PROJECT REPORT
ON

PLANT DISEASES CLASSIFICATION USING MACHINE LEARNING

Carried Out at

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING
ELECTRONIC CITY, BANGALORE

UNDER THE SUPERVISION OF

Mr. Shouvik Sarkar

C-DAC Bangalore

Submitted By

Rohan S. Dongare (220950125033)
Harshal Girhepuje (220950125037)
Hrishikesh N. Bagul (220950125038)
Mulla Farhan Dilawar (220950125051)
Nachiket Asare (220950125052)

PG DIPLOMA IN BIG DATA ANALYTICS
C-DAC, BANGALORE

Candidate’s Declaration

We hereby certify that the work being presented in the report entitled **PLANT DISEASES CLASSIFICATION USING MACHINE LEARNING ALGORITHMS**, in partial fulfillment of the requirements for the award of PG Diploma Certificate and submitted in the department of PG-DBDA of the C-DAC Bangalore, is an authentic record of our work carried out during the period, 18th February 2023 to 15th March 2023 under the supervision of **Mr. Shouvik Sarkar**, C-DAC Bangalore. The matter presented in the report has not been submitted by us for the award of any degree of this or any other Institute/University.

(Name and Signature of Candidate)

Rohan S. Dongare (220950125033)

Harshal Girhepuje (220950125037)

Hrishikesh N. Bagul (220950125038)

Mulla Farhan Dilawar (220950125051)

Nachiket Asare (220950125052)

Counter Signed by

_ _ _ _ _ _ _ _ _ _ _ _ _

ACKNOWLEDGMENT

We take this opportunity to express our gratitude to all those people who have been directly and indirectly with us during the competition of this project.

We pay thanks to Mr. Shouvik Sarkar who has given guidance and a light to us during this major project. His versatile knowledge about “title name “has eased us in the critical times during the span of this Final Project.

We acknowledge here out debt to those who contributed significantly to one or more steps. We take full responsibility for any remaining sins of omission and commission.

Students Name

CERTIFICATE

This is to certify that the work titled **PLANT DISEASES CLASSIFICATION USING MACHINE LEARNING ALGORITHMS** is carried out by Rohan S. Dongare (220950125033), Harshal Girhepuje (220950125037), Hrishikesh N. Bagul (220950125038), Mulla Farhan Dilawar (220950125051), Nachiket Asare (220950125052) the bonafide studentsof Diploma in Big Data Analytics of Centre for Development of Advanced Computing, Electronic City, Bangalore from 18th February 2023 - 15th March 2023. The Course End Project work is carried out under my directsupervision and 80% completed.

Mr. Shouvik Sarkar

C-DAC #68, Electronic

City,Bangalore - 560100,

India

ABSTRACT

The agricultural sector plays a key role in supplying quality food and makes the greatest contribution to growing economies and populations. Plant disease may cause significant losses in food production and eradicate diversity in species. Early diagnosis of plant diseases using accurate or automatic detection techniques can enhance the quality of food production and minimize economic losses. In recent years, deep learning has brought tremendous improvements in the recognition accuracy of image classification and object detection systems. Hence, in this we utilized convolutional neural network (CNN)-based pre-trained models for efficient plant disease identification. The experiments were carried out using the popular Plant Village dataset, which has 7500 image samples of different plant disease species in 3 classes. The performance of the model was evaluated through classification accuracy, sensitivity, specificity, and F1 score.

TABLE OF CONTENT

CHAPTER 1
INTRODUCTION

Agriculture is the primary source of national income for many countries, including Iran. Crop diseases are serious causes of reducing the quantity and quality of production; therefore, identifying plant diseases are of great importance. Disease symptoms can occur in different parts of the plant; however, plant leaves are commonly used to diagnose diseases. Early and accurate diagnosis is a critical first step in mitigating losses caused by plant diseases. An incorrect diagnosis can lead to improper management decisions, such as selecting the unsuitable chemical application, potentially resulting in further health loss and yield reduction.

The unaided eye method is a traditional method of identifying diseases that requires enormous manpower and is prone to human error, time-consuming, and not applicable for large fields. In addition, it is costly as it requires continuous monitoring by experts. Intelligent disease detection techniques can be beneficial in detecting a plant disease at the initial growth stages. As a reliable prediction methodology, machine learning can detect various fungal, bacterial, and viral diseases. Intelligent detection of plant diseases by utilizing machine learning algorithms is an essential research topic as it may prove advantageous in monitoring large fields and automatically detecting diseases based on symptoms appearing on plant leaves. Advanced technologies can be used to reduce the adverse effects of plant diseases by diagnosing them in early development stages. The application of artificial intelligence and computer vision for the automatic diagnosis of plant diseases is now widely studied because human monitoring of plant diseases is tedious, time-consuming, and challenging. In recent years, there has been a lot of research in the field of machine vision in agriculture, including fruit maturity classification and quality rating, fruit disease diagnosis, plant pest diagnosis, plant species classification, fruit identification in harvesting robots, weed control and recognition, and disease diagnosis and classification in plant organs. Machine vision can include a variety of sensors, such as color, multispectral, and hyperspectral cameras. In typical machine vision applications, illumination used and captured by the sensor is in the visible spectral range.

