Plant Disease Identification & Classification Using Deep Learning

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR
THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

SUBMITTED BY

Arvind Kumar Roll no. 1907350130019

Ankit Tejwan Roll no. 1907350130014

Deepak Kumar Roll no. 1907350130025

UNDER THE SUPERVISION OF

Mr. Sudhir Goswami, Assistant Professor, REC Bijnor

DEPARTMENT OF INFORMATION TECHNOLOGY
RAJKIYA ENGINEERING COLLEGE
CHANDPUR, BIJNOR

UTTAR PRADESH(INDIA)
May 2023

CANDIDATE'S DECLARATION

I hereby declare that the work presented in the dissertation titled '**Plant Disease Identification & Classification Using Deep Learning**' submitted towards the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology from Rajkiya Engineering College, Chandpur, Bijnor, UP, India is an authentic record of my own work carried out in the period of August 2022 to May 2023 under the sincere guidance of **Sudhir Goswami**, Assistant Professor at Rajkiya Engineering College ,Bijnor. It is also stated that no earlier submission of the subject matter of the work demonstrated in this dissertation has been made for the award of any other degree in this or any other University/Institute.

Ankit Tejwan (1907350130014)
Arvind Kumar (1907350130019)
Deepak Kumar (1907350130025)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Sudhir Goswami (Supervisor)
Assistant Professor
Rajkiya Engineering College , Bijnor

The B.Tech (Report) Viva-Voce examination of Ankit Tejwan (1907350130014), Arvind Kumar (1907350130019), Deepak Kumar (1907350130025) has been held on _____________ and accepted.

External Examiner **Supervisor** **Head of Department**

CERTIFICATE

This is to certify that a project entitled "**Plant Disease Identification & Classification Using Deep Learning**" has been submitted to the Department of Information Technology,Rajkiya Engineering College Bijnor (Dr. A P J Abdul Kalam Technical University, Lucknow, U. P**.**) for the fulfillment of the requirements for the degree of Bachelor of Technology in Information technology by following students of final year B.Tech. (Information Technology)

Student Name	**Roll.No.**
Ankit Tejwan	1907350130014
Arvind Kumar	1907350130019
Deeepak Kumar	1907350130025

Date:

Supervisor

Sudhir Goswami

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my supervisor Mr. Sudhir Goswami, Assistant Professor at Rajkiya Engineering College, Chandpur, Bijnor, Department of Information Technology, Rajkiya Engineering College, Chandpur, Bijnor for the useful comments, remarks, and engagement through the learning process of this Project . I cannot thank them enough for their tremendous support and help. They motivated and encouraged me throughout this work. I consider myself extremely fortunate to have a chance to work under their supervision. Despite their busy schedule, they were always approachable and took their time off to guide me and gave appropriate advice.

I also wish to thank wholeheartedly all the faculty members of the Department Information Technology for the invaluable knowledge they have imparted to me and for teaching the principles most excitingly and enjoyably. I also extend my thanks to the technical and administrative staff of the department for maintaining an excellent working facility.

I would like to thank my family for their continuous support and blessings throughout the entire process. I would also like to extend thanks to my friends for the useful discussions, constant support, and encouragement during the whole period of the work.

Signature :

Name : Ankit Tejwan
Roll No : 1907350130014

Signature :

Name : Arvind Kumar
Roll No : 1907350130019

Signature :

Name : Deepak Kumar
Roll No : 1907350130025

Date :

ABSTRACT

For their food, humans essentially depend on the environment. These foods came from trees, crops and plants. Growing of plants and crops certain illnesses are possible. These illnesses put food security at risk, so the identification and classification of plant diseases required.Plant diseases affect the growth of their respective species, therefore their early identification is very important. Deep learning has shown promising results in various image classification tasks, and its application in plant disease identification and classification is a growing field of research. In this paper, we propose a deep learning-based approach for the identification and classification of plant diseases.. Deep learning is a branch of artificial intelligence. In recent years, with the advantages of automatic learning and feature extraction, it has been widely concerned by academic and industrial circles. It has been widely used in image and video processing, voice processing, and natural language processing. At the same time, it has also become a research hotspot in the field of agricultural plant protection, such as plant disease recognition and pest range assessment, etc. The application of deep learning in plant disease recognition can avoid the disadvantages caused by artificial selection of disease spot features, make plant disease feature extraction more objective, and improve the research efficiency and technology transformation speed. This review provides the research progress of deep learning technology in the field of plant disease identification in recent years. This study provides a valuable tool for the efficient and accurate identification and classification of plant diseases, which can aid in the development of more effective disease control strategies In this paper, we present the current trends and challenges for the detection of plant leaf disease using deep learning and advanced imaging techniques. We hope that this work will be a valuable resource for researchers who study the detection of plant diseases and insect pests. At the same time, we also discussed some of the current challenges and problems that need to be resolved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RGB	Red Green Blue
MATLAB	Matrix Laboratory
GLCM	Gray-Level Co- occurrence Matrix
PCR	Polymerase Chain Reaction
RCNN.	Region-Based Neural Networks
CNN	Convolutional Neural Networks

CHAPTER 1

INTRODUCTION

1.1 Background

Plant diseases are a major concern for agriculture and food security. Early identification and classification of plant diseases can help farmers to take timely measures to prevent the spread of diseases and protect their crops. Traditional methods of plant disease identification involve visual inspection, which can be time-consuming and subject to human error.

1.2 Problem statement

The objective of this project is to develop a deep learning-based system for the automated identification and classification of plant diseases. This project aims to improve the accuracy and efficiency of disease identification and to provide a tool that can assist farmers in protecting their crops.

Fig.1.1 Leaf scars

1.3 Objectives

The main objectives of this project are:

- To develop a deep learning-based model for plant disease identification and classification

- To evaluate the performance of the proposed model on a dataset of plant images
- To compare the performance of the proposed model with traditional methods of disease identification
- Scope of the project: This project will focus on the identification and classification of a limited number of plant diseases. The dataset used for training and evaluation will be limited to images of leaves affected by the selected diseases. The proposed model will be tested on new images to evaluate its performance in identifying and classifying the selected diseases.

Overall, this project aims to demonstrate the potential of deep learning in the field of plant disease identification and classification, and to provide a starting point for further research in this area.

CHAPTER 2
LITERATURE REVIEW

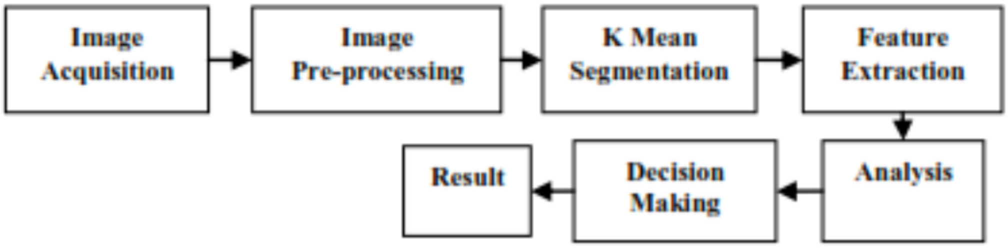
2.1 Overview of Plant Diseases

Plant diseases can have a significant impact on crop yields and food security. They can be caused by a variety of factors, including pathogens, environmental conditions, and genetic factors. Some common types of plant diseases include bacterial, fungal, viral, and nematode infections. Symptoms of plant diseases can include leaf discoloration, wilting, and abnormal growth patterns.

