A Project Report on

PREDICTION OF DISEASE IN APPLE LEAF USING CONVOLUTION NEURAL NETWORKS

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

B.RAVICHANDRA
(20H51A05D7)
P.SHIRISHA
(20H51A0572)

Under the esteemed guidance of
Dr.S.KIRUBAKARAN
(Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A+ Grade
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **" Prediction of Disease in Apple Leaf Using Convolution Neural Networks"** being submitted by **B. Ravichandra (20H51A05D7), P. Shirisha(20H51A0572),** in partial fulfillment for the award **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide workcarried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr.S.Kirubakaran
Professor
Dept. of CSE

Dr.Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure, we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Dr. S . Kirubakaran, Professor,** Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr.Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and **Shri Ch Abhinav Reddy**, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

B.Ravichandra 20H51A05D7
P.Shirisha 20H51A0572

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Plant diseases are a major cause of crop losses worldwide. The absence of specialist knowledge makes the detection of plant diseases difficult and challenging. Models based on deep learning offer promising methods to identify plant diseases using leaf images. However, the need for larger training sets, computational complexity, overfitting, etc. are the main issues that still need to be resolved with these techniques. In work, the paper proposes a convolutional neural network (CNN) model that uses augmentation techniques such as shift, shear, scaling, zoom, and flipping to generate additional samples and increase the training set without actually capturing more images. The proposed model is trained for apple crop using a publicly available dataset PlantVillage to identify Scab, Black rot, and Cedar rust diseases in apple leaves. The experimental results show that the proposed model achieves good classification accuracy and needs lesser storage and computational resources than several existing deep CNN models. The proposed model is highly suitable for deploying in handheld devices.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

Agribusiness is one of the essential sources of the subsistence of people for many decades. It contributes in the nourishment of about fifty percent of the global population. Agriculture has an immense impact on people globally, either directly or indirectly. Apple is one of the most widely consumed fruit globally and among the four most produced fruits after banana, grape, and orange. The production of apple crop has increased in the last decade, but it is not in proportion to the growth of the cultivation area.Globally, pests and diseases affect the overall production of apple crop. In India, fungal diseases are one of the major causes affecting the quality of apple fruits in Himachal Pradesh, which is the second highest producer state of apple fruits. Infections in plants are categorized into two categories- biotic and abiotic. The pathogens such as virus, fungi, and bacteria are the infectious agents responsible for biotic diseases.Some specific visible patterns are generated by responsible pathogen for most of the infections. Advancements in image processing, artificial intelligence, and computational resources like graphical processing unit (GPU) can reform the process of detection and prevention of plant diseases.

1.1.Problem Statement

Apple cultivation faces significant challenges due to various diseases that affect the health and yield of the crop. Early detection and accurate diagnosis of these diseases are crucial for effective disease management and prevention of yield losses.

1.2.Research Objective

The research objective of this study is to address the challenges associated with the detection of plant diseases by developing an efficient convolutional neural network (CNN) model utilizing augmentation techniques. Specifically, the aim is to enhance the classification accuracy for identifying Scab, Black rot, and Cedar rust diseases in apple leaves using leaf images from the publicly available dataset PlantVillage. By leveraging augmentation methods such as shift, shear, scaling, zoom, and flipping, the study aims to expand the training set without increasing the number of captured images, thereby mitigating the need for larger datasets. Furthermore, the objective is to demonstrate that the proposed model requires fewer computational resources and storage compared to existing deep CNN models while maintaining high classification accuracy. Ultimately, the goal is to offer a practical solution suitable for deployment on handheld devices for real-time plant disease detection.

1.3 Project Scope and Limitations

Project Scope:

The scope of this project involves the development and implementation of a convolutional neural network (CNN) model for the identification of plant diseases in apple crops using leaf images. The focus will be on utilizing augmentation techniques such as shift, shear, scaling, zoom, and flipping to expand the training dataset and enhance the model's ability to accurately classify diseases. The model will specifically target three common diseases affecting apple trees: Scab, Black rot, and Cedar rust. The dataset utilized for training and testing purposes will be PlantVillage, a publicly available repository of plant images.

The project aims to address several key challenges associated with traditional methods of plant disease detection, such as the lack of specialist knowledge and the difficulty in accurately identifying diseases based on visual symptoms alone. By leveraging deep learning techniques, particularly CNNs, the proposed model seeks to provide a more efficient and accurate solution for disease identification. Additionally, the project will focus on optimizing the model's performance while minimizing storage and computational resource requirements, making it suitable for deployment on handheld devices.

Limitations:

Despite its promising potential, the proposed CNN model for plant disease identification has certain limitations that need to be acknowledged. Firstly, the effectiveness of the model may be constrained by the quality and diversity of the training dataset. While augmentation techniques can help mitigate this issue to some extent by generating additional samples, the overall performance of the model could still be limited by the variability in real-world conditions and disease manifestations.

Furthermore, the generalization capability of the model may be limited to the specific diseases and crops targeted in this project (i.e., Scab, Black rot, and Cedar rust in apple trees). Extending the model to other crops or diseases may require additional training data and fine-tuning of the model architecture, which could be a significant undertaking beyond the scope of this project.

Moreover, while efforts will be made to optimize the model for deployment on handheld devices, there may still be constraints in terms of computational resources and processing power. The model's performance on such devices may be impacted by hardware limitations, potentially affecting its real-time application in the field.

Lastly, it's important to note that while the proposed CNN model can aid in the identification of plant diseases, it should not be considered a substitute for expert diagnosis and consultation. The model's predictions should be interpreted with caution, and human expertise should be utilized for confirmation and further analysis when necessary.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Plant Disease Detection Using Deep Convolutional Neural Network

2.1.1 Introduction

Plant diseases pose a significant threat to global food security, causing substantial crop losses and impacting agricultural productivity. Traditional methods for detecting and diagnosing plant diseases often rely on visual inspection by trained experts, which can be time-consuming, labor-intensive, and prone to errors. With the advent of computer vision and machine learning techniques, there has been growing interest in developing automated systems for the detection and classification of plant diseases using leaf images.[1]

Deep learning, particularly convolutional neural networks (CNNs), has emerged as a promising approach for automated plant disease diagnosis. CNNs are well-suited for image classification tasks, as they can automatically learn hierarchical features directly from raw pixel data. However, several challenges hinder the widespread adoption of deep learning models for plant disease detection.[1]

One of the primary challenges is the availability of large and diverse training datasets. Deep learning models typically require a substantial amount of labeled data for effective training. In the context of plant disease detection, obtaining annotated images for training can be difficult and expensive, especially for rare or localized diseases. Furthermore, deep learning models are prone to overfitting, where the model learns to memorize the training data rather than generalize to unseen examples. This is particularly problematic in scenarios with limited training data, as the model may fail to generalize well to new samples.[1]

To address these challenges, researchers have explored various strategies, including data augmentation techniques. Data augmentation involves generating synthetic training samples by applying transformations such as rotation, translation, scaling, and flipping to existing images. By augmenting the training dataset, researchers can increase its diversity and size, which can help improve the generalization performance of the model and mitigate overfitting.