OBJECTIVES:

There are three objectives of the proposed methodology:

i. To develop a prototype for a plant disease detection system.

ii. To apply image processing techniques to identify the disease pattern.

iii. Use machine learning algorithms to predict disease. iv. Use transfer learning techniques to predict disease.

SCOPE:

The proposed methodology is used for the precise detection of disease in crops. Which can provide controlled fertilization to farmers. Accurate identification of disease also helps farmers to identify the infection and do relatively controlled fertilization to avoid any future crop failures.

CHAPTER 2
LITERATURE SURVEY

Mr. Ramachandra Hebbar et al (2018). Plant Disease Detection Using Machine Learning proposed this paper makes use of Random Forest in identifying between healthy and diseased leaf from the data sets created. Our proposed paper includes various phases of implementation namely dataset creation, feature extraction, training the classifier and classification.

David Hughes et al (2016). Using Deep Learning for Image-Based Plant Disease Detection using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to identify 14 crop species and 26 diseases (or absence thereof). The trained model achieves an accuracy of 99.35% on a held-out test set, demonstrating the feasibility of this approach

Jun Liu and Xuewei Wang (2021). Plant diseases and pests detection based on deep learning: a review. This review provides a definition of plant diseases and pests detection problem, puts forward a comparison with traditional plant diseases and pests detection methods. According to the difference of network structure, this study outlines the research on plant diseases and pests detection based on deep learning in recent years from three aspects of classification network, detection network and segmentation network.

S. Rajagopal et al (2021) Automated plant leaf disease detection and classification using optimal Mobile Net based convolutional neural networks. Automated plant disease detection techniques are advantageous for reducing the laborious task of monitoring large crop farms and for identifying disease symptoms early on, i.e., when they appear on plant leaves. Recent advances in computer vision and deep learning (DL) models have demonstrated the value of developing automatic plant disease detection models based on visible symptoms on leaves.

Tan Soo Xian et al (2021) Plant Diseases Classification using Machine Learning. The objective of this research is to classify the plant diseases by assessing the images of the leaves with the application of Extreme Learning Machine (ELM), a Machine Learning classification algorithm with a single layer feed-forward neural network. This work proposed image features as input where the image is pre-processed via HSV colour space and features extraction via Haralick textures. The results produced from the ELM shows a better accuracy that is 84.94% when compared to other models such as the Support Vector Machine and Decision Tree.

C Jackulin, S. Murugavalli (2022). A comprehensive review on detection of plant disease using machine learning and deep learning approaches, In this approach, a comprehensive review has been made on the various techniques employed in plant disease detection using artificial intelligence (AI) based machine learning and deep learning techniques. Likewise, deep learning has also gained a great deal of significance in offering better performance outcome for detecting plant disease in the computer vision field.

Sunil S. Harakannanavar et al. (2022), Plant leaf disease detection using computer vision and machine learning algorithms, the idea behind the paper is to bring awareness amongst the farmers about the cutting-edge technologies to reduce diseases in plant leaf. Since tomato is merely available vegetable, the approaches of machine learning and image processing with an accurate algorithm is identified to detect the leaf diseases in the tomato plant.

Seyed Mohamad Javidan et al (2023), Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning, In this study, a novel image processing algorithm and multi-class support vector machine (SVM) were used to diagnose and classify grape leaf diseases, i.e., black measles, black rot, and leaf blight. The area of disease symptoms was separated from the healthy parts of the leaf utilizing K-means clustering automatically, and then the features were extracted in three color models, namely RGB, HSV, and lab.

Aanis Ahmad et al. (2023), A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools, this study presents a comprehensive overview of 70 studies on deep learning applications and the trends associated with their use for disease diagnosis and management in agriculture. The studies were sourced from four indexing services, namely Scopus, IEEE Xplore, Science Direct, and Google Scholar, and 11 main keywords used were Plant Diseases, Precision Agriculture, Unmanned Aerial System (UAS), Imagery Datasets, Image Processing, Machine Learning, Deep Learning, Transfer Learning, Image Classification, Object Detection, and Semantic Segmentation.

CHAPTER 3
METHODOLOGY

3.1 ML IMPLEMENTATION:

In the automation of multiple processes, machine learning plays a critical role. The proposed architecture was designed with that goal in mind, and it is based on machine learning methodologies. Especially in the case of detecting and categorizing images into various disease categories. This section has been structured such That the topic begins with the device specifications and data acquisition for the data used in the currently suggested approach. The second point of discussion would be image segmentation. Feature extraction would be the focus of the debate. The fourth point of discussion would be the classification method Product. Functions.