2.2 Previous Approaches for Disease Identification

In recent years, image processing techniques are used in various fields such as automation, medical etc. Even the identification of plant infection using traditional method is replaced by image processing. The image processing systems requires camera, computer and necessary software. Steps involved in plant disease detection are image acquisition, pre-processing, segmentation, feature extraction and classification [1]. Performing image enhancement, improves the quality of the image as well as the clarity. Basic primary colours red, green and blue combinations produce many varieties of colours. Hence, implementing image processing using RGB components is difficult and its range is very high. Converting RGB image into its equivalent grey image is done for easier implementation [2]. Automated plant disease using image processing technique is beneficial for the farmers as it reduces large human labours and can help to detected by symptoms at early stage [3]. MATLAB software's image processing tools are used for detecting the disease of the plants.

Image acquisition is performed using digital cameras. K-mean clustering algorithm used Euclidean distance metric method and clusters the image based on the specified number of groups [4][5]. Gray-Level Co- occurrence Matrix (GLCM) is one of the most popular methods for texture analysis. It produces a feature based gray level matrix for the colour image and measures the spatial distance between the pixels. GLCM represents the distance and angular spatial relationship of an image in a specific size. GLCM calculates how often the pixel with gray level intensity occurs. Horizontally values are represented as „i" and vertically or diagonally values to adjacent pixels are labelled as „j" [6][7][8].



(Fig. 2.2) Block Diagram of Steps Involved in Plant Infection Detection System

Mobile phones or digital camera are used to take images of infected leafs of different plants. Image processing techniques are applied on those images to get useful features for analyzing. The various steps involved are shown in the Figure 1.

Traditional methods for plant disease identification include visual inspection, serological testing, and PCR-based methods. These methods can be time-consuming, require specialized equipment, and are subject to human error. Recently, researchers have begun to explore the use of image-based methods for disease identification, such as leaf shape analysis and leaf texture analysis. These methods have shown promise in improving the accuracy and efficiency of disease identification.

2.3 Deep Learning in Image Classification

Deep learning is a type of machine learning that involves training artificial neural networks on large amounts of data. It has been successfully applied to a wide range of tasks, including image classification, natural language processing, and speech recognition. In the field of image classification, convolutional neural networks (CNNs) have been shown to be particularly effective. CNNs are designed to automatically learn features from images, making them well suited for tasks such as object detection and image segmentation.

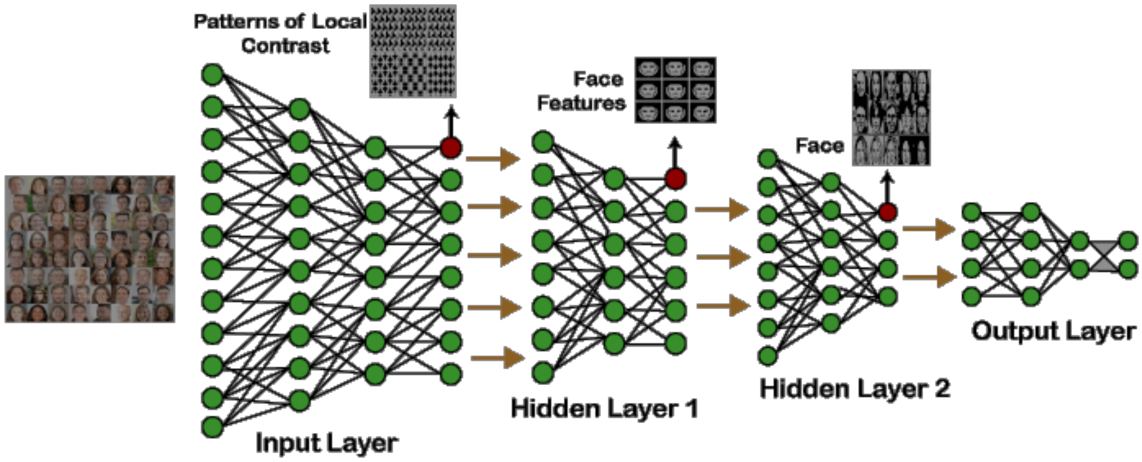


Figure 2.3 : Example of Deep Learning

The key advantage of deep learning is its ability to automatically learn features from raw data, such as images, text, or audio, without the need for manual feature engineering. This is achieved by training the neural network to learn a hierarchical representation of the data, where each layer extracts increasingly complex features from the previous layer.

Deep learning has been successfully applied to a wide range of tasks, including image classification, natural language processing, speech recognition, and even game playing. It's used in a variety of applications such as self-driving cars, image and voice recognition, natural language processing, drug discovery and many more.
Deep learning algorithms are implemented using different types of neural networks, such as feedforward neural networks, recurrent neural networks (RCNN)and convolutional neural networks (CNNs), which are well suited for tasks such as image and video analysis, and speech recognition.
Deep learning is a rapidly evolving field and new techniques and architectures are being developed to improve its performance and to make it more accessible to a wider range of applications.

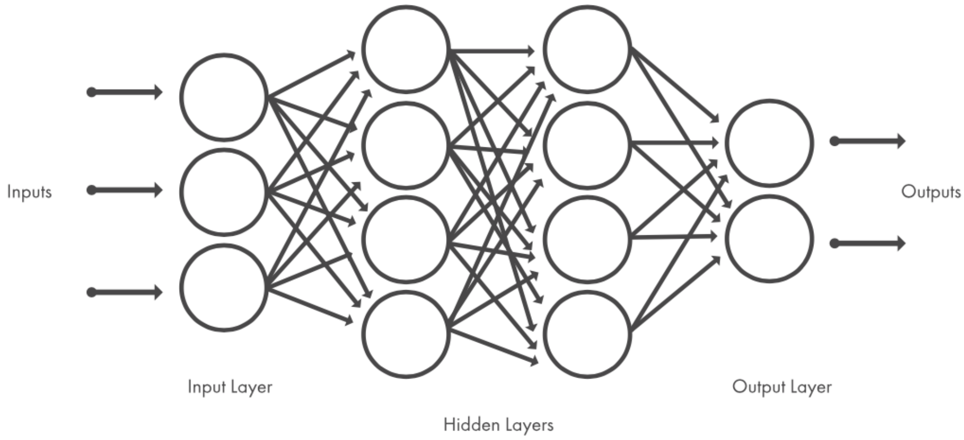


Fig 2.2.2: Neural networks, which are organized in layers consisting of a set of interconnected nodes. Networks can have tens or hundreds of hidden layers .

In recent years, there have been a number of studies that have applied deep learning techniques to the task of plant disease identification. These studies have shown that deep learning-based methods can achieve high levels of accuracy in identifying and classifying plant diseases.

In summary, the literature review shows that deep learning has a great potential in the field of plant disease identification and classification. Many studies have already shown the effectiveness of deep learning in image classification and some of them have been applied to plant disease identification with promising results.

