A Comparative Analysis of CNN Models in Deep Learning for Leaf Disease Detection

R. Jayamma[1], R. Hema Chandrika[2], Ch. Devi Durga[3]

[1]Assistant Professor, Department of Computer Science, P. B. Siddhartha College of Arts & Science, Vijayawada, Andhra Pradesh, India.
[2,3]Student, II MCA P. B. Siddhartha College of Arts & Science, Vijayawada, Andhra Pradesh, India.

Abstract:
There is a huge growth in the population of India therefore there is necessity for more food. But in farming the unavoidable plant diseases causing a large problem. There is a need to figure out as to how much food produced by the farmers is affected. Because in the coming future a greater amount of people are to be fed. Plant leaf disease detection is very important, because depending on the amount of growth in crop the farmers make their money. Then here comes CNN in help. It is tool which is smart enough to identify the diseases and their types. In order to detect the disease in plant a Convolutional Neural Network(CNN) with the help of image processing beside is in use here in our paper. A Convolutional Neural Network is an artificial neural network which is specially designed to deal with image recognition[1] tasks when an image is input. Here the idea is to use CNN models to spot diseases in apple, grape, corn and potato. This idea is to use CNN models to spot diseases in apple, grape, corn, and potato plants. We proposed an algorithm. This paper mainly focused on CNN models **CNN, AlexNet,VGG16** in deep learning that will be compared in the study.

Keywords: Image classification, Deep Learning, leaf disease, Convolutional Neural Network, Alex Net, VGG16.

Introduction:

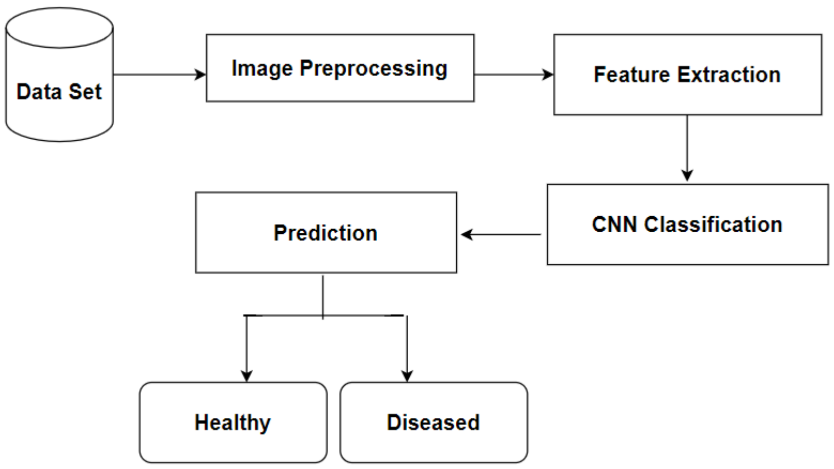
The main food source of India is crop production. To improve the productivity of crops we are taking the help of advanced technology. Many varieties of plants and crops are cultivated by Indian farmers. Therefore a lot of research is been centred to search ways to cultivate more food which is healthy. In the present scenario depending completely on human speculation to detect efficiently the diseases in plant is not a good choice. Yet the modern expansion in computer vision provides the solutions to issues faced with plant and leaf which is rapid, consistent and accurate.
In these recent years an outstanding amount of research has been done in deep learning in the fields like image recognition, sentiment analysis and speech recognition. Therefore convolutional neural network will be most effective means to detect diseases in leaf and plants and solve the problem raised[2].
An algorithm is proposed for the leaf disease detection. We will be importing libraries needed in the initial stage: OS, Tensorflow, pandas, Matplotlib, Cv2,Keras, random, NumPy pandas, etc. Neat the function is defined to label images and to load the training data. The images are been categorized based on the plant disease's code names. The resizing and is done to a group of random images while training the images and matching labels are added consequently.

Leaf images are added to do the testing and a CNN algorithm to add series of layers is for classification. These layers are like convolutional and pooling layers used for process optimization in every phase. And the required output is obtained from the dense layer. The learning rate is used to affect the rate of our models learning process. Later the data is been loaded to our built model, to designate the nature of leaf a healthy or diseased using a variable and afterwards the in this variable the model is saved. And this variable is used for the detection.

