International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT2410333

Detection and Classification on Plant Disease using Deep Learning Techniques

Chavda Rohit P., Tosal Bhalodia

Department of Computer Engineering, Atmiye University, Rajkot, Gujarat, India

ARTICLEINFO

Article History:

Accepted: 10 May 2024

Published: 28 May 2024

Publication Issue

Volume 10, Issue 3

May-June-2024

Page Number

365-375

ABSTRACT

Plant diseases are a major problem for the agriculture industry because they can cause large crop losses and jeopardize food security. Deep learning approaches have demonstrated encouraging results in automating plant disease diagnosis and detection in recent years. In the context of plant disease diagnosis, this study examines the efficacy of two well-known convolutional neural network architectures: DenseNet121 and VGG16. Plant Village datasets are used for pretrained and fine-tuning of the DenseNet121 and VGG16 architectures, respectively. The dataset includes Images of both healthy and sick plants. To guarantee the models' resilience and generalizability, the dataset include 15 different classes and 4 types of plants namely Tomato, Potato and Pepper Bell. We compare the accuracy, precision, recall, and F1-score of DenseNet121 and VGG16 for plant disease classification through extensive testing and analysis. To determine if they are practically feasible for use in real-world applications, we also examine their model complexity and computing efficiency.

Our findings show that DenseNet121 and VGG16 can both correctly diagnose plant diseases in a variety of species. Although DenseNet121 outperforms VGG16 in terms of overall accuracy and computational efficiency, both models obtain high accuracy rates. Additionally, DenseNet121 has superior generalization performance, especially in identifying uncommon or underrepresented illness classes. All things considered, this work emphasizes the promise of deep learning models-DenseNet121 in particular-as useful instruments for automated plant disease identification and points to directions for further investigation to improve the efficiency and scalability of such systems for real-world use in agriculture.

Keywords : Plant Diseases Detection and Classification, Transfer Learning, Deep Learning, DenseNet-121, VGG-16.

I. INTRODUCTION

In order to maintain crop health and optimize productivity, it is important in agriculture to identify and categorize plant diseases. Conventional illness detection techniques frequently depend on the subjective and time-consuming visual evaluation of human specialists. It has becoming increasingly common to automate these procedures using machine learning algorithms as deep learning techniques, especially in computer vision, progress.

Promising approaches for precise and effective plant disease identification and categorization come from deep learning algorithms. Deep learning models may be trained to accurately identify patterns and characteristics that indicate different plant illnesses by utilizing extensive datasets of Images showing both healthy and sick plants.

This study investigates the use of deep learning methods in the identification and categorization of plant diseases. We talk about the drawbacks of conventional techniques and how deep learning techniques can help to solve them. We also examine the most recent deep learning architectures and techniques used to plant disease classification and detection problems.

The following research topics on disease detection in the leaves of different plants have been the main focus of this research article.

- Raw image preprocessing
- Identification of low-level features
- Strategies for enhancing the accuracy of categorization and recognition
- Examining publicly available plant datasets
- Examination of CNN-based techniques for plant disease categorization.

Rest of the paper is organized as follows;

In section I, we have introduced the concept of deep learning for the image classification and detection. Section II describes the state-of-the-art algorithms in the field of convolution neural network. In section III, review of existing work has been done. Section IV analyze the existing work and potential research issues in the field of diseases detection from the plant images. Section V concludes the paper.

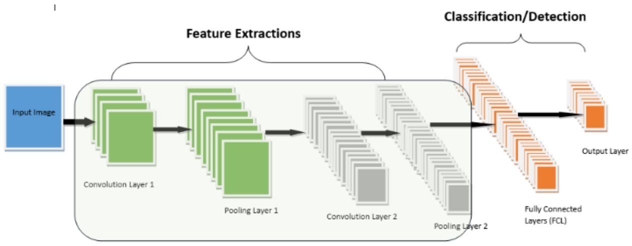

Fig. 1. Deep learning Architecture

II. INTRODUCTION TO DEEP LEARNING FOR IMAGE CLASSICIATION AND DETECTION

Prior to the development of deep learning, object identification techniques were developed using manually created features and shallow trainable neural network designs. By building complicated ensembles that mix several low-level image characteristics with high-level information from object detectors and scene classifiers, their performance became static for complex image recognition tasks. Low level feature extractions are being introduced to solve issues with standard designs while deep learning architectures are being developed at a rapid pace. When it comes to network design, these models act differently. as shown in fig. 1.

