bean (*Phaseolus vulgaris* L.) Diseases in Metekel Zone, North West Ethiopia," *J. Plant Pathol. Microbiol.*, vol. 11, no. 5, pp. 1–8, 2020,

- doi: 10.35248/2157-7471.20.11.494.
5. F. Miruts, G. Gadissa, and B. Roba, "Evaluation of Newly Released Common Bean Varieties through On-Farm Demonstrations in ATJK and Shalla Districts of Oromia Regional State, Ethiopia," *Int. J. Res. Stud. Agric. Sci.*, vol. 6, no. 1, pp. 43–48, 2020, doi: 10.20431/2454-6224.0601006.
6. W. Adila, H. Terefe, and A. Bekele, "Association of common bacterial blight (*Xanthomonas axonopodis* pv. *phaseoli*) epidemics with agro ecological factors in Southwestern Ethiopia," *Int. J. Agric. Res. Innov. Technol.*, vol. 11, no. 1, pp. 74–83, 2021, doi: 10.3329/ijarit.v11i1.54469.
7. S. H. Abed *et al.*, "A modern deep learning framework in robot vision for automated bean leaves diseases detection," *Int. J. Intell. Robot. Appl.*, vol. 5, no. 2, pp. 235–251, 2021, doi: 10.1007/s41315-021-00174-3. <https://doi.org/10.1007/s41315-021-00174-3>
8. R. Bhagwat and Y. Dandawate, "A Review on Advances in Automated Plant Disease Detection," *Int. J. Eng. Technol. Innov.*, vol. 11, no. 4, pp. 251–264, 2021, doi: 10.46604/IJETI.2021.8244.
9. R. Y. M. M. T. K. M. A. and G. P., "Analysis of the Molecular Diversity of Common Bacterial Blight (*Xanthomonas campestris* pv. *phaseoli* and *X. campestris* pv. *phaseoli* var. *fuscans*) Strains from Ethiopia Revealed by Rep-PCR Genomic Fingerprinting," *J. Biotechnol. Biomater.*, vol. 08, no. 04, 2018, doi: 10.4172/2155-952x.1000286.
10. A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agric. Technol.*, vol. 3, no. May 2022, p. 100083, 2023, doi: 10.1016/j.atech.2022.100083. <https://doi.org/10.1016/j.atech.2022.100083>
11. S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 305–310, 2022, doi: 10.1016/j.gltp.2022.03.016.
12. R. Sharma *et al.*, "Plant disease diagnosis and image classification using deep learning," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 2125–2140, 2022, doi: 10.32604/cmc.2022.020017.
13. P. Sahu, A. Chug, A. P. Singh, D. Singh, and R. P. Singh, "Deep Learning Models for Beans Crop Diseases: Classification and Visualization Techniques," *Int. J. Mod. Agric.*, vol. 10, no. 1, pp. 796–812, 2021.
14. M. M. Islam *et al.*, "DeepCrop: Deep learning-based crop disease prediction with web application," *J. Agric. Food Res.*, vol. 14, no. July, p. 100764, 2023, doi: 10.1016/j.jafr.2023.100764. <https://doi.org/10.1016/j.jafr.2023.100764>
15. J. Foucher *et al.*, "Improving Common Bacterial Blight Phenotyping by Using Rub Inoculation and Machine Learning: Cheaper, Better, Faster, Stronger," *Phytopathology*, vol. 112, no. 3, pp. 691–699, 2022, doi: 10.1094/PHYTO-04-21-0129-R.
16. D. BB, "Common Bean Improvement Status (*Phaseolus vulgaris* L.) in Ethiopia," *Adv. Crop Sci. Technol.*, vol. 06, no. 02, 2018, doi: 10.4172/2329-8863.1000347.
17. K. Belachew, M. Gebremariam, and K. Alemu, "Integrated Management of Common Bacterial Blight (*Xanthomonas axonopodis* pv. *Phaseoli*) of Common Bean (*Phaseolus vulgaris*) in Kaffa, Southwest Ethiopia," vol. 3, no. 2, pp. 85–90, 2016.
18. T. Belete, "Plant Pathology & Microbiology Common Bacterial Blight (*Xanthomonas axonopodis* pv. *phaseoli*) of Beans with Special Focus on Ethiopian Condition," no. December 2018, 2017, doi: 10.4172/2157-7471.1000403.
19. A. Abraham, "Emerging Plant Virus Disease in Ethiopian Agriculture: Causes and Control Options Among the plant pathogens, viruses are considered second only to fungi with respect to," vol. 29, no. 1, pp. 39–55, 2019.
20. L. Science and T. Issn, "Adaptation Study of Improved Haricot Beans (*Phaseolus Vulgaris* L.) Varieties at Western Oromia, Haro," vol. 25, pp. 43–47, 2014.
21. F. Martinelli *et al.*, "Advanced methods of plant disease detection. A review," pp. 1–25, 2015, doi: 10.1007/s13593-014-0246-1.
22. L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8. <https://doi.org/10.1186/s40537-021-00444-8>
23. M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in Deep Learning*, vol. 57. 2020. doi: 10.1007/978-981-13-6794-6. <http://link.springer.com/10.1007/978-981-13-6794-6>
24. D. Wu and G. G. Wang, "Causal artificial neural network and its applications in engineering design," *Eng. Appl. Artif. Intell.*, vol. 97, no. October 2020, 2021, doi: 10.1016/j.engappai.2020.104089.