In this work, the authors propose a CNN model for the detection of apple diseases using leaf images. They leverage augmentation techniques such as shift, shear, scaling, zoom, and flipping to generate additional training samples, thereby expanding the training dataset without the need for capturing more images. By doing so, they aim to address the challenges of limited data availability and overfitting while reducing the computational complexity and storage requirements of the model.

Overall, the proposed approach represents a significant advancement in the field of automated plant disease detection, offering a practical solution that is well-suited for deployment on handheld devices and other resource-constrained environments.[1]

2.1.2. Merits, Demerits and Challenges

- In existing works they didn’t apply the data augmentation technique to increase the training data.
- Pre-trained models are showing less performance compare to specially trained CNNs.
- The CNNs which are having more hidden layers takes long time two train and also require more computational resources.

2.2.3. Implementation of Plant Disease Detection Using Deep CNN

Existing methods for plant disease detection often rely on deep learning models due to their ability to learn complex patterns from images. However, employing deep CNN models with numerous layers poses challenges due to computational complexity and potential overfitting. Researchers have explored using pre-trained models like AlexNet for plant disease prediction, leveraging their learned features. Despite their success in various image recognition tasks, pre-trained models may not excel in detecting specific diseases like apple scab without fine-tuning for this purpose. Moreover, increasing the number of layers in CNNs exacerbates computational burdens during training and inference.[1]

To address these issues, a pragmatic approach involves adapting pre-trained models for plant disease detection through transfer learning. By fine-tuning models on plant disease datasets, researchers can enhance their specificity and performance for this task while mitigating computational demands compared to training from scratch. Additionally, model architectures with moderate depth can strike a balance between computational efficiency and predictive accuracy, making them more practical for real-world deployment in agricultural settings and helps farmers in avoiding their loss.[1]

2.2 Deep Leaf Disease Prediction Framework (DLDPF) with Transfer Learning for Automatic Leaf Disease Detection

2.2.1 Introduction

Deep learning models became an attractive and efficient alternative for leaf disease detection when compared with traditional models. The rationale behind this is that the deep models could handle large datasets and support for pre-trained models. For image-based detection of diseases and classification using leafs as input, deep learning models explored different crops in agriculture. This study has assumed significance in the wake of Precision Agriculture (PA) efforts across the globe. Technology driven approach in detection of crop diseases lead to innovations in early identification of problems in agricultural crops and take necessary steps. Many researchers contributed towards these using deep learning models.[2]

Traditional methods of leaf disease detection often rely on manual inspection by agricultural experts, which can be labor-intensive, time-consuming, and subject to human error. With the advancements in computer vision and deep learning techniques, automated approaches for disease detection have garnered increasing interest due to their potential to revolutionize agricultural practices.[2]

Most of the research papers use Convolutional Neural Network (CNN) architectures for deep learning-based disease detection. CNN is used in Maize disease detection based on leaves in Maize plants. There are some research pertaining to leaf disease datasets and the impact of size as explored in Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks. Deep CNN is used in rice disease prediction using rice leaves. It is understood that the existing methods are based on CNN for deep learning. However, there is a need for novel architectures with pre-trained models and the existing models have not used transfer learning. This model uses transfer learning with a deep learning framework with pre-trained deep models to classify diseases of Apple crops. There are four diseases such as, (a) leaf spots lesions are one kind of disease, (b) the yellow color lesion on the leaf is called Mosaic disease, (c) the yellow color spot on the leaf is the symptom of Rust disease, and (d) Brown spot disease. Overall, the proposed approach represents a significant advancement in the field of plant disease detection.[2]

2.2.2. Merits, Demerits and Challenges

- By leveraging transfer learning, DLDPF can achieve high levels of accuracy in detecting leaf diseases, surpassing traditional methods and generic deep learning approaches.
- The framework is computationally efficient, making it suitable for deployment in resource-constrained environments such as farms and field settings.
- Poor-quality or biased datasets may lead to inaccurate predictions and reduced generalization performance.
- Annotating large-scale datasets for training DLDPF can be time-consuming and labor-intensive, especially when dealing with diverse crop species and disease types.

2.2.3 Implementation of Deep Leaf Disease Prediction Framework (DLDPF) with Transfer Learning for Automatic Leaf Disease Detection

A framework is proposed to perform automated leaf disease detection using deep learning models. The framework is known as Deep Leaf Disease Prediction Framework (DLDPF).It is designed in such a way that it exploits pre-trained models besides transfer learning to leverage disease prediction accuracy. Instead of using baseline CNN models directly, the DLDPF has its own mechanism to improve training optimization and testing accuracy. It also exploits transfer learning by considering two pre-trained models such as AlexNet and GoogLeNet. The DLDPF framework has important phases such as training enhancement, construction of CNN, fitting the model with AlexNet precursor and then fitting the model with cascade inception using GoogLeNet besides transfer learning. AlexNet precursor enables larger sized convolution kernel so as to support a strong capability to obtain information from leaf image. The transfer learning enables performance improvement by reusing the model instead of reinventing the wheel. The ResNet model is used for empirical study and it is eventually compared with the proposed framework. In this process, the pre-processing and segmentation provide betterment in the processing of CNN. The deep CNN model is designed in such a way that it takes advantages of the AlextNet model and also the inception layers of the GoogLeNet model. It is a tailored combination of different layers such as convolution, pooling, activation function, and softmax layers. The parameters used to build the model. The results of this model revealed that the DLDPF outperforms the state-of-the-art deep learning models for automated prediction of leaf diseases.[2]

Leaf disease detection and classification is realized by proposing an algorithm named Cascade Inception-based Deep CNN with Transfer Learning (CIDCNN-TL). It takes Apple leaves as input (training and testing) besides batch size and number of epochs.It performs deep learning on the training set as per the configurations. It makes use of cascade inception along with convolutional layers and max pooling layers in order to perform the prediction of diseases. Once the model is built, it gets updated in each epoch until convergence. Cascade inception is built in such a way that it has two inception structures and two max-pooling layers. AlexNet precursor generates feature maps to help the first max-pooling layer to filter out noise. Afterwards, the inception layers are able to identify discriminating features. A total of six convolutional layers are used with different kernel size. The fully connected layer is configured to identify four kinds of Apple leaf diseases.[2]