LITERATURE SURVEY

Nishant Shelar and colleagues discovered spotinfections in leaves and categorized them according to thediseased leaf categories using various learning algorithms . Network is to acquire and analyze data from leaf photosin order to determine healthy or diseased leaves ofmedical plants using image processing methods.[1]
The paper introduces by sumit and colleagues a novel approach to image recognition by utilizing deep learning techniques. The researchers investigated three distinct neural network architectures: Faster R-CNN, R-CNN, and SSD. Their efforts resulted in a commendable validation accuracy of 94.6%. This proposed method is capable of identifying a range of diseases affecting leaves from apple, cherry, grape, peach, pepper, potato, strawberry, and tomato plants.[2]
S.Bharath , K.Vishal Kumar , R.Pavithram, T.Malathi , "Detection of Plant Leaf Disease using CNN ". Image processing and CNN model can be used to improve plant disease detection techniques. It consisting data of 38 different plant leaf diseases which used to predict.The conclusion of this paper is to predict the pattern of plant diesease using CNN.[3]
The investigation led by Murk Chohan and colleagues was centred on the identification of diseases in plant leaves. To enhance the dataset's sample size, they implemented augmentation techniques. For testing purposes, 15% of the data from the Plant Village dataset was employed. [4]
Ali Arshagi and colleagues undertook a study in this paper, focusing on the analysis of five distinct categories of potato diseases, including Healthy, Black Scurf, Common scab, Black leg, and Pink dot. The researchers proceeded to evaluate the outcomes produced by various disease classification methods, including AlexNet, GoogLeNet, VGG, and R-CNN.[5]

PROPOSED METHODOLOGY:



Data Set:

The labelled PlantVillage dataset available on kaggle comprises a collection of leaf images that encompass both healthy and diseased conditions. This dataset is structured into eight distinct categories, with all images uniformly resized to dimensions of 256 by 256 pixels.

Leaf Categories:

Potato Healthy

Potato Early blight

Healthy Grape

Grape Black rot

Healthy Corn

Corn Common rust

Healthy Apple

Apple Black rot

Image Pre-processing:

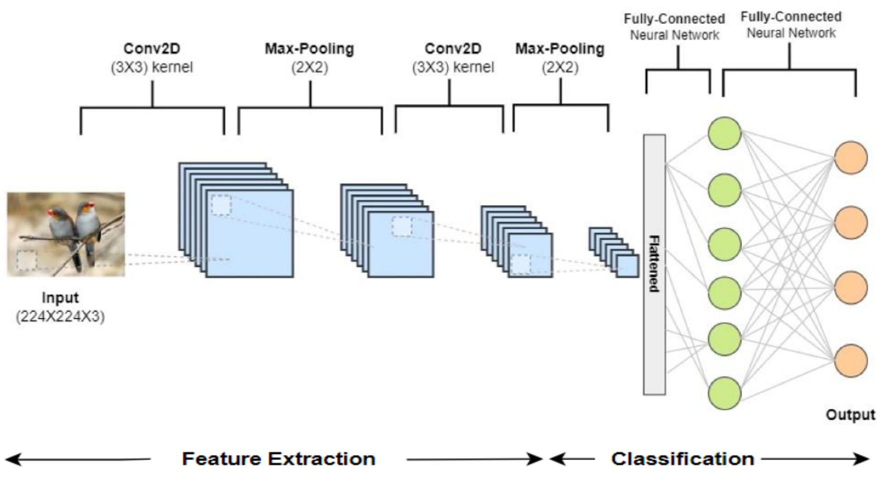
We cannot feed the image data directly to our neural network. To prepare image data for input into a neural network, a series of pre-processing steps are required. This involves converting the images into a format that the network can understand, which often means transforming them into NumPy arrays.

Feature Extraction:

In Convolutional Neural Networks (CNNs), filters are employed to extract features from images. The remarkable aspect is that during training, the network autonomously detects these filters, determining both their quantities and parameter values. This intrinsic learning process adjusts filter amounts and content. As a hyperparameter, you prescribe the desired quantity of filters and their dimensions, influencing the architecture's capability to discern intricate image characteristics.

CNN Classification:

The CNN classifier for image classification is built using a special kind of neural network called CNN. Its main job is to look at pictures and decide which category they belong to. It learns to pick out important parts from the pictures and use them to figure out the right category, like telling what's in a photo. When you give the network a picture in the form of numbers (numpy array), it gives you a number between 0 and 1 that tells you how sure it is about its decision.



Prediction:
Predicting leaf disease involves employing a Convolutional Neural Network (CNN) model. This model is utilized to distinguish whether a user-provided input leaf image is in a healthy state or exhibiting signs of disease. The CNN model's capability lies in its capacity to differentiate between healthy leaves and those afflicted by various types of diseases.

AlexNet:
CNN model AlexNet was chosen for leaf disease detection. There are over 1.2 million training images, 50,000 validation images, and 150,000 testing images. By removing the central 256×256 patch from each image, the model builders imposed a fixed size of 256×256 pixels.AlexNet's architecture consists of 8 convolutional layers, of which 5 convolutional layers and three ANN layers. Each of the convolution layers is followed by a max pooling layer. Its architecture is easy to understand. It uses ReLu activation and involves overlapping max pooling. The following image shows the architecture of AlexNet.