25. Bharadwaj, K. B. Prakash, and G. R. Kanagachidambaresan, "Pattern Recognition and Machine Learning," in *EAI/Springer Innovations in Communication and Computing*, 2021, pp. 105–144. doi: 10.1007/978-3-030-57077-4_11.
26. N. Kriegeskorte and T. Golan, "Neural network models and deep learning," *Curr. Biol.*, vol. 29, no. 7, pp. R231–R236, 2019, doi: 10.1016/j.cub.2019.02.034. <http://dx.doi.org/10.1016/j.cub.2019.02.034>
27. Y. Kibru and T. Debelee, "Detection of Bacterial Wilt on Enset Crop Using Deep Learning Approach," *Int. J. Eng. Res. Africa*, vol. 51, pp. 131–146, Nov. 2020, doi: 10.4028/www.scientific.net/JERA.51.131.
28. T. B. Shahi, C. Y. Xu, A. Neupane, and W. Guo, "Recent Advances in Crop Disease Detection Using UAV and Deep Learning Techniques," *Remote Sens.*, vol. 15, no. 9, pp. 1–29, 2023, doi: 10.3390/rs15092450.
29. L. Xu, B. Cao, S. Ning, W. Zhang, and F. Zhao, "Peanut leaf disease identification with deep learning algorithms," *Mol. Breed.*, vol. 43, no. 4, pp. 1–12, 2023, doi: 10.1007/s11032-023-01370-

8. <https://doi.org/10.1007/s11032-023-01370-8>
30. K. L. Narayanan *et al.*, "Banana Plant Disease Classification Using Hybrid Convolutional Neural Network," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/9153699.
31. J. A. Pandian, V. D. Kumar, O. Geman, M. Hnatiuc, M. Arif, and K. Kanchanadevi, "applied sciences Plant Disease Detection Using Deep Convolutional Neural Network," 2022.
32. H. Chen *et al.*, "AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf," pp. 1–17, 2022.
33. P. S. Thakur, T. Sheorey, and A. Ojha, "VGG-ICNN: A Lightweight CNN model for crop disease identification," *Multimed. Tools Appl.*, vol. 82, no. 1, pp. 497–520, 2023, doi: 10.1007/s11042-022-13144-z.
34. T. Gupta, Titunath, and V. Jain, "Plant Disease Detection using Deep Learning," *Int. Conf. Sustain. Comput. Smart Syst. ICSCSS 2023 - Proc.*, pp. 202–206, 2023, doi: 10.1109/ICSCSS57650.2023.10169268.
35. N. Merlina, E. Noersasongko, P. Nurtantio, M. A. Soeleman, D. Riana, and S. Hadiani, "Detecting the Width of Pap Smear Cytoplasm Image Based on GLCM Feature," in *Smart Innovation, Systems and Technologies*, 2021, pp. 231–239. doi: 10.1007/978-981-15-5224-3_22.
36. H. Tavakoli, P. Alirezazadeh, A. Hedayatipour, A. H. Banijamali Nasib, and N. Landwehr, "Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks," *Comput. Electron. Agric.*, vol. 181, no. 960115364, p. 105935, 2021, doi: 10.1016/j.compag.2020.105935. <https://doi.org/10.1016/j.compag.2020.105935>
37. M. A. Haque *et al.*, "Deep learning-based approach for identification of diseases of maize crop," *Sci. Rep.*, vol. 12, no. 1, pp. 1–14, 2022, doi: 10.1038/s41598-022-10140-z. <https://doi.org/10.1038/s41598-022-10140-z>
38. E. D. Cubuk, B. Zoph, D. Man, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning Augmentation Strategies from Data," no. Section 3, pp. 113–123, 2012.
39. M. Mahdi and B. Mehdi, *A systematic review on overfitting control in shallow and deep neural networks*, vol. 54, no. 8. Springer Netherlands, 2021. doi: 10.1007/s10462-021-09975-1. <https://doi.org/10.1007/s10462-021-09975-1>
40. M. Dhouibi, A. Karim, B. Salem, and S. Ben Saoud, "Optimization of CNN model for image classification," no. August, 2021, doi: 10.1109/DTS52014.2021.9497988.
41. K. Q. Weinberger, "Densely Connected Convolutional Networks," no. April, 2017, doi: 10.1109/CVPR.2017.243.
42. N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "DenseNet Convolutional Neural Networks Application for Predicting COVID - 19 Using CT Image," *SN Comput. Sci.*, vol. 2, no. 5, pp. 1–11, 2021, doi: 10.1007/s42979-021-00782-7. <https://doi.org/10.1007/s42979-021-00782-7>
43. S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep Spatial – Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, 2019, doi: 10.1109/TITS.2019.2906365.
44. V. Singh, Varsha, and A. K. Misra, "Detection of unhealthy region of plant leaves using image processing and genetic algorithm," *Conf. Proceeding - 2015 Int. Conf. Adv. Comput. Eng. Appl. ICACEA 2015*, pp. 1028–1032, 2015, doi: 10.1109/ICACEA.2015.7164858.
45. M. K. KAHSAY, "CLASSIFICATION OF WHEAT LEAF SEPTORIA DISEASE USING IMAGE PROCESSING AND DISEASE USING IMAGE PROCESSING AND," *ADDIS ABABA Sci. Technol. Univ. Classif.*, no. June, 2019.
46. M. Azath, M. Zekiws, and A. Bruck, "Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis," *J. Electr. Comput. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9981437.

