Performance Analysis of Detection of Disease on Leaf Images with Inception V3 and Mobilenet Deep Learning Techniques

Dr.S. Saritha[1], Dr.V. Satya Srinivas[2], Dudi Anuhya[3], Godugu Pavithra[4]
[1]Associate Professor, Electronics and Communication Engineering, Geethanjali College of Engineering and Technology, JNTUH, Hyderabad, India.
E-mail: saritha.saladi3188@gmail.com
[2]Associate Professor, Electronics and Communication Engineering, Geethanjali College of Engineering and Technology, JNTUH, Hyderabad, India.
E-mail: sathyavemuri@gmail.com
[3]Student, Electronics and Communication Engineering, Geethanjali College of Engineering and Technology, JNTUH, Hyderabad, India.
E-mail: anuhyadudi@gmail.com
[4]Student, Electronics and Communication Engineering, Geethanjali College of Engineering and Technology, JNTUH, Hyderabad India.
E-mail: pavithragodugu2000@gmail.com

Abstract

Indian economy depends a lot on agriculture. Around 38% of the land in India is suitable for agriculture. The crop in the agricultural land is threatened by the diseases. These diseases may have been caused by microorganisms, viruses, fungi, and bacteria. The traditional method of identifying the crop disease by the naked eye needs enormous labor, expertise with crop diseases, and needs a lot of time. In most cases, by the time the results are out, the damage is already done. To avoid such problems, an automated system can be adapted for the detection of plant diseases. The aim is to build a model that is fast, accurate, and reliable that can be used for the identification of leaf diseases. In this article, we compared three deep learning neural networks we evaluated the performance of three deep neural networks and analyzed the strengths of each neural network interms of accuracy. We also determined the working of each model and its applications.

Keywords: Leaf Diseases, Deep Learning, CNN, AlexNet, MobileNet, Inception V3, Performance Metrics.

DOI: 10.47750/pnr.2022.13.S03.018

Introduction

Agriculture contributes around 20% to GDP. It has had a major impact on the Indian economy. India is not only the largest producer but also the largest exporter of agricultural goods. India is in the first position in producing organic food. Most industries depend on raw products produced in agriculture. Therefore, agriculture is the backbone of all countries. Agriculture growth affects the economy, development, and GDP of each country. According to research, there is a need to produce 70 percent more yield than today to feed the growing population of the world by 2050. Many studies say the quality of the crop is reduced due to plant diseases. In some cases, these diseases can wipe out an entire harvest. In most cases, the diseases can be seen on the leaves, stems, and fruits of the plant. Thus, identifying these at the early stages is crucial.

There are many techniques to identify the type of leaf disease in the plant. The traditional way is to identify the disease with the naked eye. It is a time-consuming process and not an efficient method. Image processing and Deep Learning techniques can be employed for detecting plant disease. This method saves time, effort, and use of pesticides, and produces accurate results.

There exist different segmentation techniques using supervised and unsupervised machine learning techniques [1,2,3,4,5,6].

This paper discusses image pre-processing, deep learning methods to detect leaf disease, comparison of the models. Using deep learning techniques, the features are deeply processed at each layer. Different features are extracted at each layer.

The proposed system has four stages: image preprocessing, creating a model, training the images, and visualization of the results. Image preprocessing starts with rescaling the images and generating multiple augmented images. Three deep learning models are built in this paper, AlexNet, MobileNet, and Inception V3.

Our goal is to do a comparative analysis of those three modules and train them.

LITERATURE SURVEY

A. A Deep Learning-based Approach for Banana Leaf Diseases Classification

This research paper[7] was proposed by Jihen Amara, Bassem Bouaziz, and Alsaye Algregawy in the year 2017. They used a Convolutional Neural Network called LeNet to classify and identify the disease on banana leaves. The two main components of this research are image preprocessing and deep learning-based classification. In the image preprocessing part, the images are resized to 60 x 60 and converted into greyscale to standardize the further process. For the classification part, fully connected layers are created where each neuron is connected to all the neurons of the next layer. The proposed CNN model has three parts – Convolution, Pooling, and Fully connected layers. In Convolution and the Pooling layers, the features are extracted. The fully connected layer with the SoftMax activation layer is used to classify the input images into predefined classes. This architecture produced an accuracy of 98%.