2.3.1 Convolutional Neural Network

Convolutional Neural Networks are a special kind of neural network mainly used for image classification, clustering of images and object recognition. DNNs enable unsupervised construction of hierarchical image representations. To achieve the best accuracy, deep convolutional neural networks are preferred more than any other neural network. In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

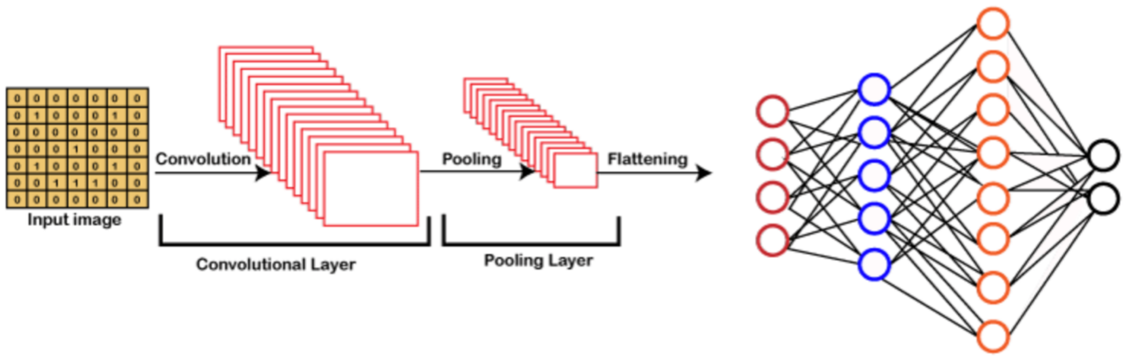


Figure 2.2 : Conventional Neural Network

A CNN typically consists of an input layer, several hidden layers, and an output layer. The input layer is where the raw image data is fed into the network. The hidden layers are made up of multiple layers of convolutional, pooling, and activation layers.

2.3.1.1 Convolutional layers

Convolutional layers are a key component of convolutional neural networks (CNNs). They are responsible for applying a set of filters to the input image in order to extract features such as edges, textures, and patterns.

The convolutional layer applies a convolution operation on the input image, which involves element-wise multiplication and summing of the input with a small set of learnable parameters called filters (also known as kernels or weights). These filters are typically small in size, such as 3x3 or 5x5 pixels, and are moved across the input image in a process called convolution.

The output of the convolution operation is called a feature map, which represents a transformed version of the input image. Each filter is responsible for extracting a specific feature from the input image, and the feature maps are a collection of these features.

The convolution operation is applied multiple times, creating multiple feature maps, each one responsible for capturing different features of the input image. These feature maps are then passed through other layers such as pooling, activation and fully connected layers to extract more complex features and make predictions.

Convolutional layers have the ability to learn hierarchical representations of the input image, which makes them well-suited for image processing tasks. They are able to learn features at different scales, such as edges in small regions and patterns in larger regions, by applying different filters in multiple convolutional layers. Additionally, the shared weights of the filters in the convolutional layers allow them to be computationally efficient and make CNNs able to process large images.

2.3.1.2 Pooling layers

Pooling layers are a key component of convolutional neural networks (CNNs) that are used to reduce the spatial size of the feature maps generated by the convolutional layers. They help to reduce the number of parameters in the network and make it more robust to small translations and deformations in the input image.

There are different types of pooling operations that can be applied, such as max pooling, average pooling and sum pooling, but max pooling is the most common one.

The max pooling operation is applied on a small region of the feature map, typically 2x2 or 3x3 pixels, and it selects the maximum value within that region. This operation is applied multiple times on the feature map, with the region sliding over the entire feature map, creating a new, smaller feature map.

Max pooling has the effect of keeping the most important information in the feature map while discarding less important information. It also helps to reduce the computational cost and the number of parameters in the network, making it more efficient and easier to train. Additionally, the pooling operation makes the CNN more robust to small translations and deformations in the input image.

Pooling layers are typically placed between the convolutional layers and the fully connected layers in a CNN. They help to extract more abstract features from the input image, making the network more robust and generalizable.

The convolutional layer applies a set of learnable filters to the input image, which is used to extract features such as edges, textures, and patterns. The pooling layer is used to reduce the spatial size of the feature maps, while preserving the most important information. The activation layer applies a non-linear function to the output of the convolutional and pooling layers to introduce non-linearity into the network.

2.3.1.3 Fully connected layer

A fully connected layer, also known as a dense layer, is a type of layer in a neural network that connects every neuron in the current layer to every neuron in the previous layer. In other words, it is a dense matrix of weights that connects all the neurons in the previous layer to the neurons in the current layer.

The main purpose of a fully connected layer is to take the features learned by the previous layers and combine them in a way that allows the network to make a final prediction or classification.

A fully connected layer typically consists of a set of neurons, each of which takes the output of all neurons in the previous layer as input, multiplies them by a set of learnable weights, and applies an activation function to produce the output.

The number of neurons in the fully connected layer is determined by the number of classes in a classification task, or by the number of the output in a regression task. Fully connected layers are typically placed at the end of a neural network, after the convolutional and pooling layers. They can be stacked multiple times to create deeper networks, but it's important to note that adding more fully connected layers increases the number of parameters in the network and makes it more difficult to train.

A CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images.

CNNs eliminate the need for manual feature extraction, so you do not need to identify features used to classify images. The CNN works by extracting features directly from images. The relevant features are not pretrained; they are learned while the network trains on a collection of images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification

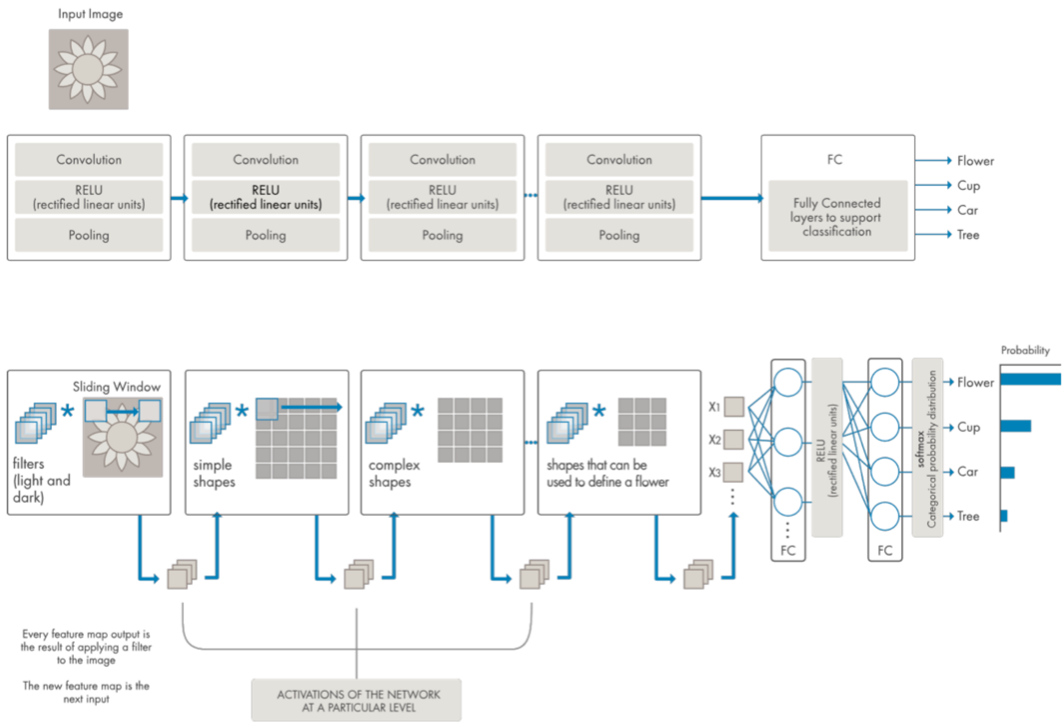


Fig 2.3.1.3: Example of a network with many convolutional layers. Filters are applied to each training image at different resolutions, and the output of each convolved image serves as the input to the next layer

2.3.2 TensorFlow

TensorFlow is an open-source software library for machine learning and deep learning developed by Google Brain Team. It is used to develop, train and deploy machine learning models, and it can be used to implement a wide range of machine learning and deep learning tasks, such as image and speech recognition, natural language processing, and even game playing.