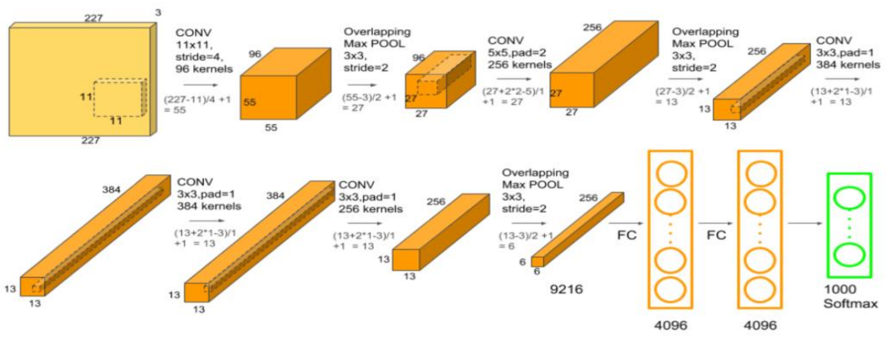


Figure (ref. shutterstock)

VGG16:
The VGG16 model attains an impressive test accuracy of 92.7% on ImageNet, an extensive dataset encompassing over 14 million training images distributed among 1000 distinct object classes. VGG16 represents an advancement over the AlexNet architecture by adopting a strategy of substituting the large filters with a series of smaller 3×3 filters. Unlike AlexNet, where the kernel size is 11 for the initial convolutional layer and 5 for the subsequent layer, VGG16 utilizes a consistent kernel size of 3×3

throughout its layers. The research team dedicated several weeks to the training of the VGG model, employing NVIDIA Titan Black GPUs to execute this computational task.