B. Farmer buddy-web-based cotton leaf disease detection

In this paper, Convolutional Neural Network[8,9,10] is implemented to detect cotton leaf disease. They mainly focused on the Alternaria Macrospore and Bacterial Blight disease of cotton. The model proposed in this paper aims to overcome the problem of locating a feature in an image. CNN is used to extract features from an image. The image is processed first where the images are digitized. Then the digitized images are given to the CNN model. The purpose of the proposed system is to develop a user-friendly web application for farmers, to identify the cotton leaf disease accurately from the images, and to provide preventive measures for the detected diseases. The training dataset consists of 513 images and the testing dataset consists of 207 images. the first step is to convert the RGB image into a gray image and resize them into 128x128. These images are passed through three hidden layers where feature extraction, pooling, and flattening are also performed. The output of the three layers is given to the SoftMax layer where the images are classified. This model achieved training accuracy of 80% and testing accuracy of 89%.

C. Using CNN using deep learning for image-based plant disease detection

This paper[11] used an open-source dataset of 54,306 images of both healthy and diseased. These images are classified into 14 crop spices and 26 diseases. This paper talks about two deep convolutional neural networks namely GoogleNet and AlexNet. They have implemented 60 different configurations. The varying parameter is deep learning architecture, mechanism of training, dataset type, and a split ratio of train and test dataset. The training of models is done on GPU because the CPU takes a lot of time for training the model. The overall accuracy varied between 85.53% (for a variation of AlexNet) to 99.34% (for a variation of GoogleNet). This paper concluded that GoogleNet performs better than the AlexNet model.

D. Crop disease detection using deep learning

This paper[12,13] was published by Omkar Kulkarni who discussed the immunity of the crop affected by climate change and how deep learning techniques help to classify the healthy crop and the diseased crop. Transfer learning is used to train the images instead of training the images from scratch. All the images are first resized into 224x224 and then converted into grey images. in the preprocessing of the image, noises from the images are removed and segmentation is done. The proposed system says that the classification is performed in two stages – first, the type of crop must be detected then the type of disease must be detected. A pre-trained model of MobileNet and Inception V3 is used. The results show that the MobileNet has achieved an accuracy score of 99.62% whereas the Inception performed slightly better than MobileNet with an accuracy score of 99.74%.

PROPOSED CONVOLUTION NEURAL NETWORK MODELS

A. Methodology

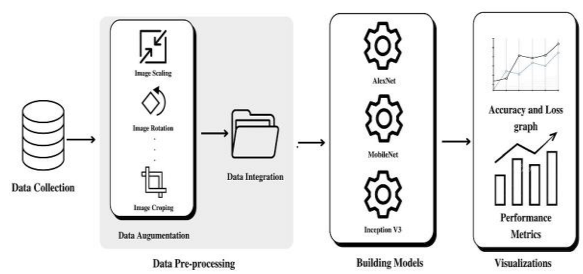

Fig. 1: Proposed Methodology

B. Data Pre-processing

First, all the images are resized to 224 x 224 pixels and 3 channels, which is given as standard input for all the deep learning models. Then using the data augmentation method, various changes are made to the images and multiple artificial images are generated for all the resized images. Changes include rescaling, shifting, random erasing, color space transformation, etc. Newly generated images are now integrated into a folder for further process.

C. CNN Models

AlexNet

In 2012, Alex Krizhevsky designed a CNN model called AlexNet. The primary purpose of Alexnet is Image recognition. The Alexnet model won the ImageNet LSVR Challenge in 2012, with 15.3% of the top five errors. Though this model was computationally expensive it was made possible because of using GPUs during the training process.

1) Architecture

The architecture of AlexNet usually consists of 5 convolutional layers. The 5 convolution layers are combination of 3 are max-pooling layers, 2 are normalization layers, 2 are fully connected layers, and 1 is a SoftMax layer. Each convolutional layer is followed by filters and ReLU functions.