A. Feature Extractions

The procedure known as "feature extraction" involves turning raw data into numerical features so that the original data set's contents may be processed. It generates superior results as compared to directly using machine learning on raw data.

Convolution layers are used to extract the features from the input training data. Each convolution layer has a set of filters to help in feature extraction. Convolution layers often learn features with increasing complexity as CNN model depth increases.

Features are obtained by taking into account the convolution of a portion of the sample data; the amount of the data section that the filter traverses each time is determined by the stride length and padding value; zero padding may or may not be given to data samples before convolution.

Following that, an activation unit called a Rectified Linear Unit (ReLU) is used to process the convolution result. This unit converts the data into a non-linear form. The ReLU output is only cut to zero if the convolution output is negative.

The ReLU output is then processed by a pooling layer. Duplicate features are captured by convolution and eliminated by the pooling layer. This layer thereby reduces the size of the data sample. Pooling operates on the presumption that the values of neighboring picture pixels are almost identical. The average, lowest, and maximum of four nearby pixel values are used for pooling. An input image may usually be half its original size with A 2*2 filter. Before pooling, the input data may or may not be subjected to zero padding.

This sequential data transmission throughout the convolution and pooling layers is repeated by the CNN model. For instructional purposes, this procedure is run through two or four times. Subsequent convolution and pooling layer output is then processed using a multi-layer neural network. Here, a neuron unit acts as a feature map containing unit-specific information.

B. Fully Connected Neural Network

Each neuron in a fully connected layer of a neural network applies a linear adjustment to the input vector using a weights matrix. All possible layer-to-layer connections are therefore present, i.e., every input influences every output in turn in the output vector.

Neural networks are collections of dependent non-linear functions. Each unique function is carried out by a neuron. In fully connected layers, the neuron transforms the input vector linearly using a weights matrix. The result is then subjected to a non-linear transformation via a non-linear activation function (f). The activation function "f" wraps the dot product between the layer's input and weights matrix. Remember that each column in the weights matrix will have a different quantity and will be adjusted as the model is trained.

III. LITERATURE REVIEW

This section provides a detailed literature review of detection and classification using different machine learning and deep learning techniques. All papers have been published prior to 2022. For the purpose of locating pertinent research on ML and DL techniques for plant disease classification and detection. We searched utilizing a number of search engines, including Web of Science, ScienceDirect, Scholar, and Scopus. Terms like "deep learning," "machine learning," "classification," "disease detection," "healthy plant," and "diseased plant" were employed. All co-authors evaluated each paper's abstract to determine whether or not it belonged there and could be included in the final version.

Deep learning algorithms based on CNN have been examined. The authors built four CNN-based models—V4, VGG-16, ResNet-50, and DenseNet-121—for plant diseases using a pre-trained model that was learned using the transfer learning approach. DenseNet-121, which contains 121 deep layers, surpassed all other networks with 99.87% training accuracy and 99.81% testing accuracy, according to their suggested transfer learning approach.The Plant Village data set, which includes 54,305 Images of both healthy and diseased plants, was used to compare the precision of each technique. where 20% of the datasets were used for validation and 80% of the datasets were used for training. They have used the on-time image augmentation approach for preprocessing, which addresses the overfitting and space issues by producing temporary augmented pictures from the provided raw image during training. Image augmentation techniques have been used to make the training set more robust. [1].

With the use of deep learning, the authors have employed Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet) to precisely segment and classify plant leaf diseases. With improved efficiency, less overfitting, and increased generalization capabilities, the model architecture offers a potential solution for picture segmentation issues. After receiving a picture of a plant leaf as input, the resultant model segments the image and labels any sick spots [2].

This study investigates the use of pre-trained deep convolutional neural networks (CNNs) for this classification job using an open dataset with 52 categories representing different plant illnesses and healthy leaves.This work assessed the efficacy of EfficientNetB3-adaptive augmented deep learning (AADL) in conjunction with pre-trained deep CNN models, such as Xception, InceptionResNetV2, InceptionV3, and ResNet50, for accurate illness diagnosis. A performance evaluation was carried out by calculating the accuracy, precision, recall, and F1 score of the parameters, which included batch size, dropout, and epoch counts. With an astounding accuracy of 98.71%, the EfficientNetB3-AADL model surpassed the other models and traditional feature-based techniques [3].