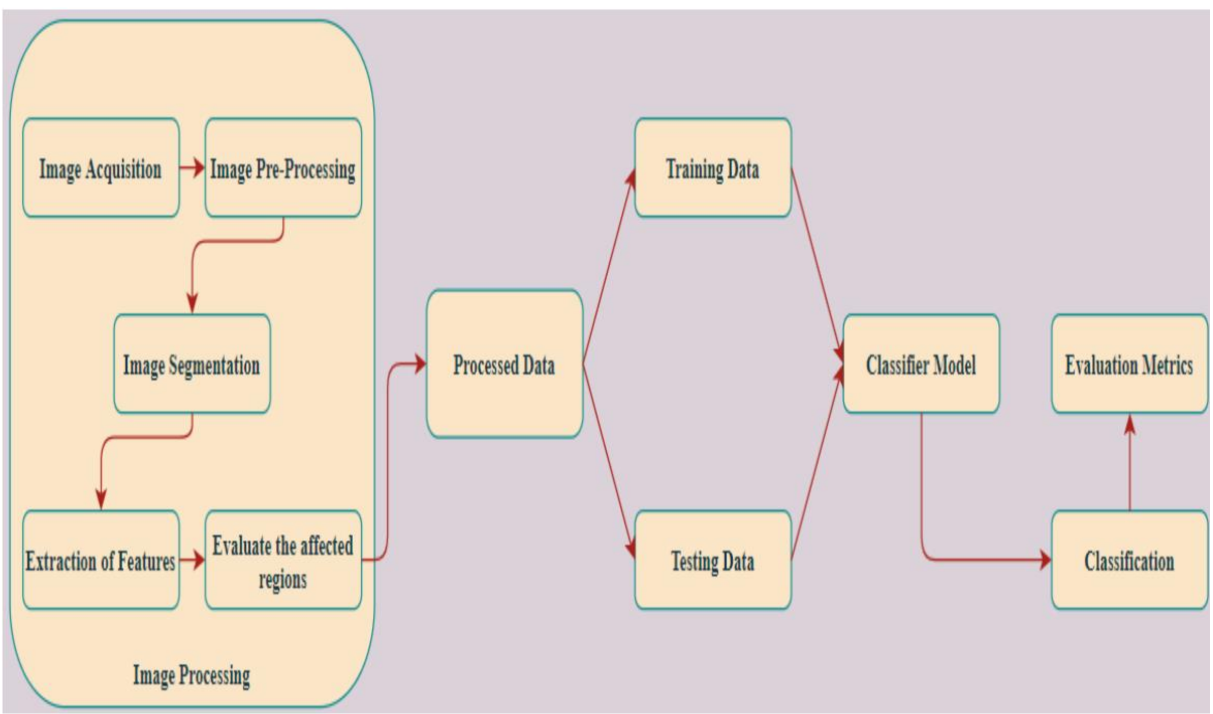


Fig.1) Flow Chart of the Machine Learning Methodology

3.2 RESEARCH METHODOLOGY:

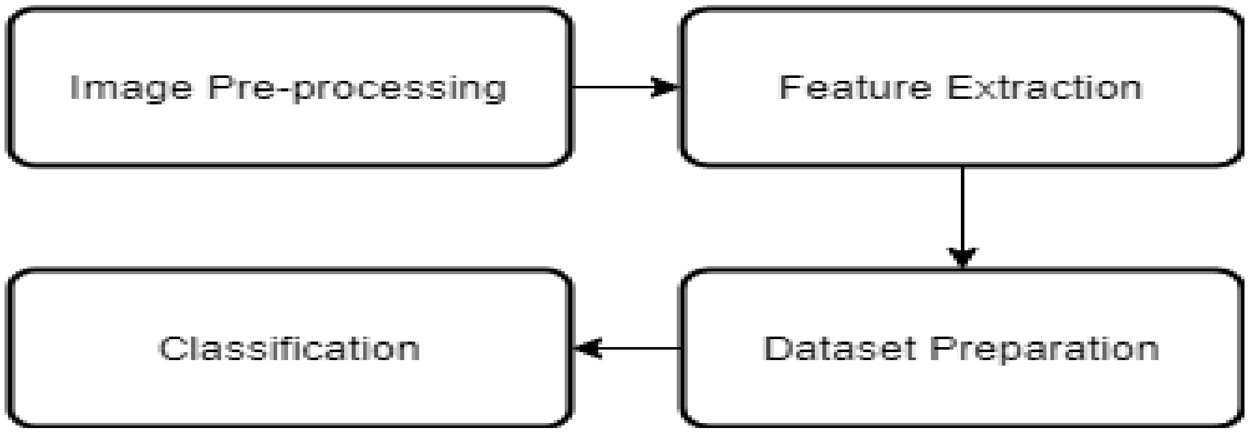


Fig 2). Diagram for Research Methodology

Image Pre-processing:

Pre-processing is a common name for operations with images at the lowest level of abstraction - both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightness's). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, and translation) are classified among pre-processing methods here since similar techniques are used.

Image pre-processing methods are classified into **four categories** according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness.

- Section 4.1 deals with pixel brightness transformations,
- Section 4.2 describes geometric transformations,
- Section 4.3 considers preprocessing methods that use a local neighborhood of the processed pixel
- Section 4.4 briefly characterizes image restoration that requires knowledge about the entire image.