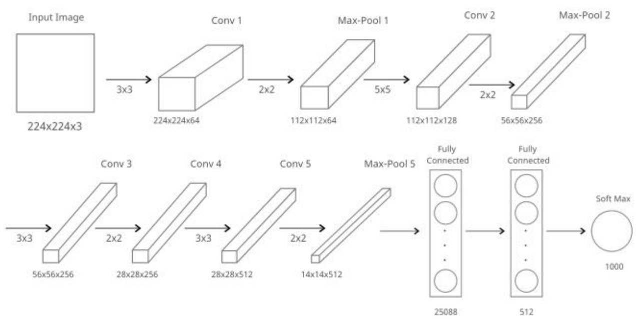

Fig. 2: AlexNet Architecture

2) Layers

a) Max Pooling

The Max pool layer is used to downsample an image. The AlexNet model uses a concept of overlapping max-pooling which is similar to the max-pooling concept. Overlapping max-pooling uses windows of size 3x3 with a stride of 2.

b) ReLU nonlinearity

Rectified Linear Unit is an activation function used in neural networks. The ReLU layer outputs the input directly if it is positive and zero otherwise.

Mathematically,

$$x = \max(y, 0) \quad (1)$$

Where y is the input and x is the output

When compared to other activation layers such as tanh, and sigmoid, ReLU is computationally more efficient and has better convergence performance.

c) Overfitting problem

Overfitting is the main problem in the AlexNet model. To avoid overfitting below two techniques are used.

i) Data Augmentation:

Data augmentation is the most common and easiest way to avoid overfitting. It is a method of generating a lot of similar images. This increases the size of the dataset and thus overfitting is reduced.

ii) Dropout:

In the layer, a random neuron is dropped from the neural network to avoid overfitting. The probability of dropout is 0.2. The neuron which is dropped does not contribute to forward propagation and backward propagation.

d) Optimizer

Optimizers are accountable for changing the parameters such as learning rate and weights to minimize the losses. Also, they help in getting the results faster. We used the Adam optimizer in our model. Adaptive Momentum or Adam is a combination of Momentum and RMSProp. Adam minimizes the loss and works efficiently.

i) Momentum: this method uses the concept of 'exponentially weighted average' of the gradients to fasten the gradient descent algorithm.

ii) Root Mean Square Propagation: Root Mean Square Propagation also called RMSprop is an advanced version of AdaGrad. AdaGrad is the cumulative sum of squared gradients whereas Adam is the exponential moving average.

MobileNet:

MobileNet is a type of deep learning that focuses on mobile applications and embedded vision applications. MobileNet is the first mobile computer vision model of TensorFlow. It is built on the concept of depthwise separable convolution.

Depthwise Separable Convolution

The MobileNet model is based on depthwise separable convolutions which reduce the number of parameters and result in a lightweight neural network. Unlike the standard convolution methods, MobileNet separates layers for filtering followed by separate layers for combining.

The depthwise separable convolution is a combination following two operations:

i. Depthwise convolution
ii. Pointwise convolution

The idea of depthwise separable convolution is to filter the depths and separate them into spatial dimensions. Each layer is followed by batch Normalization and ReLU non-linearities.

i) Depthwise Convolution:

It is a channel-wise spatial convolution (where is the dimensions of an image).

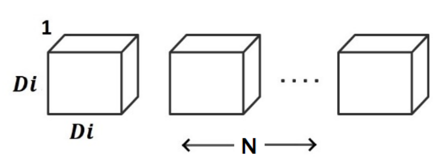

Fig. 3: Depthwise Convolution

The result of Depthwise Convolution will be:

$$Di \times Di \times k \times D_F \times D_F \quad (2)$$

Since a single convolution is applied to each channel, the number of output channels and the number of input channels will be equal.

ii) Pointwise Convolution:

The output of a depthwise convolution is combined with a kernel of size 1x1. In this convolution, the features are combined.

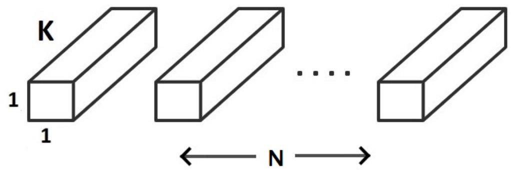

Fig. 4: Pointwise Convolution

The computational cost of pointwise convolution is:

$$k \times N \times D_F \times D_F \ (3)$$

After combining the outputs of the above two operations the resultant output will be:

$$\frac{Di \times Di \times k \times D_F \times D_F + k \times N \times D_F \times D_F}{Di \times Di \times k \times N \times D_F \times D_F} \quad (4)$$

$$= \frac{1}{N} + \frac{1}{{D_F}^2} \ (5)$$

From the above, we can conclude that MobileNet is 8 to 9 times less computational than the other standard convolutions.

iii) Width Multiplier:

The width multiplier is a parameter in MobileNet that makes the model even smaller and less computationally expensive model. It reduces the size of each layer uniformly. Suppose the input layer has K number of channels and the width multiplier is then the number of input channels will become and the number of output channels will becomeμN. Where.