2.3 MobileNet Based Apple Leaf Diseases Identification

2.3.1 Introduction

With a high nutritional and medicinal value, apple is one of the most productive types of fruit in the world. However, various diseases occur frequently on a large scale in apple production, such as Apple Alternaria leaf blotch (caused by Alternaria alternata f.sp. mali), and Apple rust (caused by Pucciniaceae glue rust), which affect the quality of fruits and thereby causing substantial economic losses. Currently, the apple leaf diseases are mainly inspected by experienced experts. They need to check the apple leaves one by one. This is a huge job. The number of leaves for one apple tree is large enough. For a whole apple yield, we do not have enough experienced experts to finish such kind of inspection task. Furthermore, a large number of errors will be appeared when these experts become tired, especially for some similar leaf diseases. Therefore, we need an algorithm to help farmers to resolve this problem. This algorithm can let non-experience-farmers to identify these apple leaf diseases without the helps from experts.[3]

A mobile-based model is built for apple leaf diseases identification based on MobileNet. This model is a mobile version CNN model. Its precision is nearly the same as the general CNN model. Meanwhile, its efficiency is high enough for apple leaf diseases identification. In this method, firstly, in order to let the disease identification as STABLE as possible, with the help of agriculture experts from Chinese Academy of Agricultural Sciences, China for obtaining all kinds of leaves with different apple leaf diseases. These datasets are taken from Shaanxi Province, China. Note that, this model have mainly used two kinds of diseases (Alternaria leaf blotch and Apple rust) to demonstrate the effectiveness of our proposed method. Secondly, in order to satisfy the LOWCOST issue, the MobileNet model is employed for apple leaf diseases identification. This is because it can be easily deployed on mobile devices. Finally, in order to achieve the goal of HIGH efficiency and precision, the MobileNet model is optimized according to the features of apple leaf diseases. Furthermore, we have also tried several other models. As a result, the MobileNet model is the best choice. A balance of efficiency and precision is achieved by using MobileNet for apple leaf diseases identification.[3]

2.3.2. Merits, Demerits and Challenges

- Allows for real-time inference and on-device processing, eliminating the need for constant internet connectivity.
- Due to its lightweight architecture, MobileNet can achieve fast inference speeds, enabling rapid identification of apple leaf diseases in the field.
- Limited Capacity may affect its ability to capture intricate features and nuances present in apple leaf images, potentially leading to lower accuracy in disease identification.
- The performance of MobileNet-based apple leaf disease identification heavily relies on the quality, diversity, and representativeness of the training data. Obtaining annotated datasets with a wide variety of apple leaf diseases, environmental conditions, and disease severities can be challenging and time-consuming.

2.3.3 Implementation of MobileNet-Based Apple Leaf Diseases Identification

Implementing MobileNet for Apple Leaf Diseases Identification begins with the collection and preparation of a diverse dataset of apple leaf images, encompassing various disease states and apple varieties. These images are annotated with corresponding labels indicating the presence of specific diseases, and the dataset is then split into training, validation, and testing sets. Subsequently, preprocessing techniques such as resizing, normalization, and data augmentation are applied to prepare the images for training. MobileNet is chosen as the base architecture due to its lightweight design, making it suitable for deployment on mobile devices with limited computational resources.[3]

The training process involves initializing MobileNet with pre-trained weights on a large-scale dataset like ImageNet to leverage transfer learning. The output layer of MobileNet is replaced with a new set of output nodes representing different apple leaf disease classes. The model is trained on the training data using optimization techniques such as stochastic gradient descent, with hyperparameters adjusted to prevent overfitting. Throughout training, the model's performance is monitored on the validation set to ensure generalization to unseen data. Once training is complete, the trained MobileNet model is evaluated on the testing dataset to assess its accuracy and performance metrics.[3]

Upon successful training and evaluation, the deployed MobileNet model is integrated into a user-friendly application or platform accessible to farmers or agricultural experts. Users can input or upload apple leaf images for disease identification through an intuitive interface. The model predicts the disease class(es) present in the images along with confidence scores, empowering users to make informed decisions regarding disease management and crop protection. Continuous monitoring and maintenance of the deployed model are essential, with periodic updates and retraining conducted to adapt to evolving disease patterns and ensure sustained accuracy and reliability in real-world scenarios.[3]

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3
PROPOSED SYSTEM

3.1. Objective of Proposed System

The proposed method in the research paper is the development of a convolutional neural network (CNN) model that uses augmentation techniques such as shift, shear, scaling, zoom, and flipping to generate additional samples and increase the training set without actually capturing more images. The CNN model consists of 3 convolutional layers and two fully connected layers after the three max-pooling units. ReLU is explored as a nonlinear activation function at each convolution layer and at first dense layer. Softmax function is employed at the output layer to classify apple plant diseases. The softmax function is responsible for multiclass classification and assumes that each sample belongs to exactly one class. The proposed model is trained for apple crop using a publicly available dataset PlantVillage to identify Scab, Black rot, and Cedar rust diseases in apple leaves.

Advantages of the Proposed System:

1. We are using data augmentation techniques to increase the training set using existing data
2. ML based methods as it doesn't require any additional efforts for feature engineering.
3. The hyperparameters are tuned using random search technique that helps to select the best suitable hyperparameters.

3.2 Algorithms Used for Proposed Model

VGG19 – VGG-19 is a convolutional neural network that is 19 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

ResNet152 – ResNet-152 from Deep Residual Learning for Image Recognition. The bottleneck of TorchVision places the stride for downsampling to the second 3x3 convolution while the original paper places it to the first 1x1 convolution. This variant improves the accuracy and is known as ResNet V1. 5.

DenseNet201 – DenseNet-201 is a convolutional neural network that is 201 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [4].

MobileNetV2 – MobileNet-v2 is a convolutional neural network that is 53 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

ResNet50 – ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the neural network trained on more than a million images from the ImageNet database.

VGG16 – VGG-16 is a convolutional neural network that is 16 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [4]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

InceptionV3 – Inception v3 is an image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the culmination of many ideas developed by multiple researchers over the years.

Xception – Xception is a convolutional neural network that is 71 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [4]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

MobileNet – MobileNet is a computer vision model open-sourced by Google and designed for training classifiers. It uses depthwise convolutions to significantly reduce the number of parameters compared to other networks, resulting in a lightweight deep neural network. MobileNet is Tensorflow's first mobile computer vision model.