Feature Extraction:

When image segmentation provides infected region to do analysis. There are lot of data that can be extracted from the image. Dimensionality reduction is a crucial step to be followed. So as to avoid model confusion and conflict also it is very important to consider all necessary features andavoid any miss. Feature Extraction is one of the most crucial steps in machine learning. Extractionof essential features is very important. Features need to be selected to avoid overfitting and under-fitting.

Dataset Preparation:

This section will explain the dataset preparation after image pre-processing and feature extraction is completed. A normalization process will be performed by subtracting the minimum value in the featuresand then dividing by the range.

The train and test datasets are shuffled to avoid any bias or patterns in the datasets before model training, data frame created were shuffled and the index were recreated so that the it are not be accidentally sorted back to the original arrangement.

Classification Process:

Depending on the features extracted, the extracted feature dataset classification need to beperformed for various categories: Early Blight, Late Blight & healthy leaves.

Image classification is the process of categorizing and labeling groups of pixels or vectors within an image based on specific rules. The categorization law can be devised using one or more spectral or textural characteristics. Two general methods of classification are ‘supervised’ and ‘unsupervised’.

Unsupervised classification method is a fully automated process without the use of training data. Using a suitable algorithm, the specified characteristics of an image is detected systematically during the image processing stage. The classification methods used in here are ‘image clustering’ or ‘pattern recognition’. Two frequent algorithms used are called ‘ISODATA’ and ‘K-mean’.

Supervised classification method is the process of visually selecting samples (training data) within the image and assigning them to pre-selected categories (i.e., roads, buildings, water body, vegetation, etc.) in order to create statistical measures to be applied to the entire image.

3.3 PROPOSED METHODOLOGY:

A series of steps need to be carefully followed for the process need to be followed in a disciplined manner:

Step-1*:* Image Acquisition for dataset creation: This step involves exploring various data sources from where data can be extracted for training the model and further how the test image input is tobe provided.

Step-2: Image Pre-processing and background removal: This is most important phase, as it involvesthe quality assurance of the data. In the image pre-processing phase image is processed to desiredcolor format, resized to desired size and images are denoised.

Step-3*:* Extraction of Features from images: On the basis of obtained region of interest which is the infected part of the leaf various image features like standard deviation, mean of red, blue and green channels, the entropy of image is extracted.

Step-4*:* Evaluate and identification of the affected region: By comparing the extracted region of interests & features which are extracted from the image, an efficient model is derived.

Step-5*:* Processed Dataset creation: The data which are processed in previous stages are processedand extracted and converted to a data frame and stored. This stored data is further utilized foranalysis purpose.

Step-6*:* Training Data Extraction

Step-7*:* Validation Data Extraction

Step-8*:* Classification: Test data has labels such as: Black Rot, Scab and Healthy, based on which classification is performed.

Step-9: Evaluation of proposed model: Depending on the obtained results from the classifier model, the evaluation metrics such as precision, recall, F1-score, and accuracy will be obtained.

CHAPTER 4
ARCHITECTURE

Introduction

In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks in deep learning.

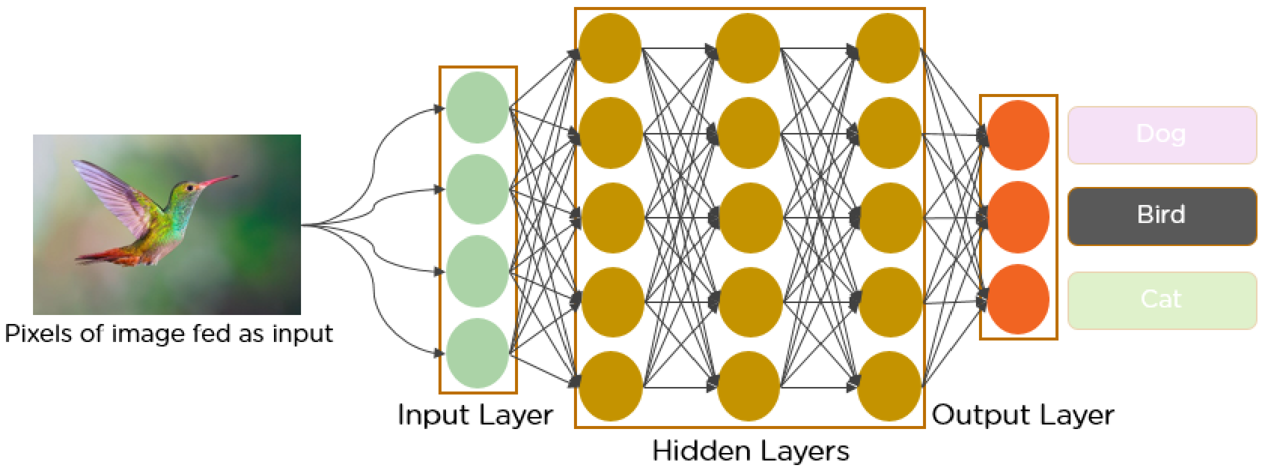


Fig.7) Layers of Convolutional Neural Networks

Since the 1950s, the early days of AI, researchers have struggled to make a system that can understand visual data. In the following years, this field came to be known as Computer Vision. In 2012, computer vision took a quantum leap when a group of researchers from the University of Toronto developed an AI model that surpassed the best image recognition algorithms and that too by a large margin.