Additionally, the researcher reviewed the use of deep learning and machine learning in the field of illness detection using a variety of data sets. They suggested doing a thorough analysis of five cutting-edge algorithms for the identification of plant diseases. According to computational studies, YOLOv5 has a high object detection accuracy[4].

A Deep Transfer Learning Approach for Pathogen-Based Classification of Plant Diseases. The EfficientNetV2B2 and EfficientNetV2B3 models are used for automated plant disease identification and classification. The pathogen causing the illness is identified through the use of transfer learning techniques. To address the problems with the Plant Village Dataset, such as uniform backdrops and controlled settings, the authors employed a new data set called Agri-ImageNet in addition to photographs of the leaves, bulbs, and flowers of sunflower and cauliflower taken in a natural realistic setting[5].

The Effective Identification of Maize Plant Leaf Diseases has been proposed using the MaizeNet Deep Learning Approach. The accurate location and categorization of several leaf diseases in maize crops was achieved with the application of MaizeNet. By employing the Faster-RCNN technique, which uses the ResNet-50 model with spatial-channel attention as its basis network for the calculation of deep keypoints that are then localized and classified into different classes, they have significantly enhanced performance. The study was evaluated using samples from three distinct classes of maize plant illnesses that were collected in a variety of settings, including complicated backgrounds, bright lights, and changes in color and size. The database used for the testing was called Corn Disease and Severity. With a mAP value of 0.94 and an average accuracy score of 97.89%, the MaizeNet model performs well. [6].

The crops of cotton and tomatoes have been utilized by writers to identify 12 prevalent plant diseases. A lightweight 2D CNN architecture based on deep learning has been suggested to categorize 12 illnesses, with 2 class labels indicating health. Despite having fewer parameters, the suggested architecture performed better than all of the pre-trained models, including VGG16, VGG19, and InceptionV3. Because the model is lightweight, the app performs very well, classifying the proper condition in a shorter amount of time—about 4.84 ms. [7].

The transformer embedded ResNet model was proposed by the author. In order for the model to extract the characteristics from the cassava leaves, the author uses this model to try to eliminate the noise from the leaves. Moreover, a FAMP-Softmax technique was put out to use the cassava leaf dataset to develop precise classification boundaries. By contrasting the output of Xception, VGG16 Inception-v3, ResNet-50, and DenseNet121, the author hopes to get accuracy gains of 3.05%, 2.62%, 3.13%, 2.12%, and 2.62% in terms of performance [8].

The Plant Village dataset, which has 50,000 photos of 14 distinct crops, was used by the author. Various pre-trained models were employed for the categorization based on the various corps photos. Additionally, a CNN technique with three convolution and three max pooling layers with two fully linked layers is used for the identification of nine tomato leaf diseases. The suggested model's average accuracy is 91.2%, with 10 classes. Classification accuracy ranges from 76% to 100% on various classes [9].

For the detection, the author employed deep learning algorithms. The performance of the ResNet50 architecture, which the authors utilized to identify potato, tomato, and corn diseases, is 98.7% for plant diseases [10].

The approach of transfer learning is investigated by the researcher in order to identify and categorize sunflower seeds. The author created a hybrid model by stacking the ensemble results of Pre-Trained VGG16 and MobileNet architectures for classification. [11].

The fully connected CNN approach was employed by the author to categorize rice crops. In order to reduce background noise from the picture during image binarization, the model will employ Otsu's global thresholding approach[12].

The author uses the transfer learning approach and residual network (ResNet). Beginning In addition to using ResNet architecture on the Palm Leaf dataset, which has 2631 colored pictures of various sizes, the author additionally applied augmentation techniques to the dataset in order to train and test models that reach 99.62% accuracy [13].

IV. ANALYSIS OF EXISTING WORK

The preceding sections include a wealth of study about the current system. The bulk of studies have enhanced the accuracy of thiss detecting system by fusing state-of-the-art designs. An examination of some studies conducted for the dieses detection system is shown in Table 1 below.