TensorFlow allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays called tensors. It provides a collection of tools that make it easy to work with data and perform complex computations, such as automatic differentiation and parallel computing.

One of the main advantages of TensorFlow is its flexibility. It allows you to easily build and train models using a variety of different architectures and algorithms, such as deep neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

TensorFlow also provides a variety of pre-built models and libraries that can be used for common machine learning tasks, such as image classification and object detection. These pre-built models can be fine-tuned to adapt to new data and tasks, or they can be used as a starting point for building custom models.

Additionally, TensorFlow has a large community of developers and users, which provides a wealth of resources, tutorials and examples to help you get started and learn how to use it efficiently.

In summary, TensorFlow is a powerful and flexible open-source software library for machine learning and deep learning that allows you to easily develop, train, and deploy models. It provides a collection of tools and pre-built models that make it easy to work with data and perform complex computations, and it has a large community of developers and users that provide resources and support.

CHAPTER 3

METHODOLOGY

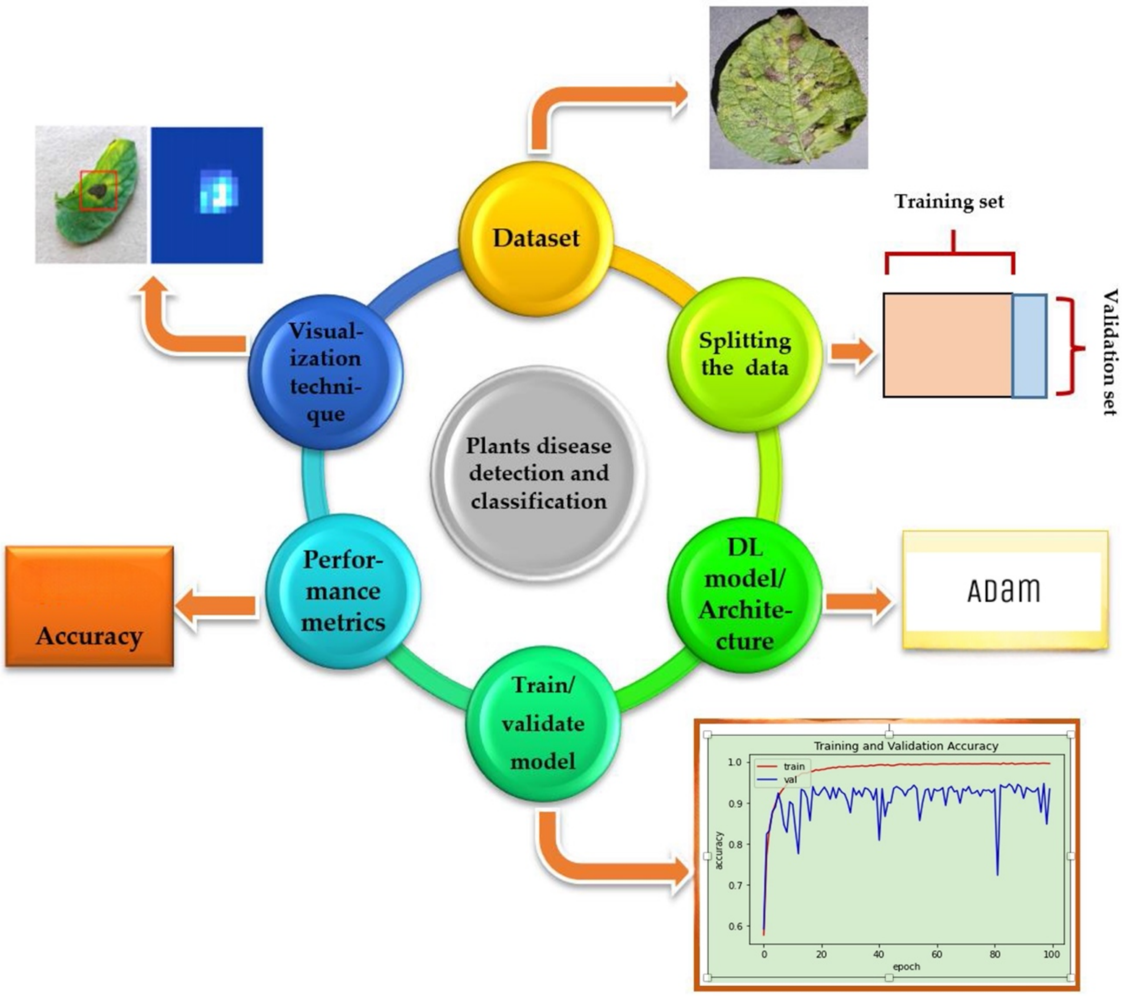


Fig.3.1: Block diagram of working method

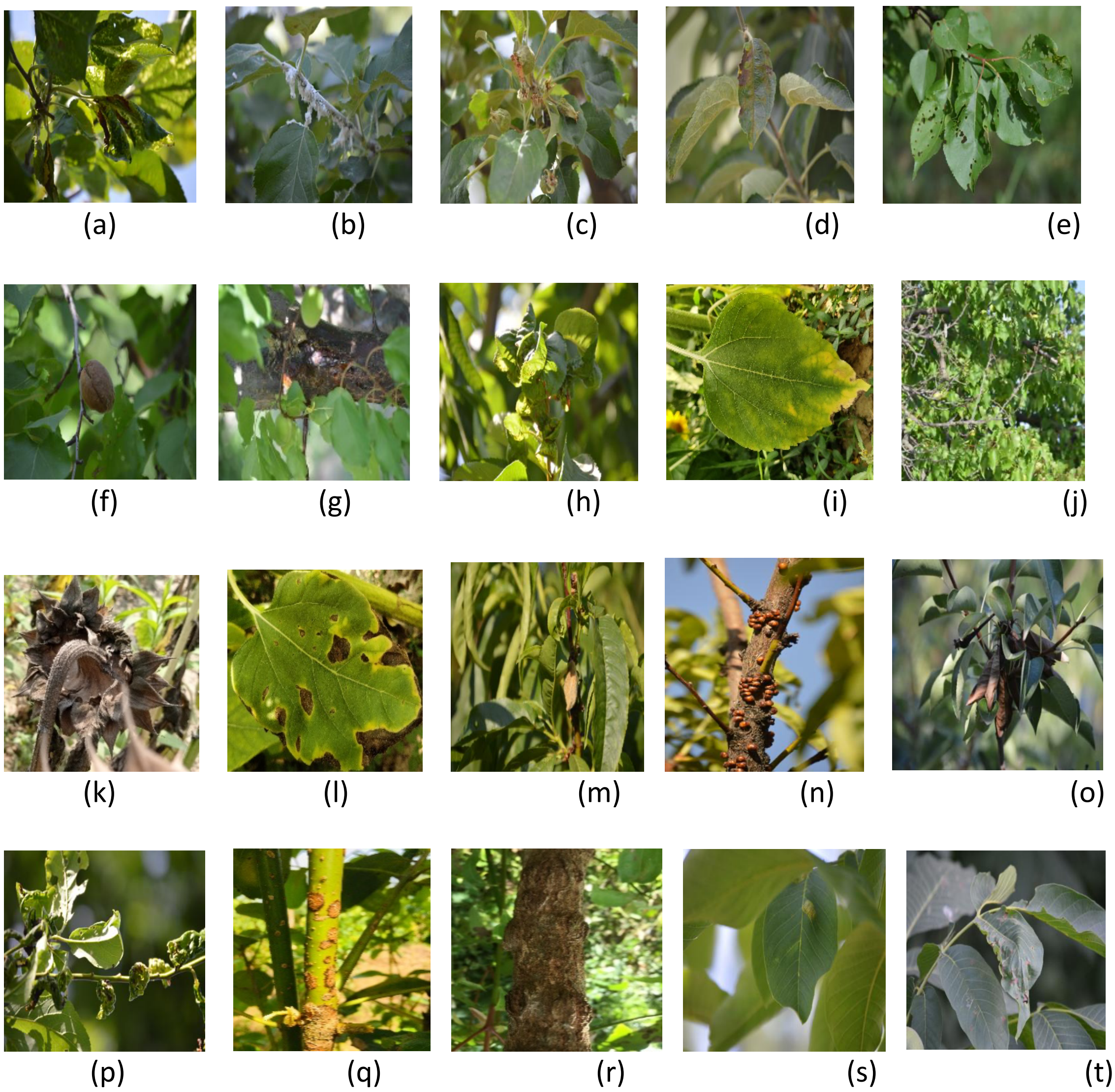
3.1 Data Collection

The dataset used for this project will consist of images of leaves affected by a selected set of plant diseases. These images will be collected from publicly available datasets and from original images taken in controlled environment. The dataset will be divided into training, validation, and test sets with the respective ratio of 7:2:1.To evaluate the performance of the model, three distinct datasets on plant diseases has been considered in this work. The datasets are Turkey plant dataset, cinnamon plant stem and branch disease dataset and sunflower fruit and leaves dataset. It is clear that the images in the collection were taken on the site and the intensity is likewise rather constant. The Turkey plant dataset is made up of 4448 images that are broken down into 15 classes that are obtained from academics working in the field of plant protection at the Agricultural Faculty of Bingol and Inonu Universities in Turkey. The Cinnamon plant stem and branch disease dataset is made up of 326 images that are broken down into 2 classes (RoughBark,StripeCanker) that are gathered from Cinnamon plantation in Sri Lanka. The sunflower fruit and leaves dataset is made up of 332 images that are broken down into 3 classes (Gray mold , Leaf scars, Downy mildew) and obtained from mendeley platform. This dataset contains total 65 classes with 15050 images of plant diseases which are classified into three categories: test, train and validation. Some classes Further detailed classification is given in the table 1.1

Class Name	**Number of images**		
	Train	***Test***	***Validation***
Apple Aphis spp	129	17	16
Apple Eriosoma lanigerum	292	38	36