The typical value of μ is 0.25,0.5,0.75,1. The Width multiplier reduces the computational cost and the number of parameters$\mu\ ^2$.

The total computational cost is:

$$D_i \times D_i \times \mu k \times D_F \times D_F + \mu K \times \mu N \times D_F \times D_F \ (6)$$

iv) *Resolution Multiplier*:

The Resolution Multiplier (ϑ) is a hyper-parameter that is used to reduce the computational cost of a neural network. A resolution multiplier is applied to input images and the internal representation of each layer is reduced byϑ.

Where $\vartheta \in (0,1]$

Resolution Multiplier reduces the computational cost byϑ^2.

The total computational cost with width multiplier and resolution multiplier can be expressed as:

$$D_i \times D_i \times \mu k \times \vartheta D_F \times \vartheta D_F + \mu K \times \mu N \times \vartheta D_F \times \vartheta D_F (7)$$

Inception V3:

The Inception V3 is a Convolutional Neural Network used for image classification released in the year 2015. It is an advanced version of the Inception V1 model. Inception V3 has better optimization techniques, less computational cost, a deeper neural network, and higher efficiency. It has 42 layers.

The major modifications on Inception V3 are:

i) Factorization of Convolutions:

The larger convolutions are factorized into smaller convolutions to reduce the dimensions. Consider a module of the Inception V1 model.

The above module has 5x5 convolutional layers therefore the computational cost is expensive. By replacing the 5x5 convolution layers with two 3x3 convolution layers the computational cost is reduced.

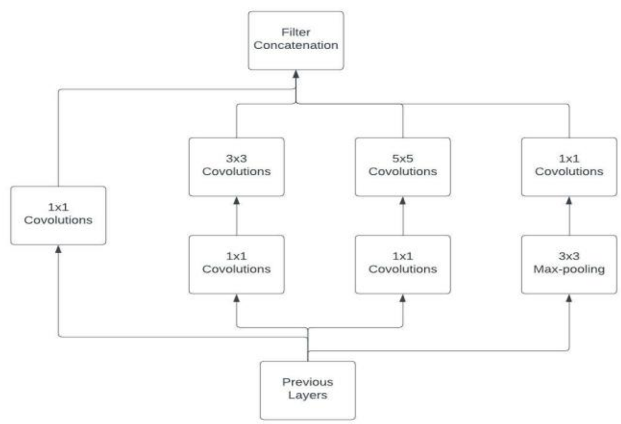

Fig. 5: Inception V1 Model

As a result, the number of parameters is reduced. Thus, the computational cost is also reduced.

ii) Auxiliary Classifiers:

Auxiliary Classifiers' objective is to improve the convergence of a very deep neural network. This helps vanish the gradient problem in a neural network. Though the Auxiliary Classifiers do not affect the early stage of the training, at the end of training Auxiliary Classifiers show higher accuracy. Therefore, Auxiliary Classifiers act as a regularizer in the Inception V3 model.

iii) Efficient Grid Size Reduction:

In the inception V3 model, the activation dimension of the network filters is expanded to reduce the grid size. If we consider a grid size of n x n with k filters, after reduction the grid size will become n/2 x n/2 with 2k filters. Both convolution and pooling are done in two parallel blocks and concatenated.