Table 1 illustrates how we have considered the publications that addressed various plant diseases affecting several plants. The dataset utilized in the corresponding papers and the type of disease addressed have also been mentioned. Plant Village, a dataset comprising over 65,000 samples of various plants, is the most commonly used dataset. The researchers' chosen architectures have also been considered, and their performance has been mentioned.

Ref.	Crop Focus	Disease Addressed	Dataset	Classes	Model	Model Performance
[1]	14	38	Plant Village	38	DenseNet121 ResNet50 VGG16 InceptionV4	99.81 98.73 82.75 97.59
[2]	Several	Several	Kaggle	58	Inception V3 InceptionReNetV2 ResNet50 Xception EfficientNetB3AADL	97.08 97.74 95.94 98.50 98.71

[3]	39	40	Plant Village Coffee Leaf Custom dataset	40	Cascading Autoencoder With attention Residual U-Net (CAAR-UNet)	95.26
[4]	Several	Several	PlantDoc	Various	Yolov5	High Accuracy
[5]	Sun Flower Cauli Flower	Several	Sunflower Cauliflower AgriImageNet	Varous	EfficientNetV2B2 EfficientNetV2B3	96.82 97.35
[6]	Maize	3	Corn Disease and Severity	3	MaizeNet	97.89
[7]	Tomato	14	Plant Village	14	2D CNN Architecture	97.36
[8]	Cassava	5	Kaggle	5	T-RNet(Transformer embedded Res-Net)	90.63
[9]	Tomato	9	Plant Village	9	Mobilenet VGG-16 InceptionV3 Proposed Model	63.75 77.2 63.4 91.2
[10]	Potato Tomato Corn	16	Plant Village	16	ResNet50	98.7
[11]	Sunflower	4	Google Image	4	VGG-16 MobileNet	81 86
[12]	Rice	3	Kaggle	3	CNN	99.7
[13]	Palm	3	Kaggle	3	ResNet	99.62

Table 1: Analysis of Existing work

V. DEEP LEARNING ARCHITURES FOR IMAGE CLASSIFICATION AND DETECTION

A. DensNet-121

In a traditional feed-forward convolutional neural network (CNN), each convolutional layer (apart from the first) takes input, processes the previous convolutional layer's output, and creates an output feature map that is passed on to the convolutional layer after it. Because of this, each layer and the one following it have 'L' direct connections for 'L' layers. However, as the number of layers in the CNN increases, or as the layers go deeper, the "vanishing gradient" problem manifests itself. As a result, more data may "vanish" or be lost during the transfer from the input to the output layers, which reduces the network's ability to train effectively.

DenseNets mitigate this problem by modifying the traditional CNN architecture and simplifying the layer-to-layer link topology. The design known as DenseNet, in which each layer is directly connected to every other layer, is known as a densely linked convolutional network. For 'L' layers, there are L(L+1)/2 direct connections.

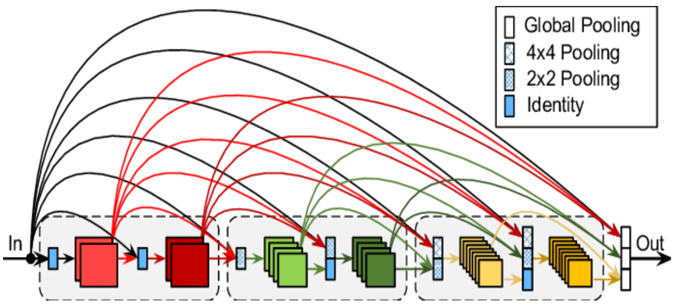

Fig. 2. DensNet121 Architecture

DenseNet-121 comprises the following layers:
Layer 1: 1 7x7 Convolution
Layer 2: 58 3x3 Convolution
Layuer 3: 61 1x1 Convolution
Layer 4: 4 AvgPool
Layer 5: 1 Fully Connected Layer
Layer 6 : DenseNet's components consist of Connectivity:
Rather of being summed, the feature maps from each previous layer are concatenated and used as inputs in each subsequent layer. DenseNets require fewer parameters than a corresponding regular CNN since redundant feature maps are eliminated, allowing for feature reuse. Thus, the feature-maps of all previous layers, x0,...,xl-1, are sent into the lth layer as input:
Xl = Hl ([x0, x1,….,xl-1])
where [x0,x1,...,xl-1] represents the feature-map concatenation, or the output generated in each of the layers that came before l (0,...,l-1). To make implementation easier, Hl's many inputs are concatenated into a single tensor.