Apple Monillia laxa	204	26	25
Apple Venturia inaequalis	506	64	63
Apricot Coryneum beijerinckii	880	111	110
Apricot Monillia laxa	68	9	8
Cancer symptom	60	9	7
Cherry Aphis spp	284	37	35
Downy mildew	96	12	12
Drying symptom	111	15	13
Gray mold	57	8	7
Leaf scars	112	14	14
Peach Monillia laxa	251	32	31
PeachParthenolecanium corni	341	44	42
Pear Erwinia amylovora	172	22	21
Plum Aphis spp	56	7	7
RoughBark	126	17	15
StripeCanker	134	18	16
Walnut Eriophyes erineus	55	8	6
Walnut Gnomonia leptostyla	144	18	18

Table 1.1 Class name with number of images

Some of the sample images from each class is are listed below :



(a) Apple Aphis spp
(b) Apple Eriosoma lanigerum
(c) Apple Monillia laxa
(d) Apple Venturia inaequalis
(e) Apricot Coryneum beijerinckii
(f) Apricot Monillia laxa
(g) Cancer symptom
(h) Cherry Aphis spp
(i) Downy mildew
(j) Drying symptom
(k) Gray mold

(l) Leaf scars
(m) Peach Mornillia laxa
(n) Peach Parthenolecanium corni
(o) Pear Erwinia amylovora
(p) Plum Aphis spp
(q) RoughBark
(r) StripeCanker
(s) Walnut Eriophyes erineus
(t) Walnut Gnomonia leptostyla

3.2 Data Pre-processing

The collected images will undergo pre-processing steps to prepare them for training. This will include resizing the images to a consistent size, normalizing the pixel values, and applying data augmentation techniques to increase the size of the dataset.

3.3 Data augmentation

Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data. This includes making small changes to data or using deep learning models to generate new data points. Data augmentation is useful to improve the performance and outcomes of machine learning models by forming new and different examples to train datasets. If the dataset in a machine learning model is rich and sufficient, the model performs better and more accurately.

For machine learning models, collecting and labeling data can be exhausting and costly processes. Transformations in datasets by using data augmentation techniques allow companies to reduce these operational costs.

3.4 Model Selection and Training

A convolutional neural network (CNN) will be used as the model for this project. The CNN architecture will be selected based on its performance on the validation set. The model will be trained on the training set and fine-tuned on the validation set.

To achieve the faster processing time and higher accuracy in classification problems, deep learning models are developed. CNN model is used in this work. The simple and highly effective CNN model uses. The architecture of this model consists of 18 Neural Layers. The proposed model has three modules to identify the plant disease: pre-processing, feature extraction and feature classification.

CNN model suggests very much efficient computationally and also achieved state of art result on dataset which is 95.86% top-1 accuracy. Model scaling is about scaling the existing model in terms of model depth, model width, and less popular input image resolution to improve the performance of the model.

Steps Involves:

- Image Acquisition
- Pre-processing (resizing)
- Split into Test, train and Validation
- Training on data
- Validation
- Testing
- Visualization of results