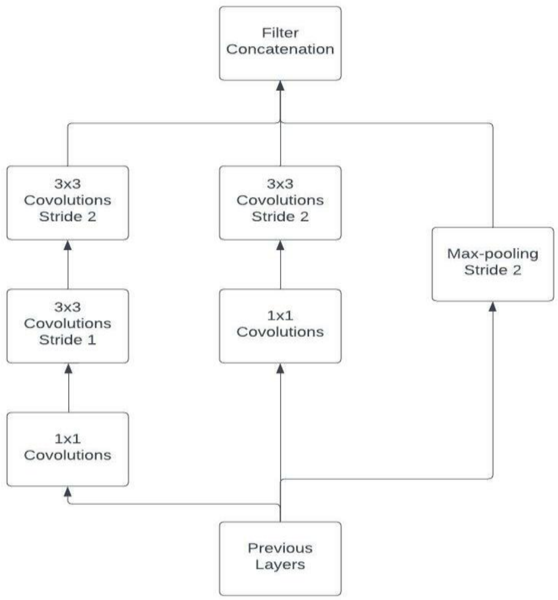

Fig. 6: Inception V3 Model with Extended Filter Banks. In the above architecture, the 5x5 convolutional layer is divided into two 3x3 Convolutional layers

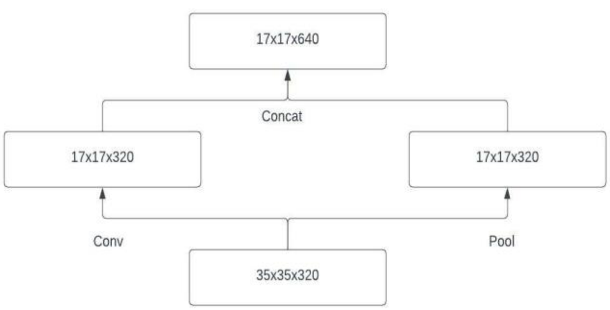

Fig. 7: Internal implementation of Inception V3. Given an input image of 35x35x320, is divided into two 17x17x320 (one convolution layer and one max-pooling layer) and concatenated. Therefore, the output will be 17x17x640

DATASET

The dataset that is being used is taken from Kaggle [14]. The dataset consists of approximately 87000 color images of healthy and unhealthy leaves. There are 13 plants and 38 classes in the dataset. The plant species include potato, corn maize, bell pepper, apple, etc. and the 38 classes have both healthy and diseased leaves. Diseases include black rot, bacterial spot, powdery mildew, w, etc.

RESULTS

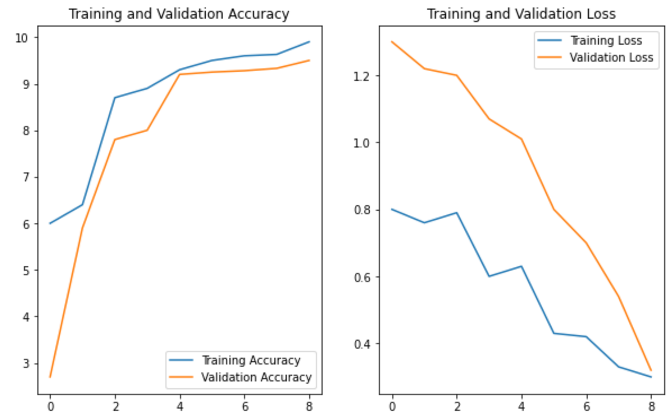

Fig. 8: Inception V3 Model Accuracy and Loss graph. Training and Validation Accuracy graph (On Left) and Training and Validation Loss Graph (on Right)

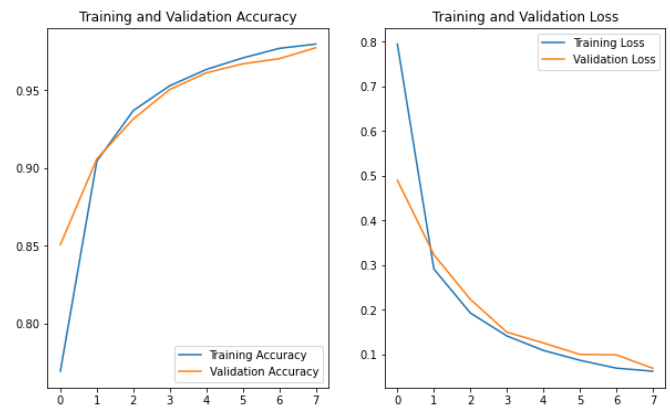

Fig. 9: MobileNet Model Accuracy and Loss graph. Training and Validation Accuracy graph (On Left) and Training and Validation Loss Graph (on Right)