DenseBlocks
The concatenation approach is impractical when feature map sizes differ. However, while this reduces the dimensionality of feature maps, downsampling layers which do this are an essential part of CNNs because they enable quicker processing.

To make this easier, DenseNets are divided into DenseBlocks; inside a block, the number of filters that divide the feature maps fluctuates, but the feature map sizes remain constant. The layers that sit between the blocks are known as the "Transition Layers," and they cut the number of channels in half.

Growth Rate
Think of the attributes as the general state of the network. The feature map grows as one navigates through each thick layer because each layer contributes 'K' features (pre-existing features) to the global state. The network growth rate, or parameter 'K,' determines the amount of data that is added to each layer of the network. If each function H l creates a feature map, the lth layer has k feature maps.
kl = k0 + k * (l-1)
Bottleneck Layers
Even while each layer only produces k feature-maps as an output, there could be many inputs—particularly for layers that follow—in this hierarchy. Before each 3x3 convolution, a 1x1 convolution layer may be inserted as a bottleneck layer to improve processing speed and efficiency.

B. ResNet 50
Training a strong image classification model, like ResNet50, on large datasets can yield state-of-the-art results. The use of residual connections, which allow the network to take up a set of residual functions that convert the input into the desired output, is a key achievement in it. These remaining connections allow the network to learn more deeper structures without addressing the problem of vanishing gradients.

In the convolutional layers of ResNet50, batch normalization and ReLU activation come after many convolutional layers. These layers are responsible for extracting features from the input image, such as shapes, edges, and textures. Convolutional layers are followed by max pooling layers, which reduce the spatial dimensions of the feature maps while preserving the most important characteristics.
The identification block and the convolutional block are the two primary ResNet50 building blocks. The identity block is a simple block that, after going through several convolutional layers, puts the input back to the output. Consequently, residual functions, which translate input into desired output, may be learned by the network. The convolutional block resembles the identity block because a 1x1 convolutional layer was added to reduce the number of filters before the 3x3 convolutional layer.

The 50-layer ResNet's construction block is intended to function as a bottleneck. A bottleneck residual block, also called a "bottleneck," uses 1×1 convolutions to reduce the number of parameters and matrix multiplications. This significantly speeds up the training of each layer. It uses a stack of three layers in instead of only two.

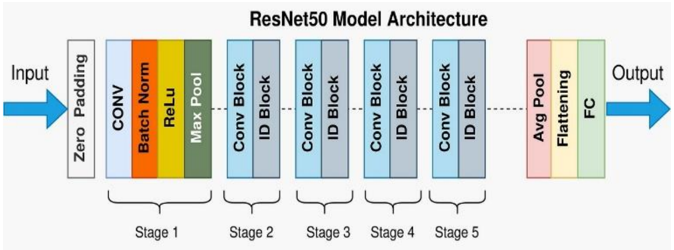

Fig. 3. ResNet50 Architecture

The following are the elements that make up the 50-layer ResNet architecture:

• 7×7 kernel convolution alongside 64 other kernels with a 2-sized stride.

• max pooling layer with a 2-sized stride.

• 9 more layers—3×3,64 kernel convolution, another with 1×1,64 kernels, and a third with 1×1,256 kernels. These 3 layers are repeated 3 times.

• 12 more layers with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels, iterated 4 times.

• 18 more layers with 1×1,256 cores, and 2 cores 3×3,256 and 1×1,1024, iterated 6 times.

• 9 more layers with 1×1,512 cores, 3×3,512 cores, and 1×1,2048 cores iterated 3 times.

Average pooling, followed by a fully connected layer with 1000 nodes, using the softmax activation function.

C. VGG 16

One kind of artificial neural network is called a convolutional neural network, or ConvNet. A convolutional neural network has an input layer, an output layer, and several hidden layers. The CNN (Convolutional Neural Network) variation known as VGG16 is one of the best computer vision models on the market right now. The model's creators studied the networks and enhanced the depth using an architecture with relatively small (3 × 3) convolution filters, showing a significant improvement over the state-of-the-art setups. They raised the depth to 138 trainable parameters, or around 16–19 weight layers.