3.4.1 Model Architecture

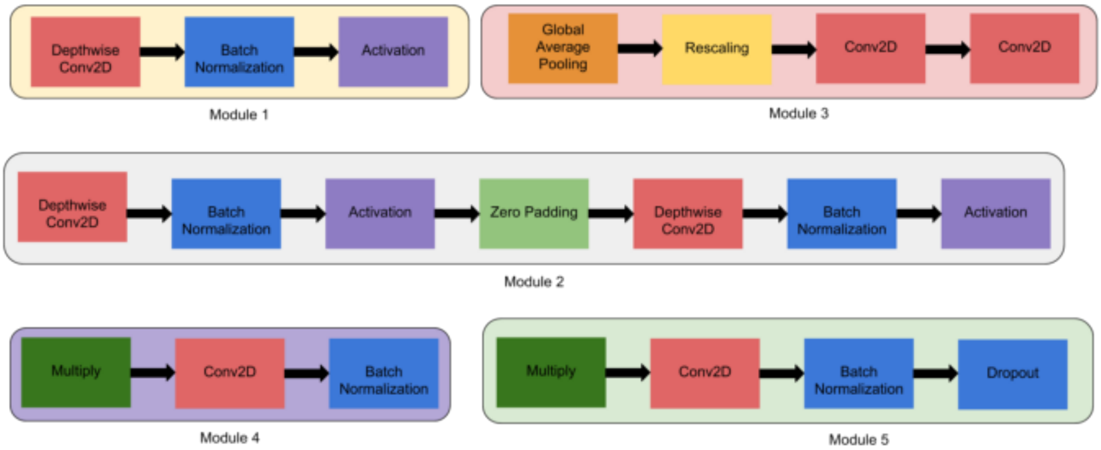


Fig 3.4.1: Model Architecture view

3.4.2 System Architecture

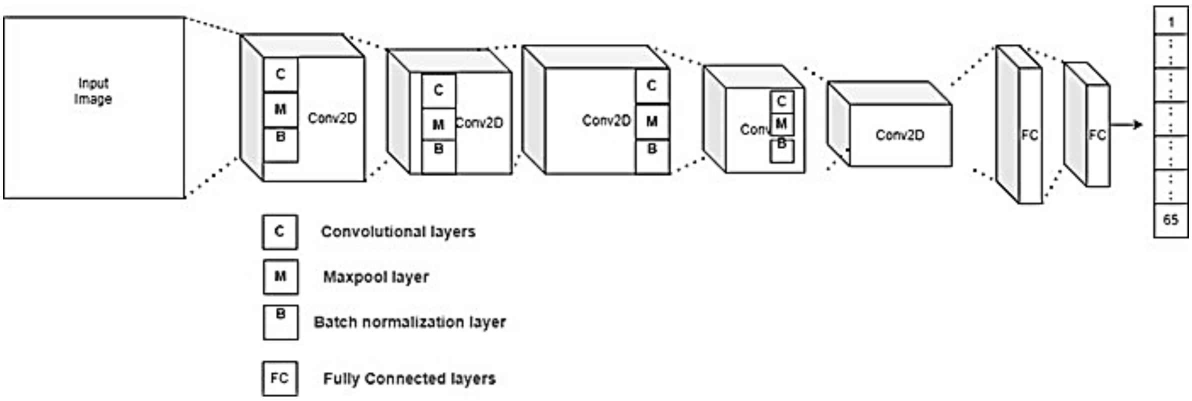


Figure 3.4.2 : System Architecture

3.5 Evaluation and Validation

The performance of the model will be evaluated on the test set. The metrics used to evaluate the performance will include accuracy, precision, recall, and F1-score. The model will also be tested on new images to evaluate its performance in identifying and classifying the selected diseases in real-world scenarios.

Overall, the methodology for this project involves collecting a dataset of images of plant leaves affected by selected diseases, pre-processing the images, training a deep learning model using the pre-processed images and evaluating the model using various metrics.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Overview of Results

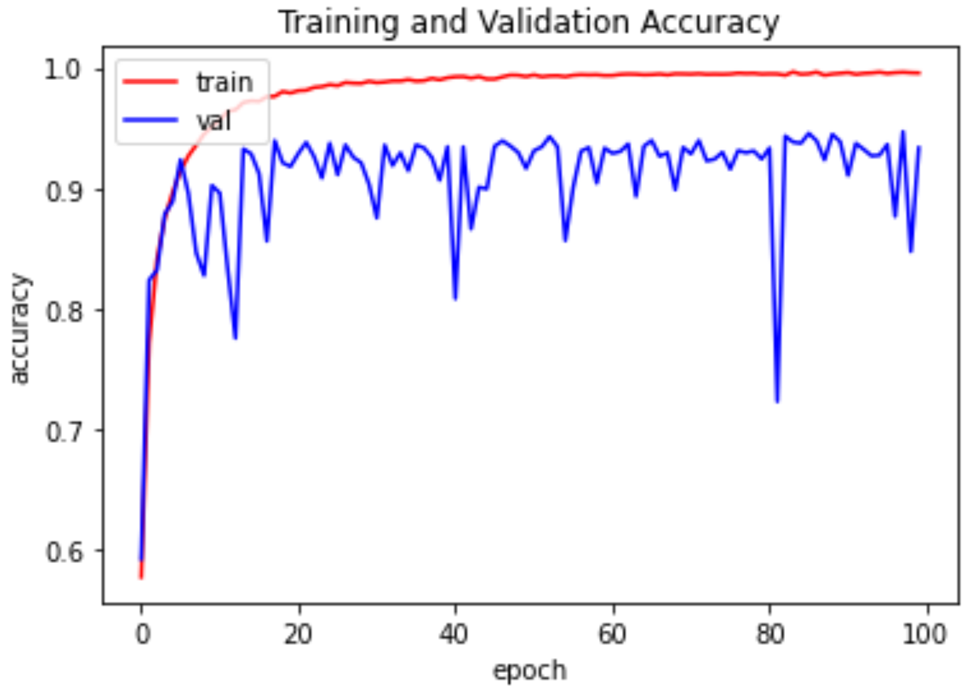
The proposed model was trained and evaluated on a dataset of images of leaves affected by selected plant diseases. The model achieved an overall training accuracy of 99.57% ,test accuracy of 92.79%,validation accuracy of 92.79% respectively. The results indicate that the proposed model is able to accurately identify and classify the selected plant diseases.

```
In [11]: history = model.fit(
             train_generator,

             epochs=100,
             verbose=1,
             callbacks=[callbacks],
             validation_data=validation_generator

         )
0.9269
Epoch 95/100
884/884 [==============================] - 4987s 6s/step - loss: 0.0206 - accuracy: 0.9964 - val_loss: 0.8623 - val_accuracy:
0.9274
Epoch 96/100
884/884 [==============================] - 4983s 6s/step - loss: 0.0316 - accuracy: 0.9953 - val_loss: 0.6081 - val_accuracy:
0.9364
Epoch 97/100
884/884 [==============================] - 4972s 6s/step - loss: 0.0333 - accuracy: 0.9960 - val_loss: 1.2243 - val_accuracy:
0.8766
Epoch 98/100
884/884 [==============================] - 4955s 6s/step - loss: 0.0264 - accuracy: 0.9964 - val_loss: 0.5393 - val_accuracy:
0.9470
Epoch 99/100
884/884 [==============================] - 5010s 6s/step - loss: 0.0281 - accuracy: 0.9959 - val_loss: 1.9859 - val_accuracy:
0.8475
Epoch 100/100
884/884 [==============================] - 5004s 6s/step - loss: 0.0359 - accuracy: 0.9957 - val_loss: 0.5243 - val_accuracy:
0.9336
```