Deep CNN – Deep convolutional neural networks (CNN or DCNN) are the type most commonly used to identify patterns in images and video. DCNNs have evolved from traditional artificial neural networks, using a three-dimensional neural pattern inspired by the visual cortex of animals. CNN architectures can be scaled up to accommodate larger and more complex datasets by increasing the depth and width of the network. CNNs suitable for a wide range of applications, from simple image classification tasks to more challenging tasks.

YoloV5 – YOLO v5 uses a new method for generating the anchor boxes, called "dynamic anchor boxes." It involves using a clustering algorithm to group the ground truth bounding boxes into clusters and then using the centroids of the clusters as the anchor boxes.

YoloV8- YOLOv8 is the latest version of the YOLO algorithm, which outperforms previous versions by introducing various modifications such as spatial attention, feature fusion, and context aggregation modules.

3.3 Designing

System Architecture:

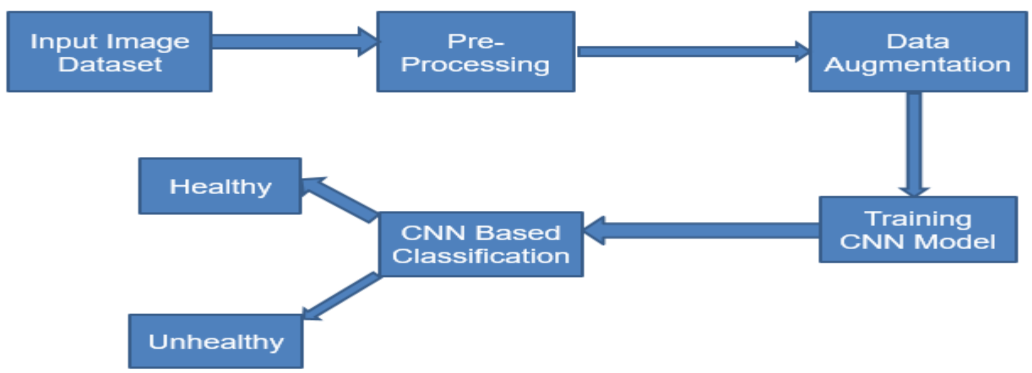


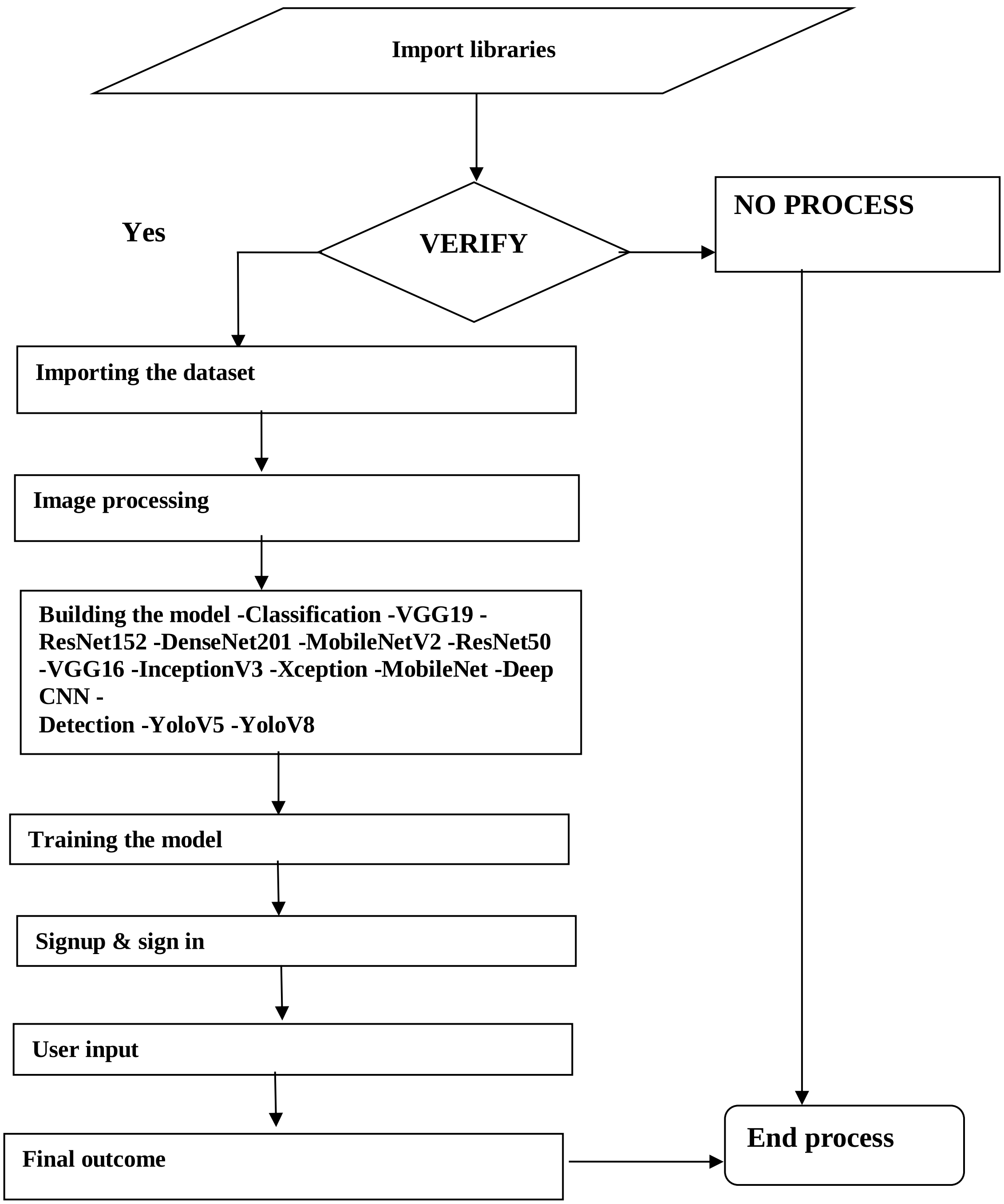
Figure 3.3.1 System Architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the

transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



3.3.1. UML Diagram

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

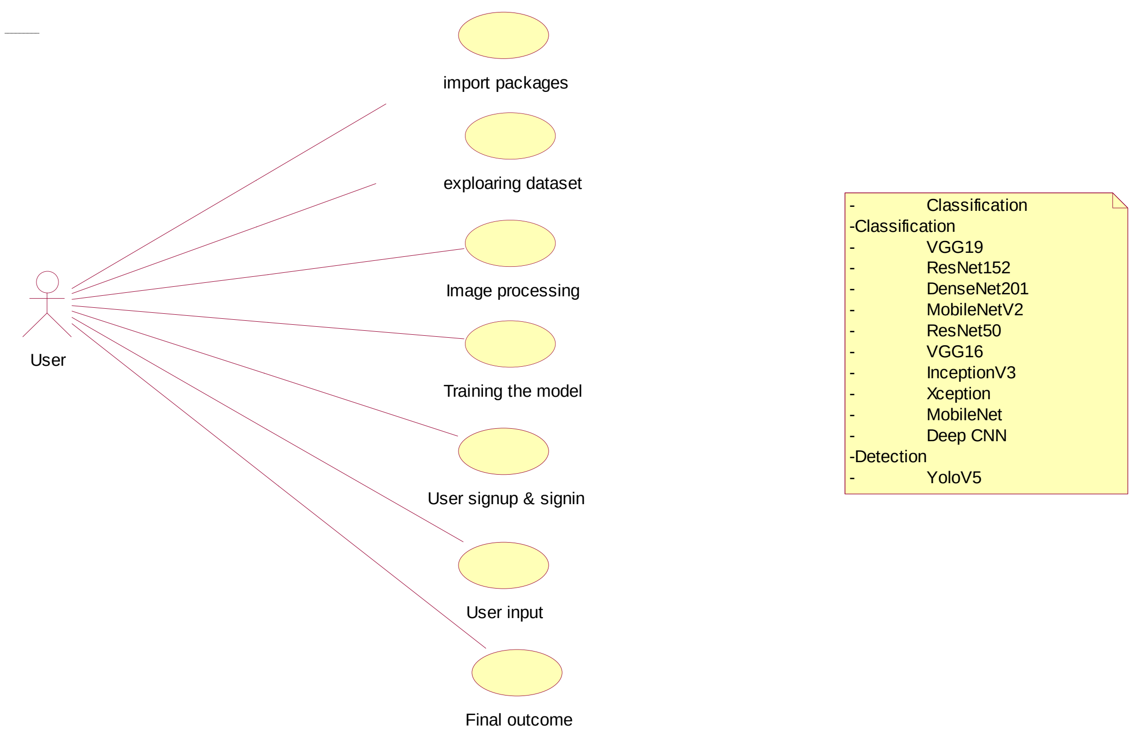
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.
8. Allows easy modification and evolution.
9. Focus on capturing information.
10. Avoid unnecessary complexity.

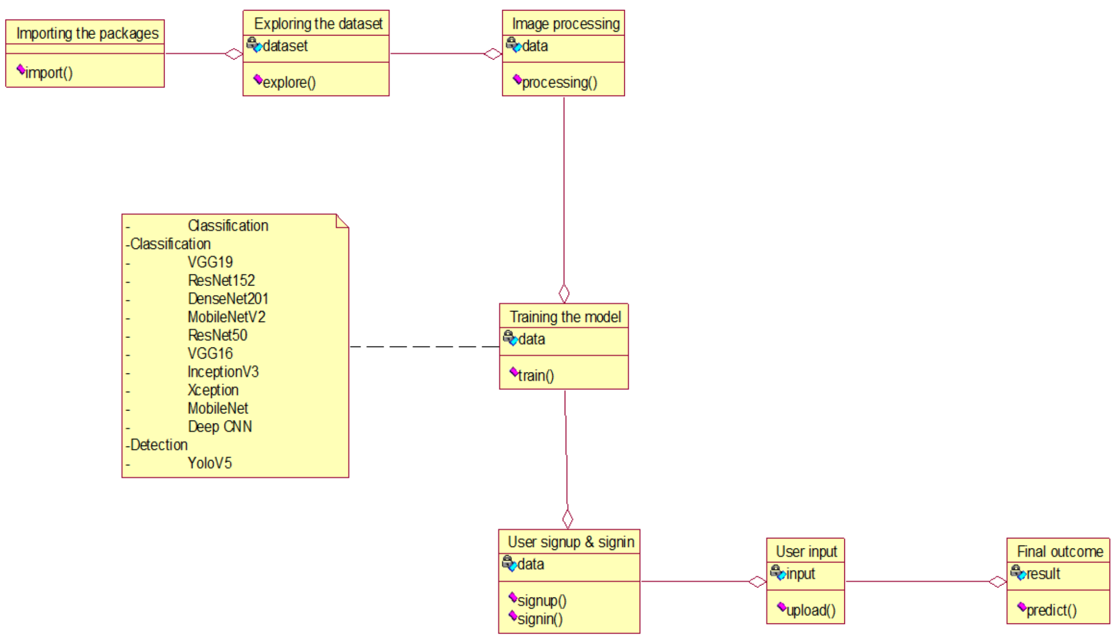
A. Use case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



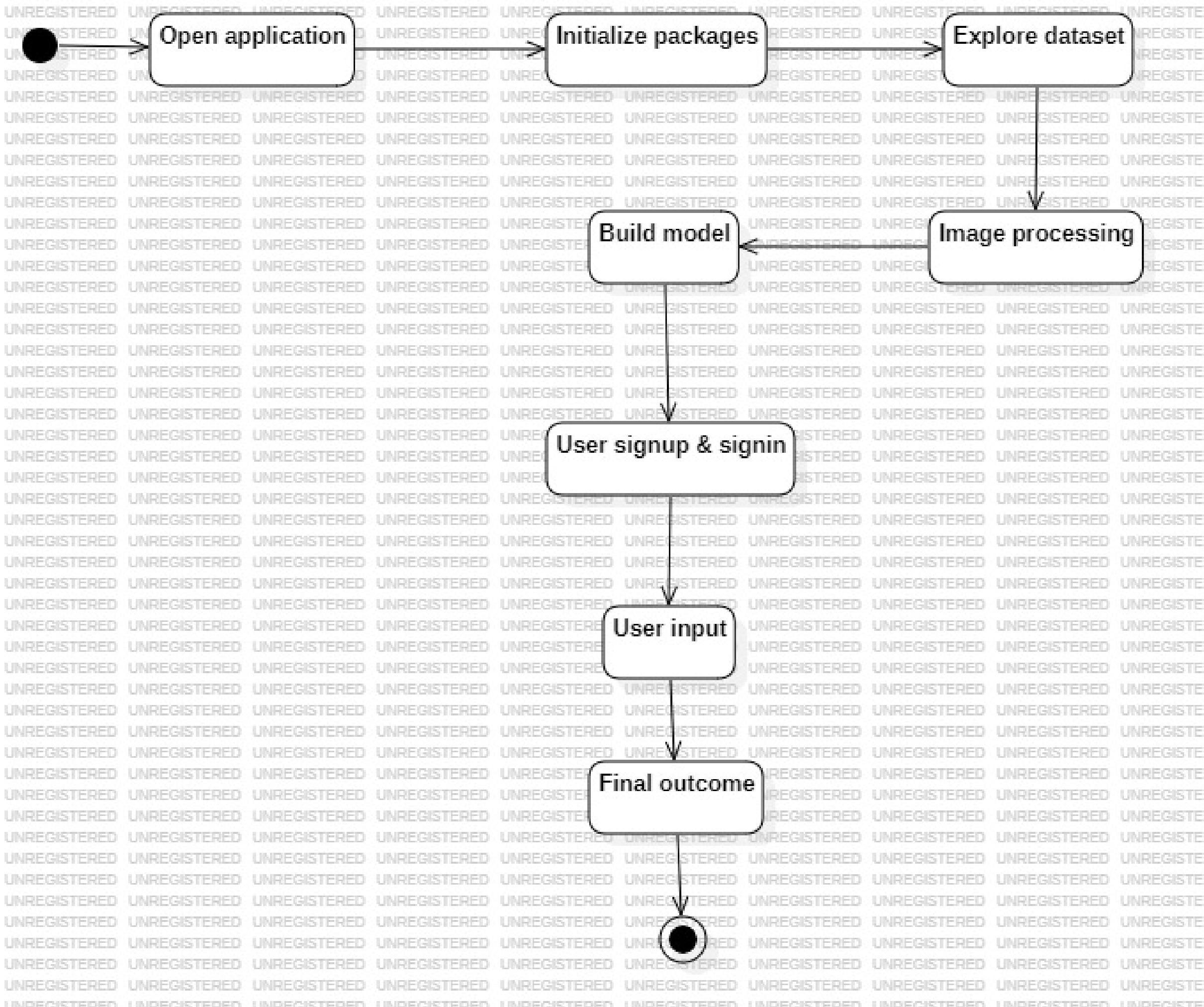
B. Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



C. Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

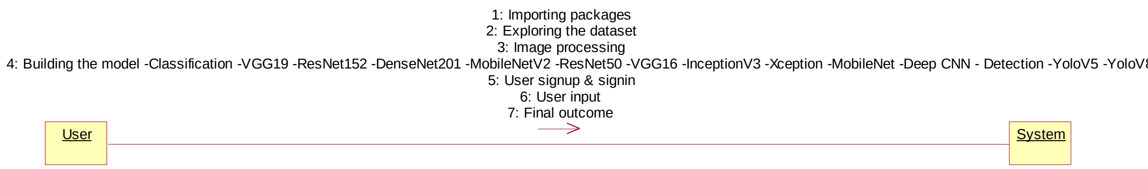


D. Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

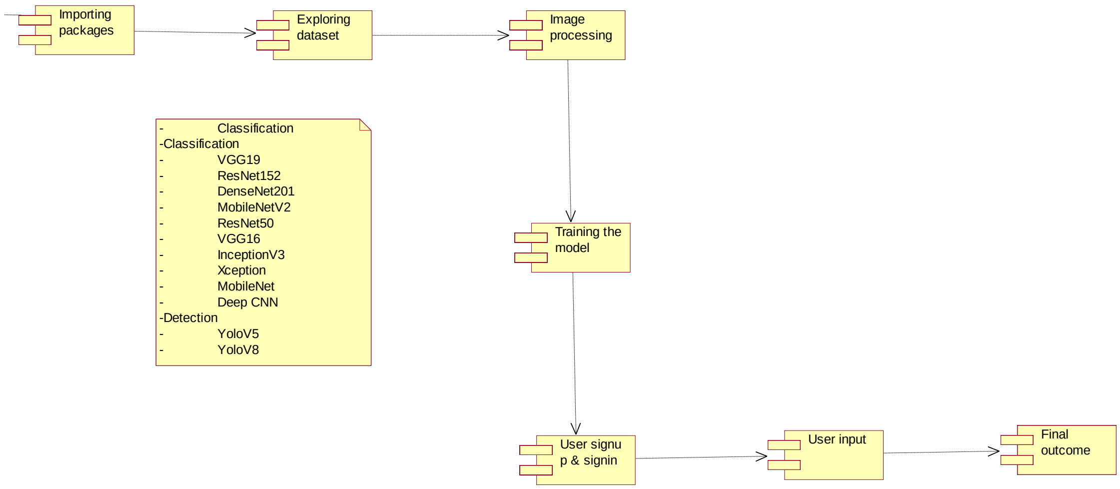
E. Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