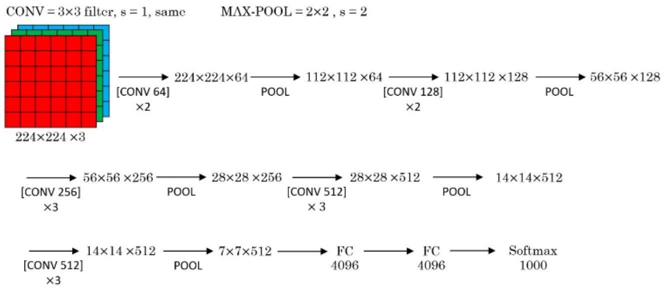

Fig. 4. VGG16 Architecture

The object identification and classification algorithm VGG16 is able to classify 1000 pictures into 1000 different categories with an accuracy rate of 92.7%. With transfer learning, this popular picture categorization method is easy to use.
VGG16 stands for sixteen layers that are weighted. VGG16 only has sixteen weight layers, or learnable parameter levels, out of a total of twenty-one layers—

sixteen convolutional layers, five Max Pooling layers, and three Dense layers.

D. AlexNet

AlexNet was created by Hinton student Alex Krizhevsky, the 2012 ImageNet competition winner. Deeper neural networks, such as the amazing vgg, GoogleLeNet, were also introduced following that year. Its official data model has an accuracy percentage of 57.1%, with the top 1-5 attaining 80.2%. This is already really impressive—even for traditional machine learning classification algorithms.

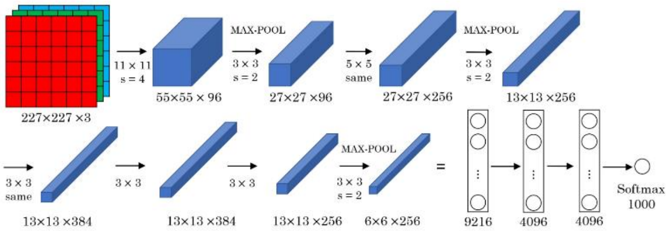

Fig. 5. AlexNet Architecture

GPU was employed by the AlexNet architecture to improve training efficiency. AlexNet is composed of two normalized layers, two completely connected layers, three max-pooling levels, five convolution layers, and one softmax layer. Each convolution layer is composed of a non-linear activation function called "ReLU" and a convolution filter. The input size is fixed and the max-pooling function is used since the pooling layers have totally connected layers.

Despite the fact that the input size is often specified as 224x224x3, some padding causes it to really measure 227x227x3. Above all, AlexNet has more than 60 million parameters.

Important Features:

"ReLU" is utilized as an activation function instead of "tanh."

AlexNet demonstrates that saturating activation functions like as Tanh or Sigmoid allow deep CNNs to be trained much more quickly. ReLUs can help AlexNet achieve a 25% training error rate, as seen by the solid curve in the following figure. Compared to a network that employs tanh (dotted curve), this is six times faster. The CIFAR-10 dataset was utilized for the assessment.

Dropout is another often-mentioned concept that effectively prevents neural networks from overfitting. Rather of using the general linear model, a regular technique is used to prevent the model from overfitting. By altering the neural network's design, dropout is implemented in the network.

After randomly deleting a specific number of neurons with a given probability and keeping the same number of neurons in the input and output layers, update the parameters in line with the neural network's learning approach. The next time, choose a new random Till the training is complete, eliminate a few neurons.

Techniques for enhancing data include color normalization, jittering, flipping, and cropping. In order to generate a batch of "new" data from the existing data, data augmentation additionally uses translation and noise to artificially increase the size of the training set.