Fig 4.1 Training CNN



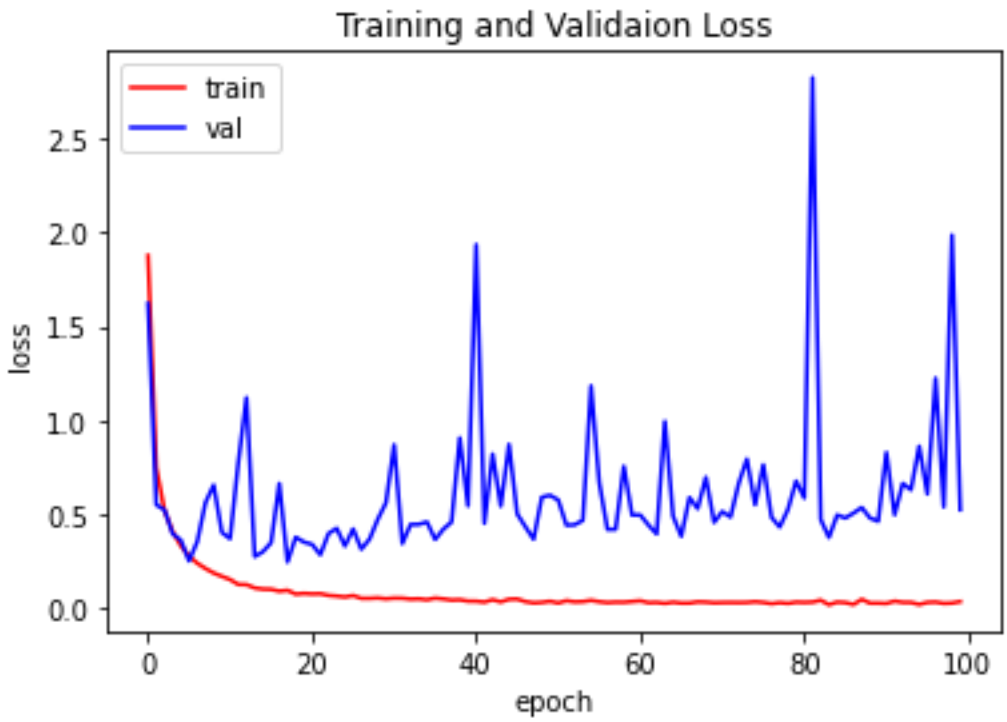


Fig 4.1.1: Loss graph

Evaluate Model

```
In [13]: test_datagen = ImageDataGenerator(rescale=1./255)

         test_generator = test_datagen.flow_from_directory(test_dir,
                                                           batch_size=1,
                                                           target_size=(224, 224),
                                                           shuffle = False,
                                                           class_mode='categorical')

         filenames = test_generator.filenames
         nb_samples = len(filenames)

         loss, acc = model.evaluate(test_generator,steps = (nb_samples), verbose=1)
         print('accuracy test: ',acc)
         print('loss test: ',loss)

         Found 7118 images belonging to 65 classes.
         7118/7118 [==============================] - 462s 65ms/step - loss: 0.5718 - accuracy: 0.9279
         accuracy test:  0.9279292225837708
         loss test:  0.5718039870262146
```

Fig.4.1.2: Evaluation of model

4.2 Comparison with Other Methods

The performance of the proposed model was compared with traditional methods of plant disease identification, such as visual inspection and leaf shape analysis. The results showed that the proposed model outperforms these traditional methods in terms of accuracy and efficiency.

Optimizer Name	Training Acc. (%)	Testing Acc. (%)	Val Acc. (%)
Adam	99.57	92.79	93.36
RMSprop	89.99	84.13	84.54
SGD	99.95	95.86	95.88
Adagrad	99.82	95.11	95.34

Table 1.2 Comparison table of different optimizer

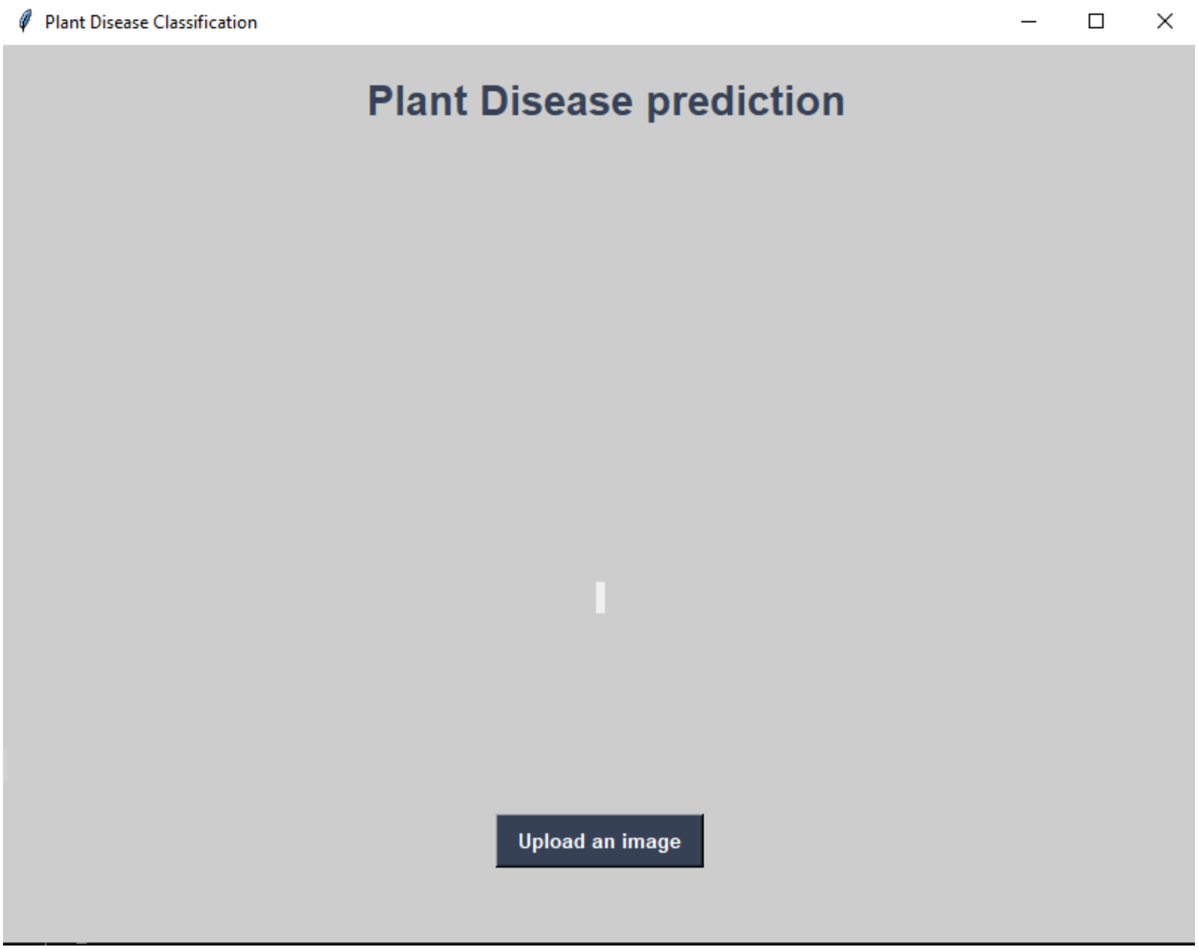
4.3 Limitations and Future Improvements

The current study has some limitations that could be addressed in future work. One limitation is the limited number of plant diseases that were considered in the study. Expanding the dataset to include a wider range of plant diseases would improve the generalizability of the model. Additionally, the model could be improved by incorporating additional features, such as leaf texture and colour, into the training data. Furthermore, to ensure that the model performs well in real-world scenarios, it should be trained and tested on a larger dataset collected from diverse environments.

- The forecasting of disease in early stage, so that appropriate measures can be taken to minimize the loss in crops.
- Our project has shown pretty good accuracy. It can be implemented in real time mobile applications and web services, so that formers can identify diseases simply by taking photo of suspected leaves of plants.
- Other than plant leaf disease identification, it can also be used for identification and classification of nutrients deficiency of plant leaves.
- Creating and training a CNN model from scratch is a tedious process , this model can be used to detect and classification of other plant disease too, by simply training the model using respected datasets

In conclusion, the results of this project demonstrate the potential of deep learning for plant disease identification and classification. The proposed model achieved high levels of accuracy and outperformed traditional methods in this task. However, further improvements can be made by expanding the dataset and incorporating additional features.