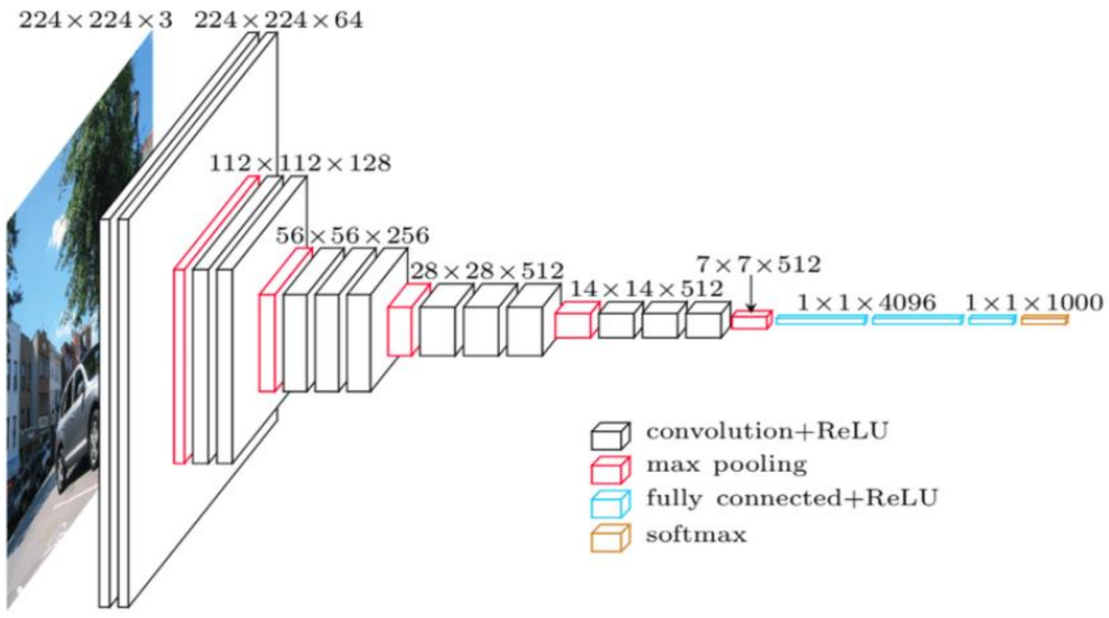
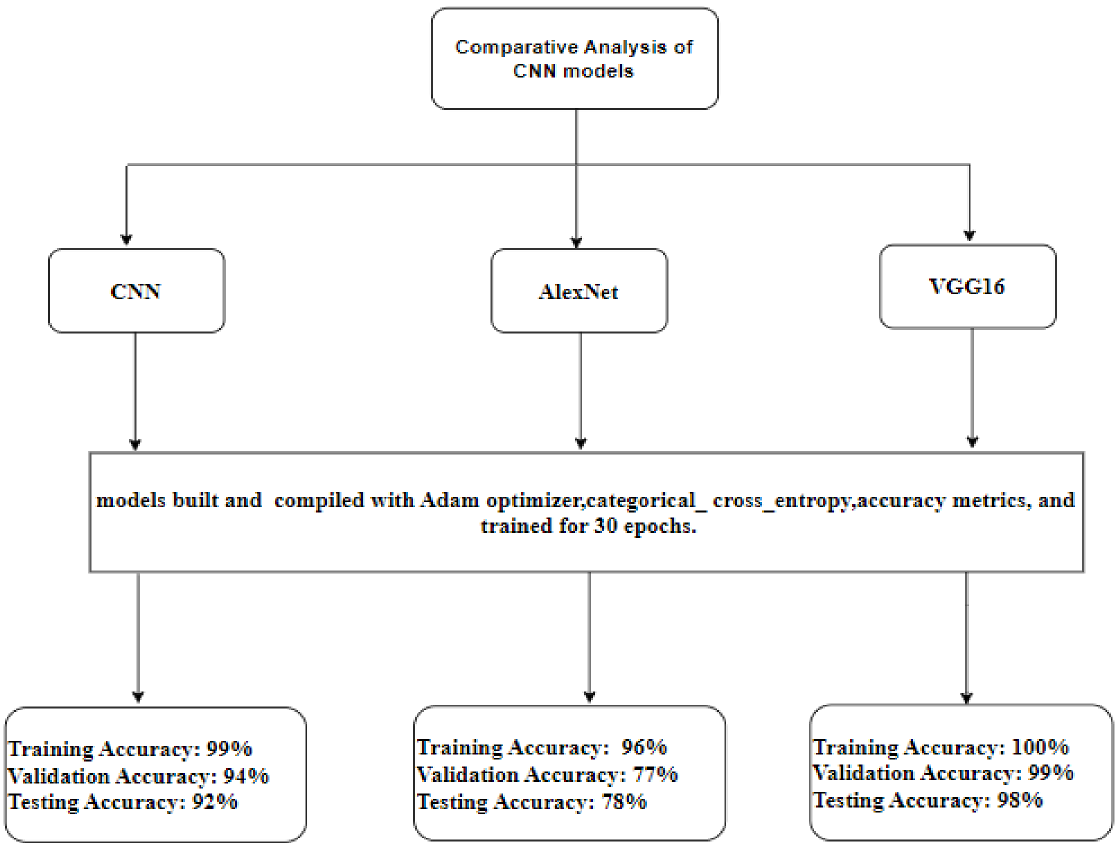
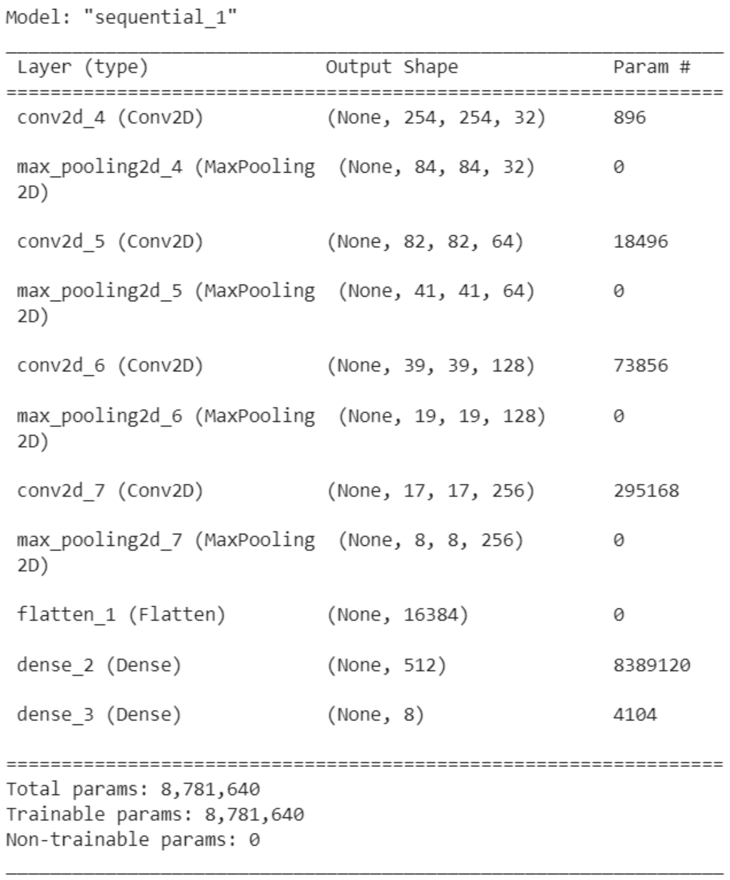


Figure (i2 tutorials)

Block diagram of the results of Proposed work:

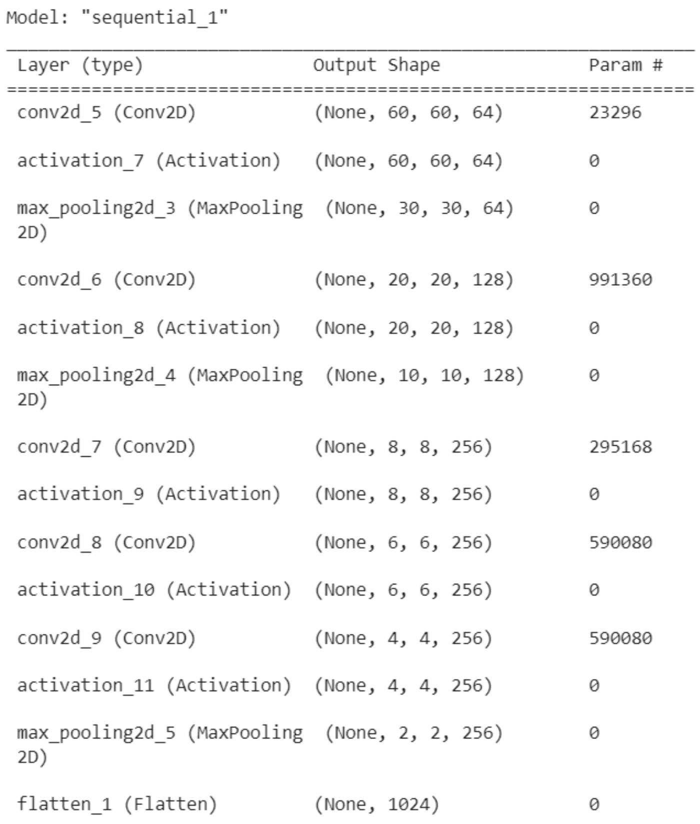


Model Summary Of CNN:

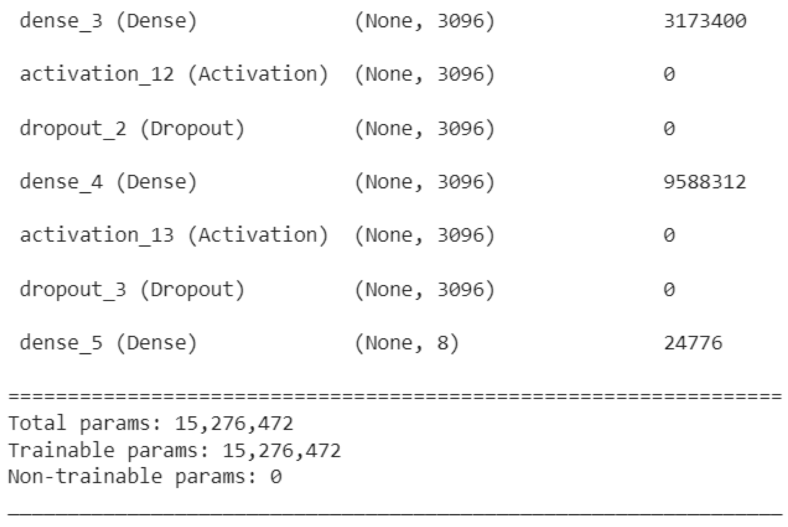


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 254, 254, 32)      896

 max_pooling2d_4 (MaxPooling  (None, 84, 84, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 82, 82, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 41, 41, 64)       0
 2D)

 conv2d_6 (Conv2D)           (None, 39, 39, 128)       73856

 max_pooling2d_6 (MaxPooling  (None, 19, 19, 128)      0
 2D)

 conv2d_7 (Conv2D)           (None, 17, 17, 256)       295168

 max_pooling2d_7 (MaxPooling  (None, 8, 8, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 16384)             0

 dense_2 (Dense)             (None, 512)               8389120

 dense_3 (Dense)             (None, 8)                 4104

=================================================================
Total params: 8,781,640
Trainable params: 8,781,640
Non-trainable params: 0
_________________________________________________________________
```

Model Summary Of AlexNet:

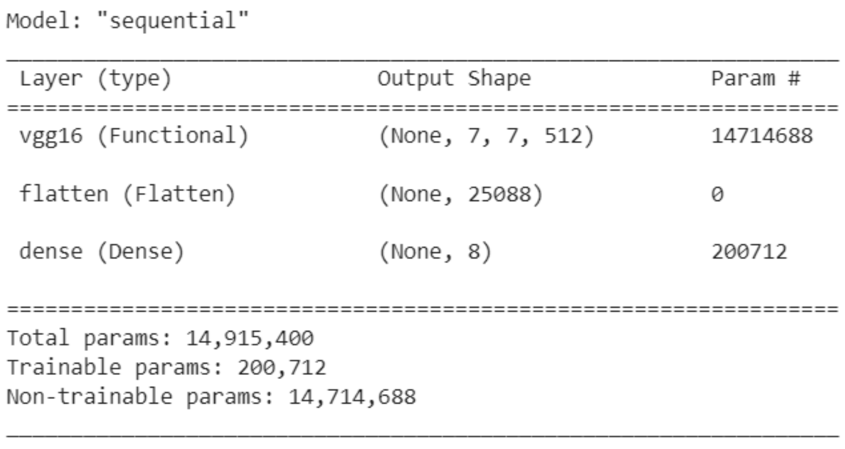


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 60, 60, 64)        23296

 activation_7 (Activation)   (None, 60, 60, 64)        0

 max_pooling2d_3 (MaxPooling  (None, 30, 30, 64)       0
 2D)

 conv2d_6 (Conv2D)           (None, 20, 20, 128)       991360

 activation_8 (Activation)   (None, 20, 20, 128)       0

 max_pooling2d_4 (MaxPooling  (None, 10, 10, 128)      0
 2D)

 conv2d_7 (Conv2D)           (None, 8, 8, 256)         295168

 activation_9 (Activation)   (None, 8, 8, 256)         0

 conv2d_8 (Conv2D)           (None, 6, 6, 256)         590080

 activation_10 (Activation)  (None, 6, 6, 256)         0

 conv2d_9 (Conv2D)           (None, 4, 4, 256)         590080

 activation_11 (Activation)  (None, 4, 4, 256)         0

 max_pooling2d_5 (MaxPooling  (None, 2, 2, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 1024)              0
```



```
 dense_3 (Dense)             (None, 3096)              3173400

 activation_12 (Activation)  (None, 3096)              0

 dropout_2 (Dropout)         (None, 3096)              0

 dense_4 (Dense)             (None, 3096)              9588312

 activation_13 (Activation)  (None, 3096)              0

 dropout_3 (Dropout)         (None, 3096)              0

 dense_5 (Dense)             (None, 8)                 24776

=================================================================
Total params: 15,276,472
Trainable params: 15,276,472
Non-trainable params: 0
_________________________________________________________________
```