E. Ensemble Learning Techniques

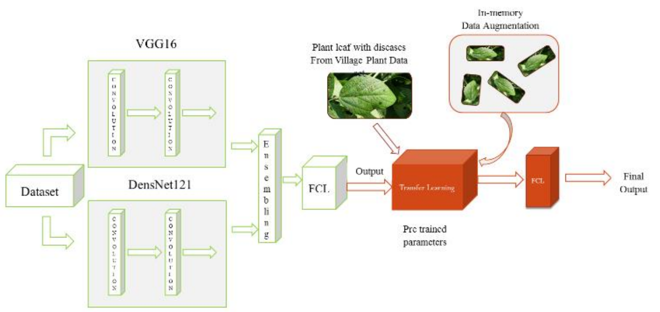

Fig. 6. Proposed System

In this proposed model we have used transfer learning based on ensembling of two famous network DensNet121 and VGG16 has been used and there accurecy has been averaged and average of there accurecy has been considered in that ensemble model has been attached with fully connected layers within memory data augmentation technique through this ensemble mechanism we are able to achieve 99.10%

and the graph of loss and accurecy over a 50 epoch has been shown in figure.

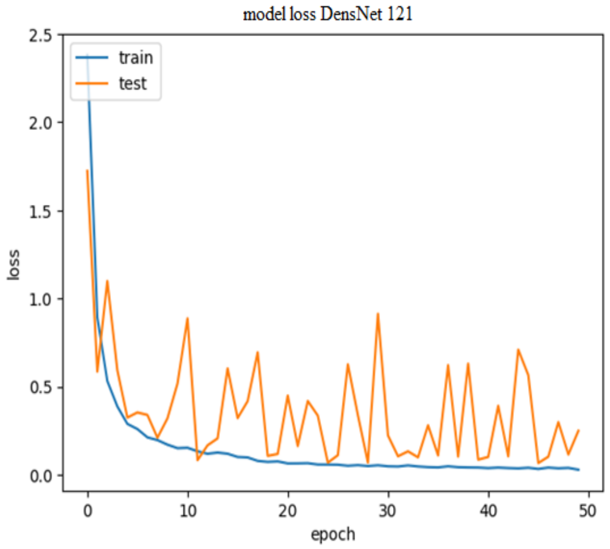

Fig. 7. Model Loss DensNet121 with 50 epoch

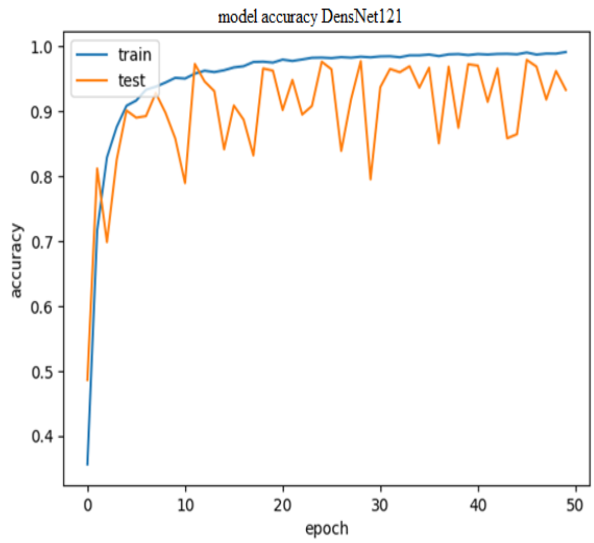

Fig. 8. Model Accuracy DensNet121 with 50 epoch

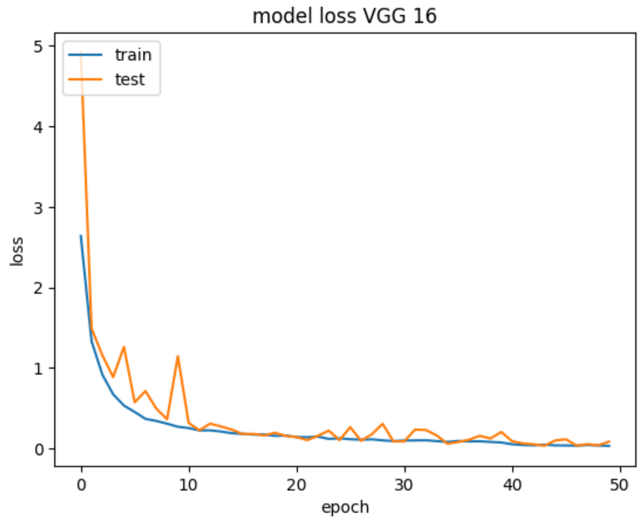

Fig. 9. Model Loss VGG16 with 50 epoch

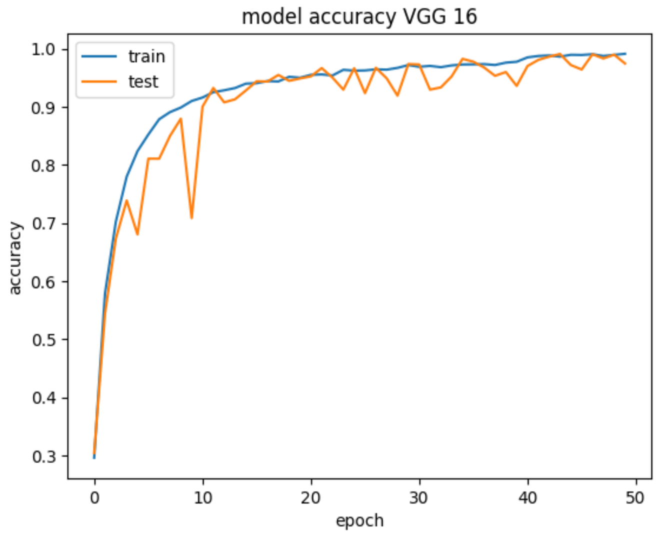

Fig. 10. Model Accuracy VGG16 with 50 epoch

VI. CONCLUSION

Deep learning architectures for this kind of detection have shown to be a potential method for improving detection accuracy. Different writers have put forth separate or combined methods for classifying and identifying plant sieves. According to the study of the current system, we might use architecture during the training phase to get improved performance. But a lot of problems usually call for deep neural network optimization throughout the validation and testing stages.

VII. REFERENCES

[1]. Eunice, J., Popescu, D. E., Chowdary, M. K., & Hemanth, J. (2022). Deep learning-based leaf disease detection in crops using images for agricultural applications. Agronomy, 12(10), 2395.

[2]. Adnan, F., Awan, M. J., Mahmoud, A., Nobanee, H., Yasin, A., & Zain, A. M. (2023). EfficientNetB3-Adaptive Augmented Deep Learning (AADL) for Multi-Class Plant Disease Classification. IEEE Access.