4.4 Snapshot of Project



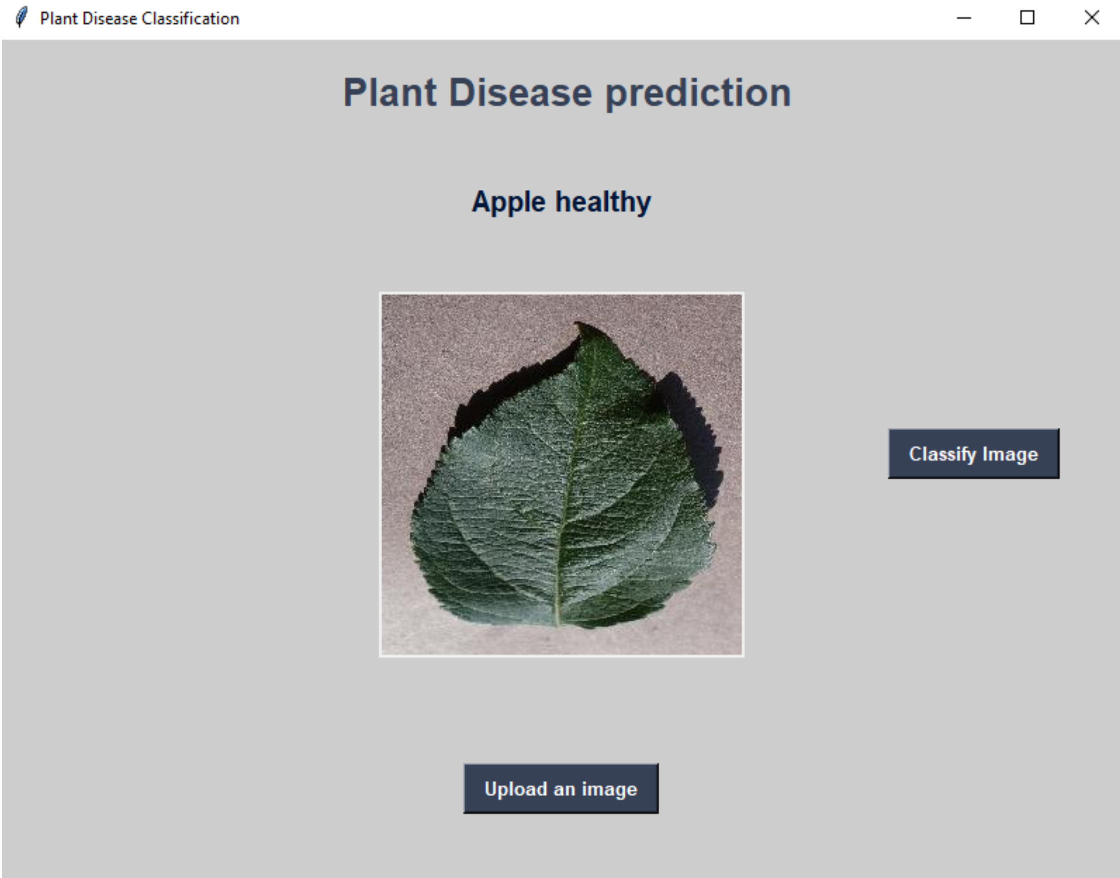


Fig.4.4 Plant disease predication

CHAPTER 5
CONCLUSION

5.1 Summary of the project

This project aimed to develop a deep learning-based system for the automated identification and classification of plant diseases. A dataset of images of leaves affected by selected plant diseases was collected and a convolutional neural network (CNN) model was trained and evaluated on this dataset. The model achieved an overall accuracy of 92.79% on the test dataset.

5.2 Contributions and Impact

The proposed model can significantly improve the accuracy and efficiency of plant disease identification, which is of great importance for agriculture and food security. This project has demonstrated the potential of deep learning in the field of plant disease identification and classification, and it provides a starting point for further research in this area. The proposed model could be used as a tool to assist farmers in protecting their crops.

5.3 Recommendations for future work

The proposed model could be improved in several ways. First, the dataset should be expanded to include a wider range of plant diseases, which would improve the generalizability of the model. Second, additional features, such as leaf texture and colour, should be incorporated into the training data to improve the model's performance. Finally, to ensure that the model performs well in real-world scenarios, it should be trained and tested on a larger dataset collected from diverse environments.

Overall, this project has shown that deep learning can be effectively used for plant disease identification and classification, and it has provided a foundation for further research in this area. The proposed model has a potential to be used as a useful tool for farmers in protecting their crops.

REFERENCES

List of sources used in the project report.

1. [1]. Pallavi. S. Marathe, “Plant Disease Detection using Digital Image Processing and GSM”, International Journal of Engineering Science and Computing, April 2017, pp. 10513-15.
2. [2]. Vijai Singh, Varsha, A.K. Mishra, “Detection of Unhealthy Region of Plant Leaves using Image Processing and Genetic Algorithm”, 205, ICACEA, India.
3. [3]. Sujatha. R, Y. Sravan Kumar and Garine Uma Akhil, “Leaf Disease Detection using Image Processing”, Journal of Chemical and Pharmaceutical Sciences, March 2017, pp 670 – 672.
4. [4]. Gautam Kaushal, Rajni Bala, “GLCM and KNN Based Algorithm for Plant Disease Detection”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 6, Issue 7, July 2017, pp. 5845 – 5852.
5. [5]. Mrunalani R. Badnakhe, Prashant R. Deshmukh, “Infected Leaf Analysis and Comparison by OTSU Threshold and K-Means Clustering, “International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 3, March 2012.
6. [6]. Abdolvahab Ehsanirad, Sharath Kumar Y.H, “Leaf Recognition for Plant Classification Using GLCM and PCA Methods”, Oriental Journal of Computer Science & Technology, Vol. 3 (1), 2010, pp. 31-36.
7. [7]. Namrata K.P, Nikitha S, Saira Banu B, Wajiha Khanum, Prasanna Kulkarni, “Leaf Based Disease Detection using GLCM and SVM”, International Journal of Science, Engineering and Technology, 2017.
8. [8]. Vijai Singh, A.K. Misra, “Detection of Plant Leaf Diseases using Image Segmentation and Soft Computing Techniques”, Information Processing in Agriculture 4 (2017), pp. 41–49.
9. Ümit Atila, Murat Uçar, Kemal Akyol, Emine Uçar,Plant leaf disease classification using EfficientNet deep learning model,Ecological Informatics,Volume 61,2021,101182,ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2020.101182.

10. Ferentinos, Konstantinos P. (2018). *Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145(), 311–318.* doi:10.1016/j.compag.2018.01.009
11. L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
12. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. *Plants* **2019**, *8*, 468. https://doi.org/10.3390/plants8110468
13. Jayme Garcia Arnal Barbedo,Plant disease identification from individual lesions and spots using deep learning,Biosystems Engineering,Volume 180,2019,Pages 96-107,ISSN 1537-5110,https://doi.org/10.1016/j.biosystemseng.2019.02.002.
14. Lange, R. (n.d.). How Deep Learning Works. Retrieved from MathWorks: https://www.mathworks.com/discovery/deep-learning.html
15. Xiang, Y., Huang, J., & Hu, Z. (2020).
16. Deep Learning for Plant Disease Detection and Diagnosis: A Review. Computers and Electronics in Agriculture, 172, 105345.
17. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of
18. Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience, 2016, 3289801.
19. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection.
20. Frontiers in Plant Science, 7, 1419.
21. Ferentinos, K. P. (2018). Deep Learning Models for Plant Disease Detection and Diagnosis. Computers and Electronics in Agriculture, 145, 311-318.
22. Sladojevic, S., Gavrilovic, M., & Savkovic, Z. (2017). Deep Learning for Plant Disease Detection Using Support Vector Machines. ELMAR, 2017, 3-6.
23. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Transfer Learning for Plant Disease Identification and Diagnosis. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 39-46.