F. Component diagram:

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



G. Deployment diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

3.3.2 Stepwise Implementation and Code

Implementation

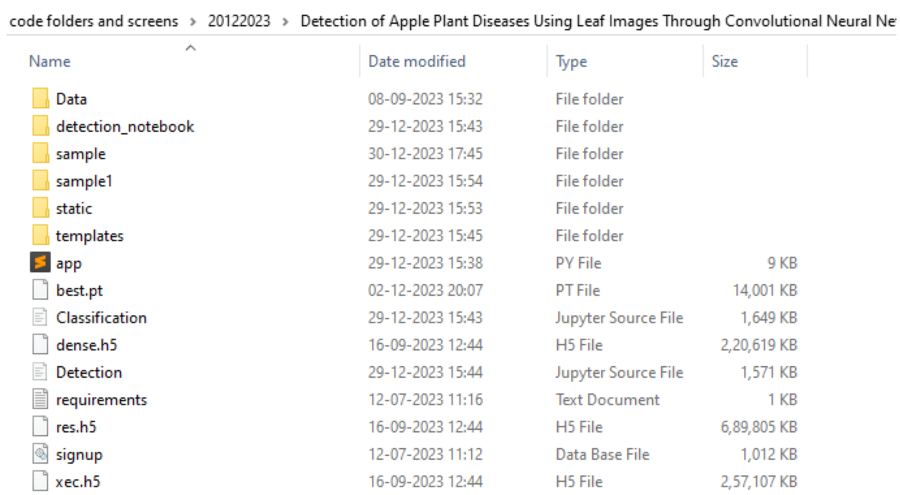


Fig:3.3.2.1 Open the code folder which contains project source code files

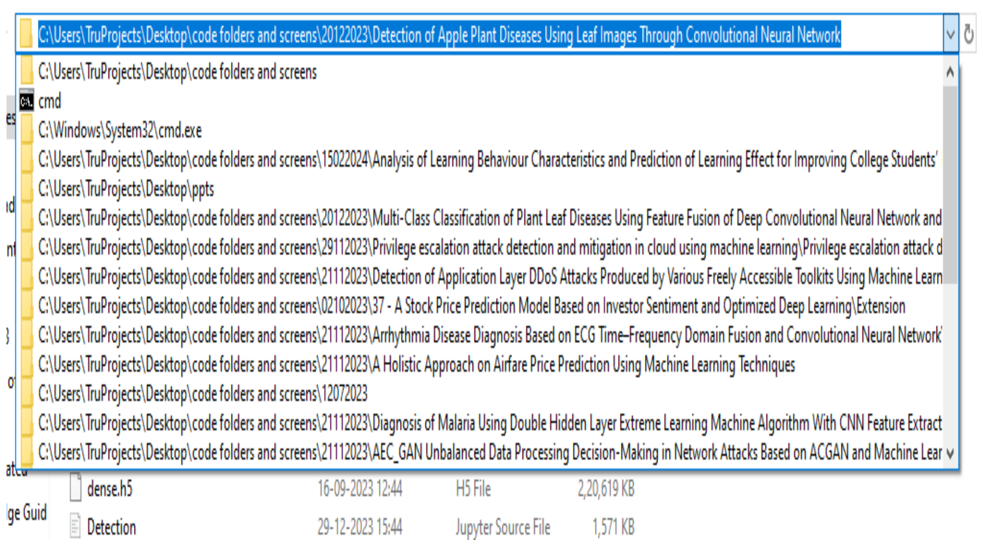


Fig 3.3.2.2 Copy the path of the code folder from the address bar

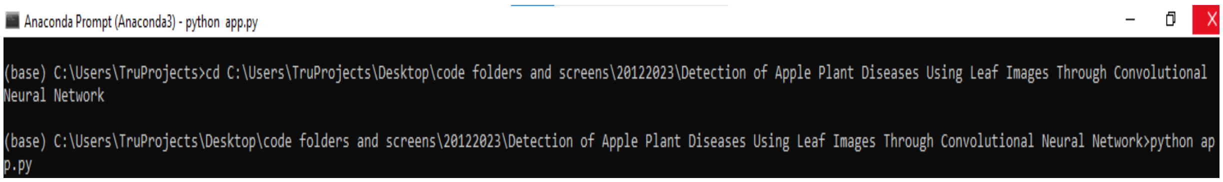


Fig 3.3.2.3 Open anaconda prompt, type cd and paste the path and run the app.py file

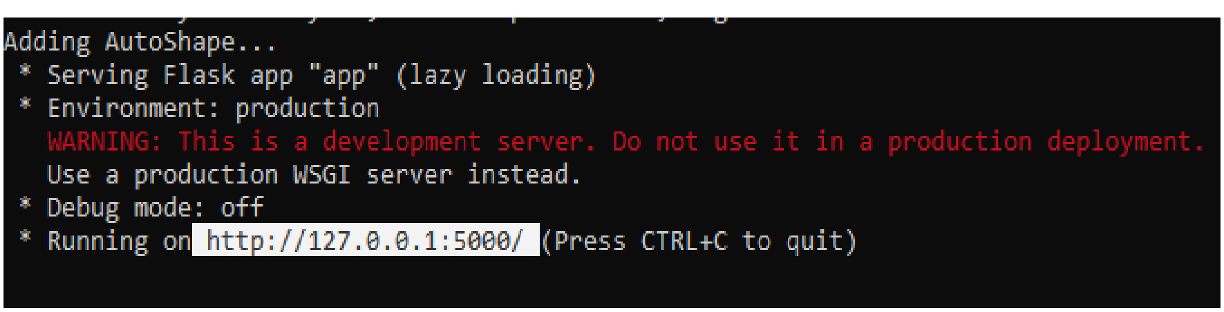


Fig 3.3.2.4 Copy the local link provided by the framework and paste it any web browser