[3]. Abinaya, S., Kumar, K. U., & Alphonse, A. S. (2023). Cascading Autoencoder with Attention Residual U-Net for Multi-Class Plant Leaf Disease Segmentation and Classification. IEEE Access.

[4]. Balafas, V., Karantoumanis, E., Louta, M., & Ploskas, N. (2023). Machine learning and deep learning for plant disease classification and detection. IEEE Access.

[5]. Rani, K. A., & Gowrishankar, S. (2023). Pathogen-based Classification of Plant Diseases: A Deep Transfer Learning Approach for Intelligent Support Systems. IEEE Access.

[6]. Masood, M., Nawaz, M., Nazir, T., Javed, A., Alkanhel, R., Elmannai, H., ... & Bourouis, S. (2023). MaizeNet: A deep learning approach for effective recognition of maize plant leaf diseases. IEEE Access.

[7]. Peyal, H. I., Nahiduzzaman, M., Pramanik, M. A. H., Syfullah, M. K., Shahriar, S. M., Sultana, A., ... & Chowdhury, M. E. (2023). Plant Disease Classifier: Detection of Dual-Crop Diseases using Lightweight 2D CNN Architecture. IEEE Access.

[8]. Zhong, Y., Huang, B., & Tang, C. (2022). Classification of cassava leaf disease based on a non-balanced dataset using transformer-embedded ResNet. Agriculture, 12(9), 1360.

[9]. Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Computer Science, 167, 293-301.

[10]. Arshad, M. S., Rehman, U. A., & Fraz, M. M. (2021, May). Plant disease identification using transfer learning. In 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2) (pp. 1-5). IEEE.

[11]. Malik, A., Vaidya, G., Jagota, V., Eswaran, S., Sirohi, A., Batra, I., ... & Asenso, E. (2022). Design and evaluation of a hybrid technique for detecting sunflower leaf disease using deep learning approach. Journal of Food Quality, 2022, 1-12.

[12]. Upadhyay, S. K., & Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. International Journal of Information Technology, 1-15.

[13]. Ahmed, M., & Ahmed, A. (2023). Palm tree disease detection and classification using residual network and transfer learning of inception ResNet. Plos one, 18(3), e0282250.