The AI system, which became known as AlexNet (named after its main creator, Alex Krizhevsky), won the 2012 ImageNet computer vision contest with an amazing 85 percent accuracy. The runner-up scored a modest 74 percent on the test.

At the heart of AlexNet was Convolutional Neural Networks a special type of neural network that roughly imitates human vision. Over the years CNNs have become a very important part of many

Computer Vision applications and hence a part of any computer vision course online. So let's take a look at the workings of CNNs or CNN algorithm in deep learning.

What exactly is a CNN?

In deep_learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep_neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

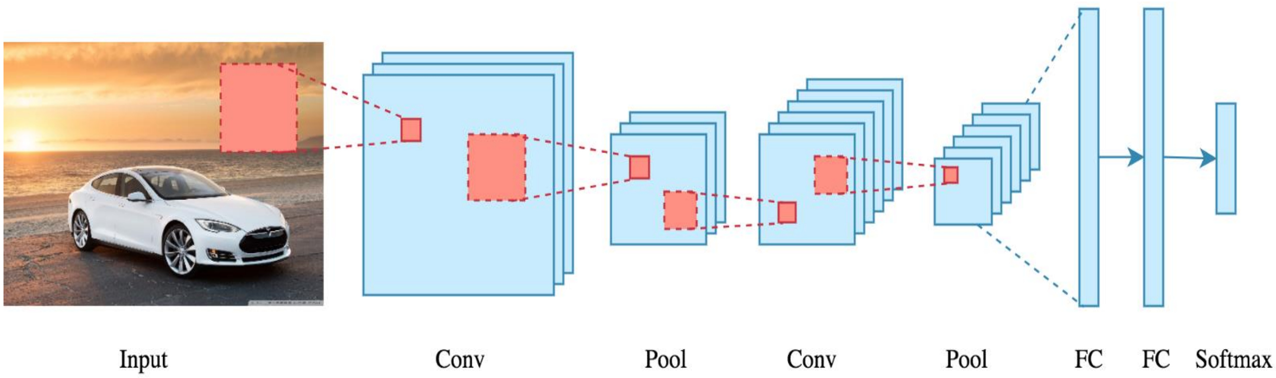


Fig 9) CNN LAYERS

But we don't really need to go behind the mathematics part to understand what a CNN is or how it works. Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

4.2 How CNN work?

Here, we applied a Convolutional neural network (CNN) based approach which is a method of DL that takes input as an image and gives importance to many other objects in the image, as well as differentiates between them. The amount of pre-processing needed by a CNN is substantially less than that required by other classification methods. While simple techniques need the hand-engineering of filters, with enough training, CNN learns these filters/characteristics. Our architecture mainly contains the following layers:

- Convolution layer
- Pooling layer
- Fully connected layer

The above figure represents the working of CNN. The input in the form of the image after preprocessing the data and extracting the required features when passed through CNN passes through 3 layers of CNN and it is precisely represented. The final output is then displayed.

- **Input Layer:** The input layer of CNN consists of the dataset. The input data will be represented as a 3X3 matrix.
- **Convolution Layer:** A layer that uses filters to learn from smaller sections of input data to obtain features from an image.
- **Pooling Layer:** This layer is used to shrink the image's dimensionality, lowering the processing power required for subsequent layers. There are two variations of pooling.
- **Max pooling:** The pixel with the maximum value as input is selected and transferred to the output while parsing input. It is the most used approach compared to average pooling.
- **Fully Connected Layer (Dense):** This is one of CNN's last layers, and it can recognize features that are significantly linked with the output class. The result is a one-dimensional vector created by flattening the pooling layer results.
- **Dropout Layer:** Used to reduce model overfitting problem by removing a random set of neurons in that layer. It is connected with the FC layer.
- **SoftMax:** This is the activation’s function that assists in classifying individual input images of the dataset into several classes depending on the learned properties from the network.
- **Output Layer:** The output layer holds the final classification result.

4.3 Limitations:

While convolutional neural networks (CNNs) are a powerful and popular tool for image classification and other computer vision tasks, they do have some limitations. Here are a few:

- **Limited to grid-like structures:** CNNs are designed to work with inputs that have a grid-like structure, such as images. This means that they may not be well-suited for tasks that involve non-grid-like data, such as natural language processing or time series analysis.
- **Overfitting:** CNNs can be prone to overfitting, meaning that they may perform well on the training data but not generalize well to new data. Techniques such as regularization can help mitigate this problem, but it can still be a challenge in some cases.
- **Computationally intensive:** CNNs can be computationally intensive and require significant computing resources to train and deploy. This can be a challenge for applications with limited computing power or memory.
- **Limited interpretability:** While CNNs can be very good at classifying images, it can be difficult to understand how they arrived at their predictions. This lack of interpretability can make it challenging to diagnose or troubleshoot issues with the model.
- **Requires large amounts of labelled data**: CNNs typically require large amounts of labelled data to train effectively. This can be a challenge in some applications where labelled data is scarce or expensive to obtain.