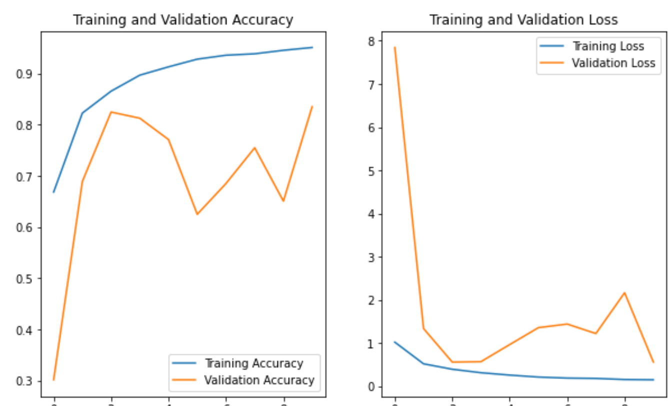

Fig. 10: AlexNet Model Accuracy and Loss graph. Training and Validation Accuracy graph (On Left) and Training and Validation Loss Graph (on Right)

Figures 8,9,10 shows the performance of the deep learning algorithms in terms of accuracy and loss.

Table I: Accuracy and Loss values during the training process.

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
Inception V3	0.9904%	0.9794%	0.0325%	0.0753%
MobileNet	0.9807%	0.9747%	0.0570%	0.0842%
AlexNet	0.9897%	0.9564%	0.0430%	0.1412%

Table II: Comparison of Performance Metrics between AlexNet, MobileNet, and Inception V3

Performance Matrix	Inception V3	MobileNet	AlexNet
True Positive	0.806	0.025	0.026
True Negative	36.30	36.16	36.08
False Positive	0.179	0.978	0.974
False Negative	0.179	0.972	0.975
Sensitivity/Recall	0.817	0.025	0.026
Specificity	0.995	0.973	0.973
Precision	0.833	0.026	0.026
Negative Predicted Value	0.995	0.973	0.973
False Positive Value	0.004	0.026	0.026
False Negative Value	0.182	0.973	0.973
False Discovery Rate	0.166	0.973	0.974

Each parameter of performance metrics is described below:

True Positive (TP): A result where the model predicts the positive class correctly.

True Negative (TN): A result where the model predicts the negative class correctly.

False Positive (FP): A result where the model predicts the positive class incorrectly.

False Negative (FN): A result where the model predicts the negative class incorrectly.

Precision: the probability of truly predicted values out of all positively predicted values. Precision is the measure of quality. The value of precision lies between 0 and 1

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

Recall The percentage of positive predicted values out of all total positives. It is a measure of quantity.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

F1 Score: It is calculated as the harmonic mean of Recall and Precision.

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (10)$$

Accuracy: the percentage of correctly predicted out of all the predictions

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

Specificity: the percentage of correct negative predictions out of all negative predictions. TNR lies between 0 and 1. The model having a value near 1 is a better model.

$$TNR = \frac{TN}{TN + FP} \quad (12)$$

Negative predictive value: the ratio of negative results to total negative results.

$$NPV = \frac{TN}{TN + FN} \quad (13)$$

Positive Prediction Value: the ratio of positive results to total positive results

$$PPV = \frac{TP}{TP + FP} \quad (14)$$

Fall out or false positive rate: the probability of false results.

$$FPR = \frac{FP}{TN + FP} \quad (15)$$

False negative rate: the number of true positive values that are missed.

$$FNR = \frac{FN}{TP + FN} \quad (16)$$

False discovery rate: FDR is the measure of accuracy when multiple parameters are being calculated at once.

$$FDR = \frac{FP}{TP + FP} \quad (17)$$

Table III: Comparison of the number of parameters of each model

Model	Total Trainable Parameters	Trainable Parameters	Non-trainable Parameters
Inception V3	22,871,366	1,068,582	21,802,784
MobileNet	4,390,894	156,918	4,233,976
AlexNet	10,642,678	10,620,198	4,480