Model Summary of VGG16:



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 8)                 200712

=================================================================
Total params: 14,915,400
Trainable params: 200,712
Non-trainable params: 14,714,688
_________________________________________________________________
```

PROPOSED ALGORITHM:

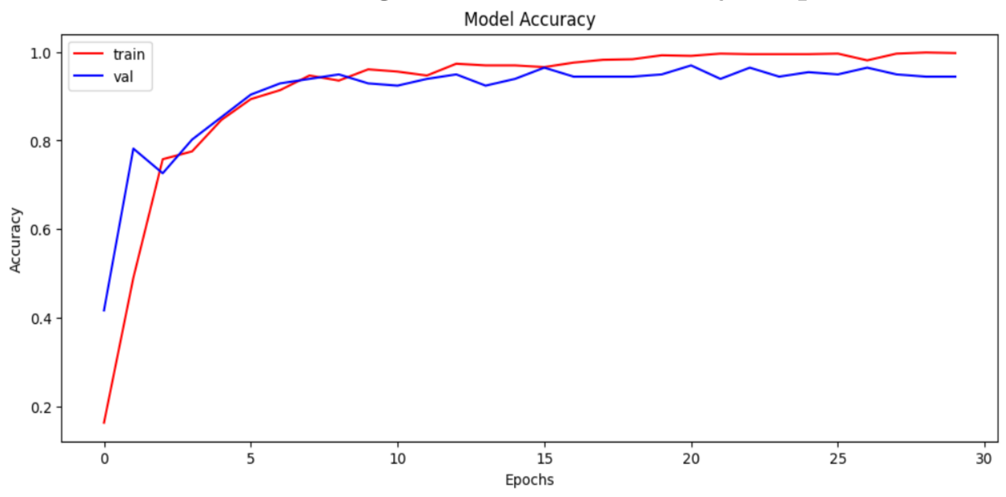
1. Begin by importing the necessary libraries, laying the foundation for your project's development.
2. Acquire the image dataset, a critical step in obtaining the data for your project.
3. Convert the images into arrays, an essential process for transforming visual data into a machine-understandable format.
4. Split the dataset into training and testing portions, preparing it for effective model training and evaluation.
5. Apply normalization to the split images, enhancing model performance and convergence.
6. Perform one-hot encoding, enabling the model to comprehend categorical labels efficiently.
7. Construct a Convolutional Neural Network (CNN) model to extract intricate features from the images.
8. To compile the model, employ the categorical cross-entropy as the loss function, accuracy as the chosen metric, and utilize the Adam optimizer.
9. Train the CNN over 30 epochs, refining its ability to identify health conditions from diseased ones.
10. Utilize the trained model to predict whether an uploaded image signifies health or disease, enabling effective diagnosis.

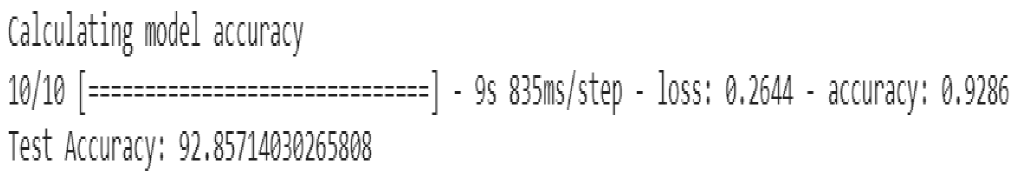
RESULTS:

After training the models for 30 epochs, we obtained high training accuracy results: 99% for CNN, 96% for AlexNet, and 100% for VGG16. The validation accuracy, though slightly lower, remained strong: 94% for CNN, 77% for AlexNet, and 99% for VGG16. In the testing phase, the models performed well, achieving an accuracy of 92% for CNN, 78% for AlexNet, and 98% for VGG16. You can see the graphical representation of the training and validation accuracy of CNN models in the figures below.

By clicking on the "Browse Files" button, you can upload images of leaves. In this case, we uploaded pictures of corn and grape leaves. The model will use these images to determine if the corn and grape leaves are healthy or diseased, and provide corresponding predictions.The predictions shown in the below figures.

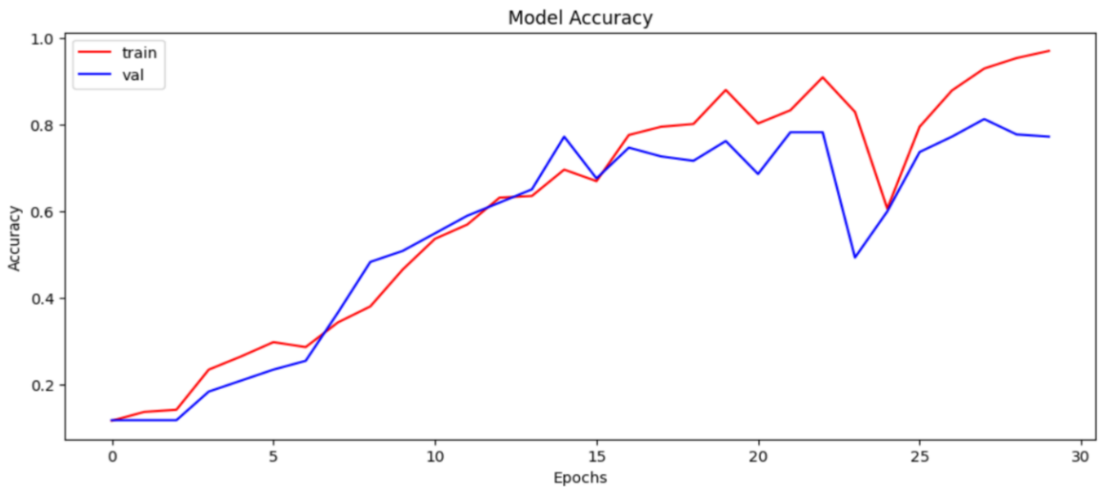
CNN Training and validation Accuracy Graph

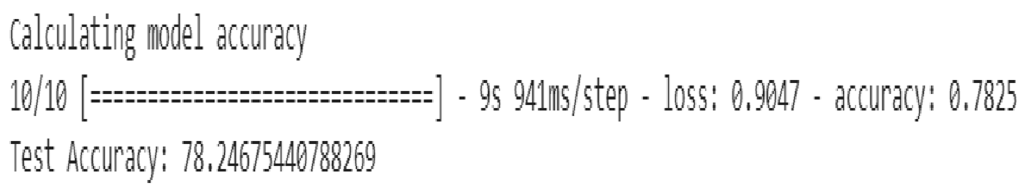