CHAPTER 5

IMPLMENTATION

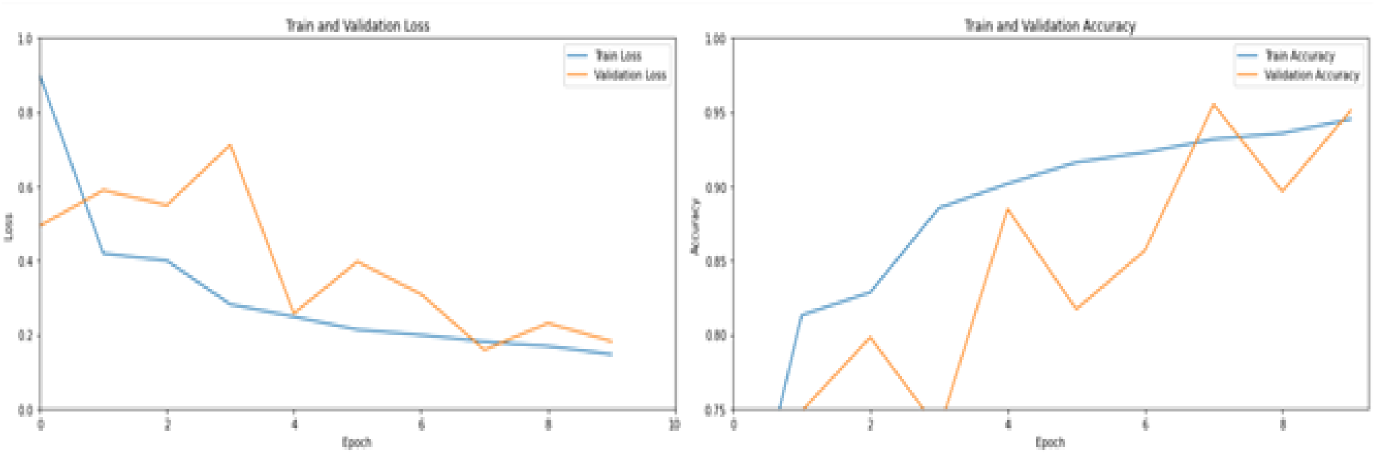
A) APPLE_SCAB Model:

```
model_1 = keras.models.Sequential()

model_1.add(keras.layers.Conv2D(64, 5, activation='relu', input_shape=(256, 256, 3)))

model_1.add(keras.layers.Dropout(0.1))
model_1.add(keras.layers.MaxPooling2D())

model_1.add(keras.layers.Conv2D(128, 5, activation='relu'))
model_1.add(keras.layers.Dropout(0.2))
model_1.add(keras.layers.MaxPooling2D())

model_1.add(keras.layers.Flatten())
model_1.add(keras.layers.Dense(256, activation='relu'))
model_1.add(keras.layers.Dense(38, activation='softmax'))
#opt = keras.optimizers.Adam(learning_rate=0.01)
model_1.compile(optimizer='SGD',loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model_1.summary()
```

Model: "sequential"

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 252, 252, 64)      4864

 dropout (Dropout)           (None, 252, 252, 64)      0

 max_pooling2d (MaxPooling2D  (None, 126, 126, 64)     0
 )

 conv2d_1 (Conv2D)           (None, 122, 122, 128)     204928

 dropout_1 (Dropout)         (None, 122, 122, 128)     0

 max_pooling2d_1 (MaxPooling  (None, 61, 61, 128)      0
 2D)

 flatten (Flatten)           (None, 476288)            0

 dense (Dense)               (None, 256)               121929984

 dense_1 (Dense)             (None, 38)                9766

=================================================================
Total params: 122,149,542
Trainable params: 122,149,542
Non-trainable params: 0
_________________________________________________________________
```

Result of SCAB:



MATRICS:

```
print("Train Accuracy  : {:.2f} %".format(history.history['accuracy'][-1]*100))
print("Test Accuracy   : {:.2f} %".format(accuracy_score(labels, predictions) * 100))
print("Precision Score : {:.2f} %".format(precision_score(labels, predictions, average='micro') * 100))
print("Recall Score    : {:.2f} %".format(recall_score(labels, predictions, average='micro') * 100))
print("F1 Score        : {:.2f} %".format(f1_score(labels, predictions, average='micro') * 100))
```

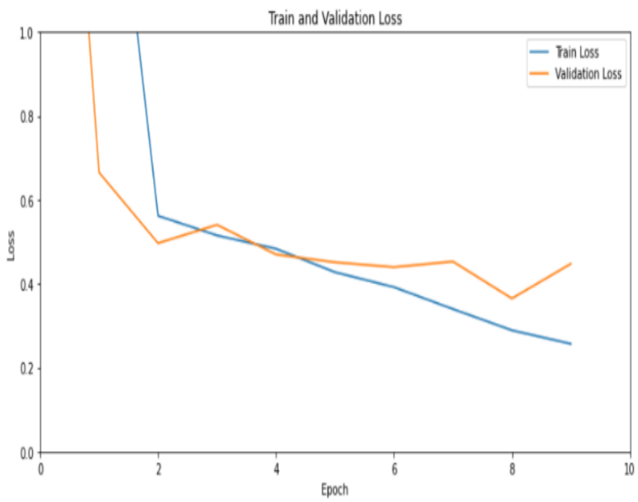
```
Train Accuracy  : 94.53 %
Test Accuracy   : 95.13 %
Precision Score : 95.13 %
Recall Score    : 95.13 %
F1 Score        : 95.13 %
```

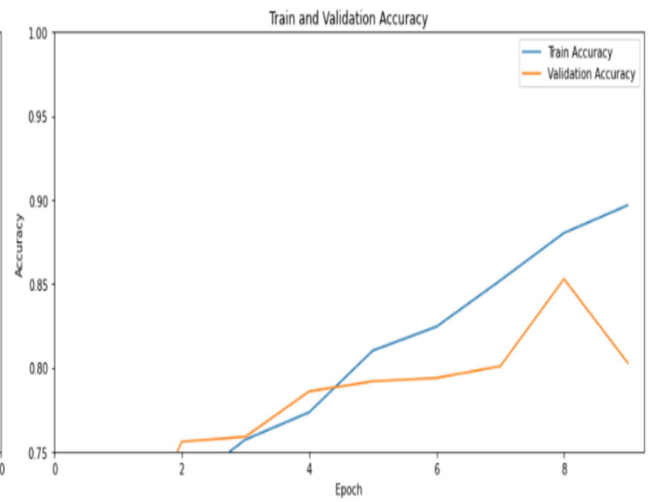
B) APPLE_BLAK_ROT Model:

```
model_1 = keras.models.Sequential()

model_1.add(keras.layers.Conv2D(64, 5, activation='relu', input_shape=(256, 256, 3)))

model_1.add(keras.layers.Dropout(0.1))
model_1.add(keras.layers.MaxPooling2D())

model_1.add(keras.layers.Flatten())

model_1.add(keras.layers.Dense(38, activation='softmax'))

model_1.compile(optimizer='SGD',loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model_1.summary()
```

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 252, 252, 64)      4864

 dropout_8 (Dropout)         (None, 252, 252, 64)      0

 max_pooling2d_8 (MaxPooling  (None, 126, 126, 64)     0
 2D)

 flatten_8 (Flatten)         (None, 1016064)           0

 dense_8 (Dense)             (None, 38)                38610470

=================================================================
Total params: 38,615,334
Trainable params: 38,615,334
Non-trainable params: 0
_________________________________________________________________
```

Result of BLACK_ROT:





MATRICS:

```
print("Train Accuracy  : {:.2f} %".format(history.history['accuracy'][-1]*100))
print("Test Accuracy   : {:.2f} %".format(accuracy_score(labels, predictions) * 100))
print("Precision Score : {:.2f} %".format(precision_score(labels, predictions, average='micro') * 100))
print("Recall Score    : {:.2f} %".format(recall_score(labels, predictions, average='micro') * 100))
```

```
Train Accuracy  : 89.68 %
Test Accuracy   : 80.28 %
Precision Score : 80.28 %
Recall Score    : 80.28 %
```

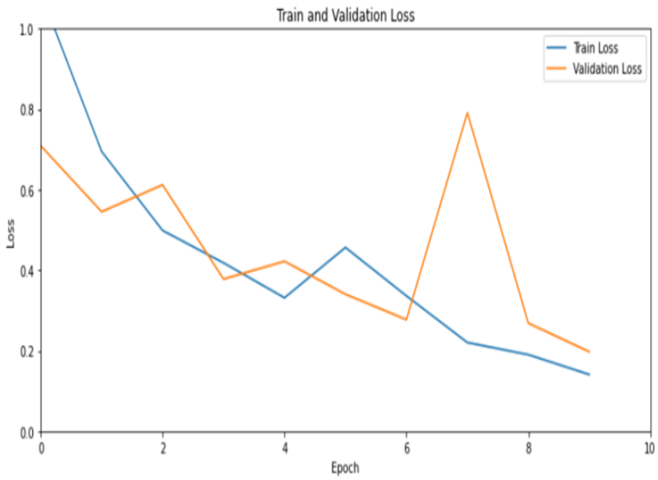
C) COMBINATIONATION OF APPLE_SCAB AND BLACK ROT:

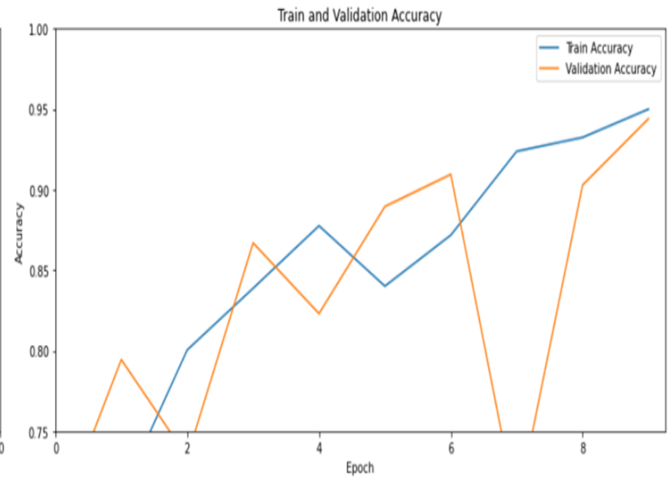
```
model_1 = keras.models.Sequential()

model_1.add(keras.layers.Conv2D(64, 5, activation='relu', input_shape=(256, 256, 3)))

model_1.add(keras.layers.Dropout(0.1))
model_1.add(keras.layers.MaxPooling2D())

model_1.add(keras.layers.Conv2D(128, 5, activation='relu'))
model_1.add(keras.layers.Dropout(0.2))
model_1.add(keras.layers.MaxPooling2D())

model_1.add(keras.layers.Flatten())
model_1.add(keras.layers.Dense(256, activation='relu'))
model_1.add(keras.layers.Dense(38, activation='softmax'))

model_1.compile(optimizer='SGD',loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model_1.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 252, 252, 64)      4864

 dropout_2 (Dropout)         (None, 252, 252, 64)      0

 max_pooling2d_2 (MaxPooling  (None, 126, 126, 64)     0
 2D)

 conv2d_3 (Conv2D)           (None, 122, 122, 128)     204928

 dropout_3 (Dropout)         (None, 122, 122, 128)     0

 max_pooling2d_3 (MaxPooling  (None, 61, 61, 128)      0
 2D)

 flatten_1 (Flatten)         (None, 476288)            0

 dense_2 (Dense)             (None, 256)               121929984

 dense_3 (Dense)             (None, 38)                9766

=================================================================
Total params: 122,149,542
Trainable params: 122,149,542
Non-trainable params: 0
_________________________________________________________________
```

Result of APPLE_SCAB AND BLACK ROT:





MATRICS:

```python
print("Train Accuracy  : {:.2f} %".format(history.history['accuracy'][-1]*100))
print("Test Accuracy   : {:.2f} %".format(accuracy_score(labels, predictions) * 100))
print("Precision Score : {:.2f} %".format(precision_score(labels, predictions, average='micro') * 100))
print("Recall Score    : {:.2f} %".format(recall_score(labels, predictions, average='micro') * 100))
```

```
Train Accuracy  : 95.55 %
Test Accuracy   : 96.34 %
Precision Score : 96.34 %
Recall Score    : 96.34 %
```

CHAPTER 6
CONCLUSION

A thorough study on several machine and deep learning algorithms for plant disease recognition and classification has been published. After that, different machine learning classification algorithms may be applied to detect plant diseases in an effort to assist farmers with automatic disease detection of all types of agricultural diseases that were to be found. This analysis goes over various DL methods for detecting plant diseases. Here, the newest advancements in deep learning technology for the diagnosis of plant leaf diseases. We believe that this work will be a valuable resource for researchers attempting to identify plant diseases.

FUTURE SCOPE:

1. The disease detection system can be integrated in cloud system for efficient result processing.
2. Integration of automated disease detection system with sensors to measure soil properties

CHAPTER 7
REFERENCES

- Sardogan, M., Tuncer, A., and Ozen, Y. “Plant Leaf Disease Detection and Classification Based on CNN with the LVQ Algorithm”. In: 3rd Int. Conf. Comp. Sci. Eng. (2018) 382–385
- Shrivastava, V. K., Pradhan, M. K., Minz, S., and Thakur, M. P.: “Rice plant disease classification using transfer learning of deep convolution neural network.” Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch., 42 (2019) 631–635
- Uğuz, S. and Uysal, N.: “Classification of olive leaf diseases using deep convolutional neural networks”. Neural Comp. Appl., 5 (2020)
- Golhani, K., Balasundram, S. K., Vadamalai, G., & Pradhan, B. (2018). “A review of neural networks in plant disease detection using hyperspectral data”. Information Processing in Agriculture, 5(3), 354–371. https://doi.org/10.1016/j.inpa.2018.05.002
- Tan Soo Xian , Ruzelita Ngadiran, “Plant Diseases Classification using Machine Learning (2021)” Journal of Physics: Conference Series.”
- Mohamad Javidan, Ahmad Banakar , KeyvanAsefpour Vakilian, Yiannis Ampatzidis “Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning”, Smart Agriculture Technology.Vol 3 (2023).
- C Jackulin, S. Murugavalli, (2022) “A comprehensive review on detection of plant disease using machine learning and deep learning approaches”, Measurements Volume 27.
- Sharada P. Mohanty, David P. Hughes and Marcel Salathé (2016), “Using deep learning for image based plant disease detection”