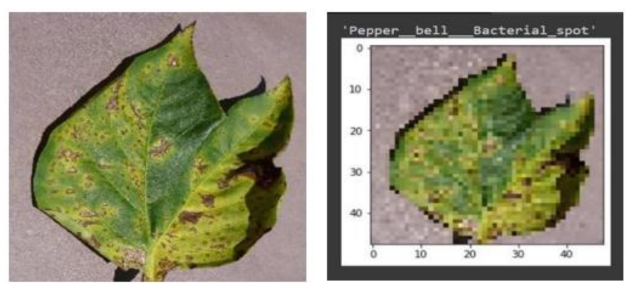

Fig. 11: Bell Pepper Bacterial Spot

Conclusion

The main aim of the models is to detect leaf diseases. For detecting the leaf disease, images of leaves are used. These images are trained using three different models, AlexNet, MobileNet, and Inception V3. In this article, we have compared all three models based on the computational cost, accuracy, and parameters of performance metrics. From the results, the computational cost of Inception V3 falls between

AlexNet and MobileNet. It is also clear from the results that the accuracy of Inception V3 is higher i.e., 99.04%, and also it has better values of performance metrics compared to the other two models. the computational cost of the Inception V3 comes between AlexNet and MobileNet. Therefore, the Inception V3 model is better than the other two models.

References

Y Karuna, Saritha Saladi, and C. V. Narasimhulu. “Segmentation of tumor using PCA based modified fuzzy C means algorithms on MR brain images.” International Journal of Imaging Systems and Technology 30, no. 4 (2020): 1337-1345.

Karuna, Y, Saritha Saladi, Pramodh Konduru, and G. Ramachandra Reddy. “Magnetic Resonance Brain Images Individual Recognition with PCA.” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-4S February, 2019.

Karuna, Y., Saladi, Saritha, & Bhattacharyya, B. “Brain Tissue Classification using PCA with Hybrid Clustering Algorithms. “International Journal of Engineering & Technology, 2018 7(2.24), 536-540.

Saladi Saritha, and N. Amutha Prabha. “A Novel Fuzzy Factor For MRI Brain Image Segmentation with Intuitionistic Fuzzy Kernel Clustering Approach.” Journal of Advanced Research in Dynamical and Control Systems 2018, Volume 10-Special Issue-02:639-652.

Saladi Saritha, and N. Amutha Prabha, “MRI brain segmentation in combination of clustering methods with Markov random field”. International Journal of Imaging Systems and Technology, 00:1–10.

Saritha, Saladi, and N. Amutha Prabha, “Analysis of Denoising Filters on MRI Brain Images”, in International Journal of Imaging Systems and Technology, Volume 27, no. 3 (2017): 201-208.

“Detection of leaf disease using convolutional neural network”, by, A T Madhavi1, MM Yugesh, Srinidhi K, CSS. Vijayasundar published in the year 2020 (International journal of current engineering and scientific).

“Farmer Buddy-Web Based Cotton Leaf Disease Detection Using CNN” a research paper by Shantanu Kumbhar, Amita Nilawar, Shruti Patil, Bodireddy Mahalakshmi, Manasi Nipane from the year 2019.

Latha, Kethu Sneha, Y Karuna, and Saladi Saritha. “Brain Tumor Segmentation, Detection and Grading in MRI Images.” Design Engineering 5 (2021): 1841-1852.

Pavan, Panditi Sai, Y Karuna, and Saladi Saritha. “Mri Brain Tumor Segmentation With Slic And Convolutional Neural Networks.” Journal of Critical Reviews 7, no. 19 (2020): 4454-4462.

“Using Deep Learning for Image-Based Plant Disease Detection” proposed by Sharada P. Mohanty, David P. Hughes, and Marcel Salathé.

Jihen Amara, Bassem Bouaziz, and Alsayed Algergawy research paper “A Deep Learning-based Approach for Banana Leaf Diseases Classification” (2017)

Debasish Das, Mahinderpal Singh, Sarthak Swaroop Mohanty, S. Chakravarty, “Leaf Disease Detection using Support Vector Machine”, Published in 2020 International Conference on Communication and Signal Processing (ICCSP) by IEEE.

https://www.kaggle.com/datasets /vipoooool/new-plant-diseases-dataset.

Ibrahim, S., & Koksal, M. E. (2021). Commutativity of sixth-order time-varying linear systems. Circuits, Systems, and Signal Processing, 40(10), 4799-4832.