```
Calculating model accuracy
10/10 [==============================] - 9s 835ms/step - loss: 0.2644 - accuracy: 0.9286
Test Accuracy: 92.85714030265808
```

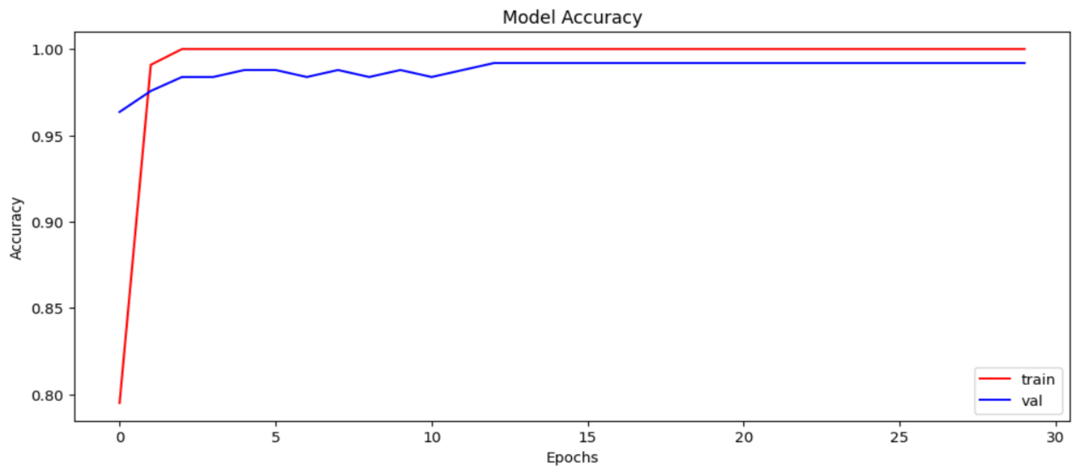
AlexNet Training and Validation Accuracy Graph

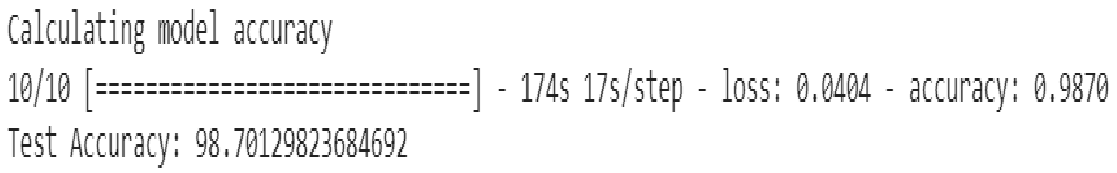




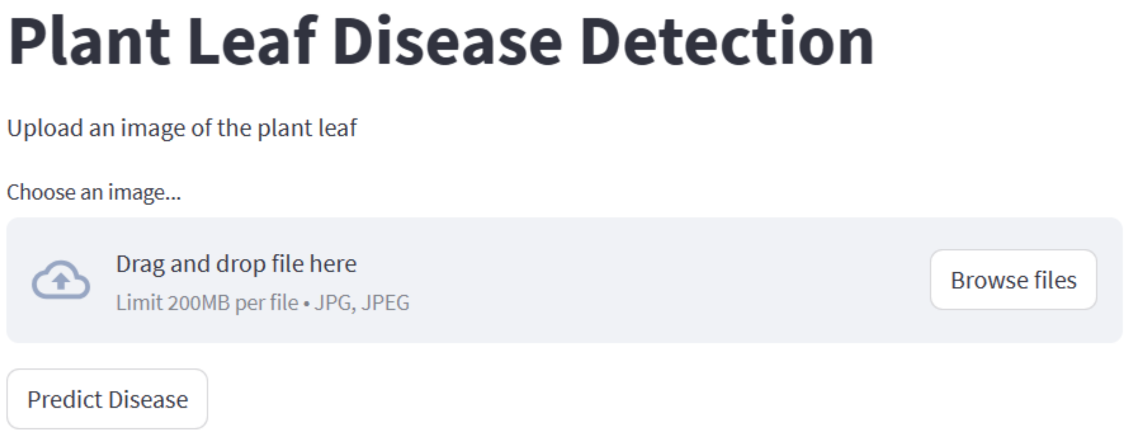
```
Calculating model accuracy
10/10 [==============================] - 9s 941ms/step - loss: 0.9047 - accuracy: 0.7825
Test Accuracy: 78.24675440788269
```

VGG16 Training and Validation Accuracy Graph





```
Calculating model accuracy
10/10 [==============================] - 174s 17s/step - loss: 0.0404 - accuracy: 0.9870
Test Accuracy: 98.70129823684692
```



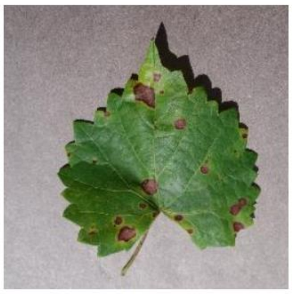
Plant Leaf Disease Detection

Upload an image of the plant leaf

Choose an image...

```
(256, 256, 3)
```

This is Corn_(maize) leaf with _healthy



```
(256, 256, 3)
```

This is Grape leaf with _Black_rot

```
(256, 256, 3)
```

This is Potato leaf with _healthy

```
(256, 256, 3)
```

This is Corn_(maize) leaf with *Common_rust*

CONCLUSION:

The Convolutional Neural Network (CNN) models were compiled using the Adam optimizer, which is known for its efficient optimization capabilities. For measuring the model's performance, the categorical cross-entropy loss function was utilized. This loss function is commonly employed in multi-class classification tasks. Additionally, accuracy was chosen as the metric to assess how well the models were able to classify the input data into their respective categories. The uploaded images of leaves will be predicted by neural network model as one of the 8 categories. In our observation the model gives the following results: among the three models, VGG16 demonstrated strong performance, achieving an impressive accuracy of 98% during the testing phase. This result indicates that the VGG16 model was effective in accurately classifying the test data.

REFERENCES:

1. Shelar, N., Shinde, S., Sawant, S., Dhumal, S., & Fakir, K. (2022). Plant Disease Detection Using Cnn.In ITM Web of Conferences (Vol. 44, p. 03049).EDP Sciences.
2. Kumar, S. (2021). Plant disease detection using CNN. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 2106-2112.
3. Bharath, S., Kumar, K. V., Pavithran, R., & Malathi, T. (2020). Detection of plant leaf diseases using CNN.International Research Journal of Engineering and Technology (IRJET), 4Computer Science and Engineering, SRM Institute of Science and Technology, Chennai. India.
4. Chohan, M., Khan, A., Chohan, R., Katpar, S. H., &Mahar, M. S. (2020). Plant disease detection using deep learning. International Journal of Recent Technology and Engineering, 9(1), 909-914.
5. Arshaghi, A., Ashourian, M., &Ghabeli, L. (2023). Potato diseases detection and classification using deep learning methods.Multimedia Tools and Applications, 82(4), 5725-5742.