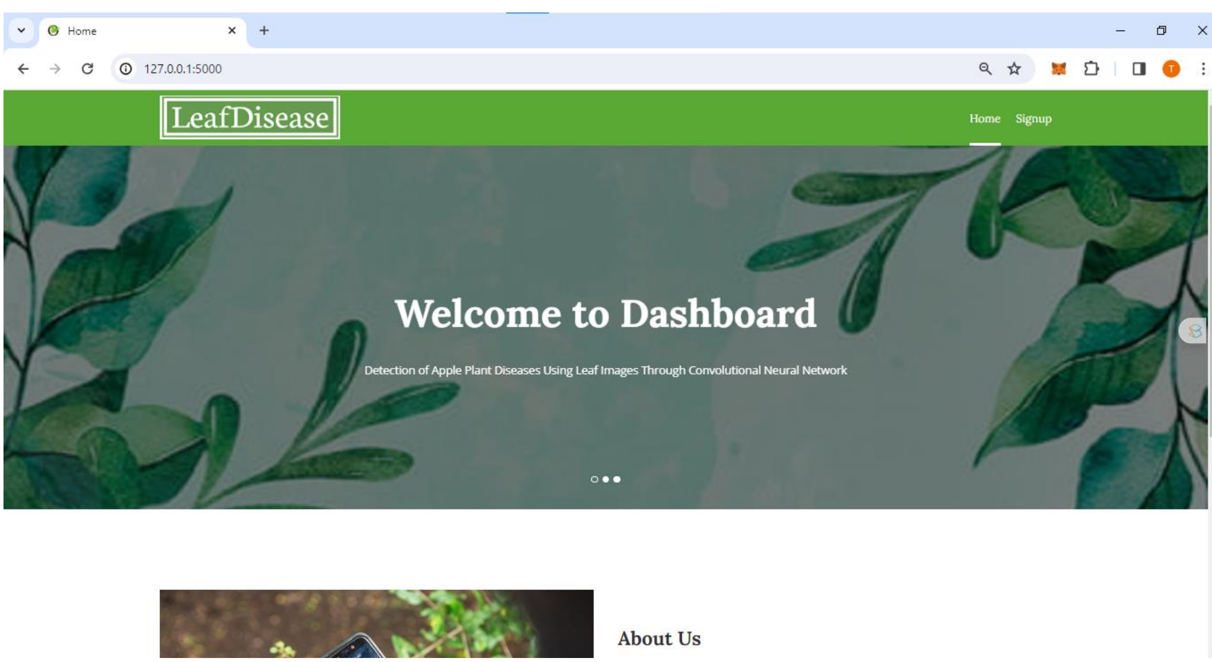


Fig 3.3.2.5 Home Page

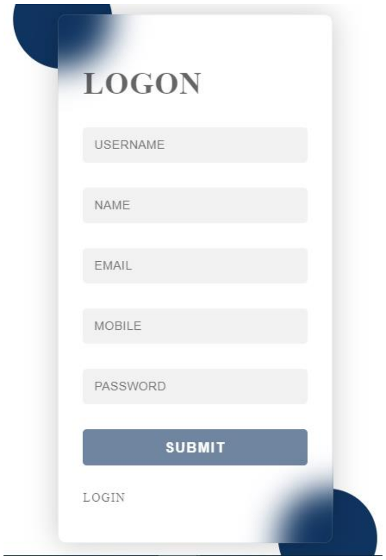


Fig 3.3.2.6 Signup page

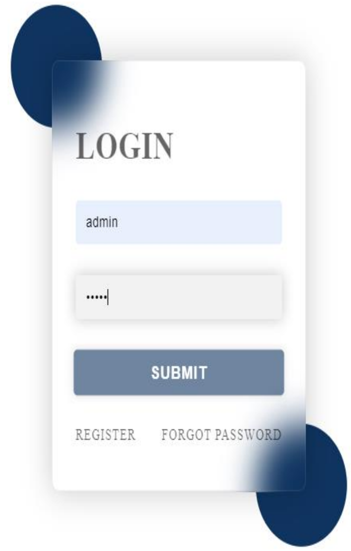


Fig 3.3.2.7 Login Page

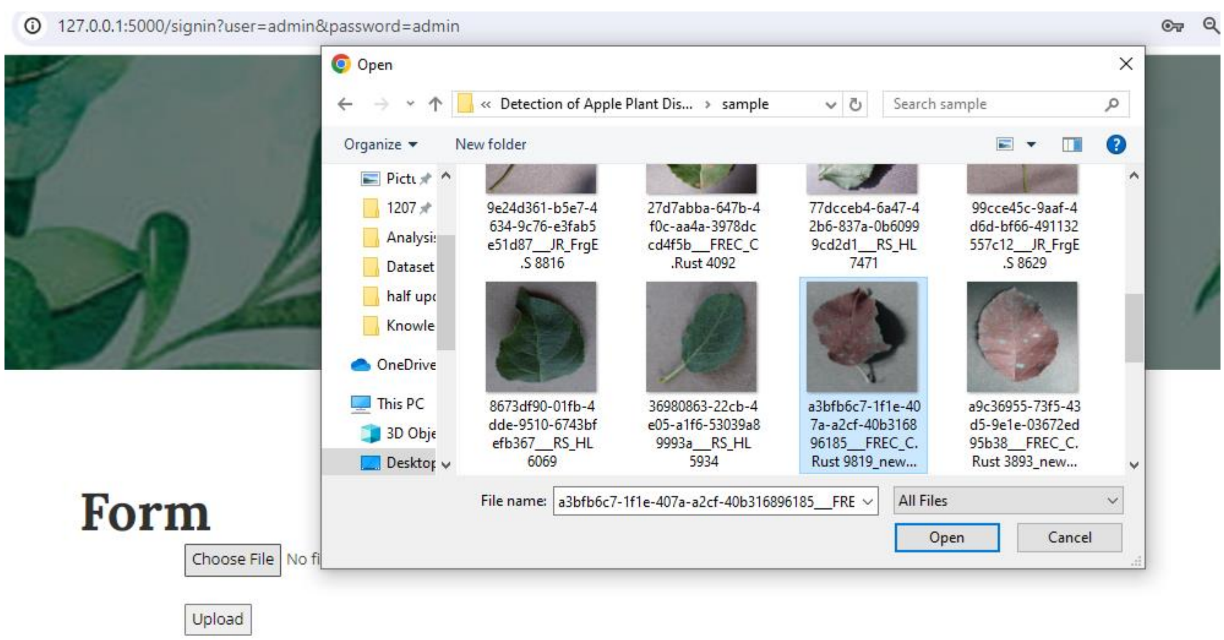


Fig 3.3.2.8 Upload Image

Code:

```
import os
import cv2
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import mahotas
import pickle
# Set seed for reproducibility
seed = 9
# Models dictionary
models = {
    "Logistic Regression": "lr",
    "K Neighbors Classifier": "knn",
    "Decision Tree Classifier": "dtc",
    "Random Forest Classifier": "rf",
    "Linear Discriminant Analysis": "lda",
    "Gaussian Naive Bayes": "nb",
    "SVC": "svm",
}
# Model labels
labels_mapping = {
    0: "Apple___Apple_scab",
    1: "Apple___Black_rot",
    2: "Apple___Cedar_apple_rust",
    3: "Apple___healthy"
}

# Helper functions for feature extraction
def rgb_bgr(image):
return cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
```

```
def bgr_hsv(image):
return cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
def img_segmentation(rgb_img, hsv_img):
    lower_green = np.array([25, 0, 20])
    upper_green = np.array([100, 255, 255])
    healthy_mask = cv2.inRange(hsv_img, lower_green, upper_green)
    result = cv2.bitwise_and(rgb_img, rgb_img, mask=healthy_mask)
    lower_brown = np.array([10, 0, 10])
    upper_brown = np.array([30, 255, 255])
    disease_mask = cv2.inRange(hsv_img, lower_brown, upper_brown)
    disease_result = cv2.bitwise_and(rgb_img, rgb_img, mask=disease_mask)
    final_mask = healthy_mask + disease_mask
    final_result = cv2.bitwise_and(rgb_img, rgb_img, mask=final_mask)
    return final_result

def fd_hu_moments(image):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    feature = cv2.HuMoments(cv2.moments(image)).flatten()
    return feature

def fd_haralick(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    haralick = mahotas.features.haralick(gray).mean(axis=0)
    return haralick

def fd_histogram(image, mask=None):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    bins = 8
    hist = cv2.calcHist( [image], [0, 1, 2], mask, [bins, bins, bins], [0, 256, 0, 256, 0, 256] )
    cv2.normalize(hist, hist)
    return hist.flatten()
```

```
def main():
    st.set_page_config(
    page_title='Plant Disease Detection for Apple Leaves',
    page_icon="☘"
    )
    st.title("Plant Disease Detection for Apple Leaves")
    st.subheader("Upload an image and select a model to detect the disease")
    uploaded_file = st.file_uploader("Upload an image of an apple leaf", type=["png", "jpg",
    "jpeg"])
    if uploaded_file is not None:
        image = cv2.imdecode(np.frombuffer(uploaded_file.read(), np.uint8), 1)
        st.image(image, caption="Uploaded Image", use_column_width=True)
        model_name = st.selectbox("Select a model for disease detection",
        list(models.keys()))
if st.button("Detect"):
# Convert uploaded image to RGB and display
rgb_image = rgb_bgr(image)
st.subheader("RGB Image")
fig_rgb = plt.figure()
plt.imshow(rgb_image)
plt.axis('off')
st.pyplot(fig_rgb)
# Convert RGB image to HSL
# Convert  RGB image to display
```

```
hsl_image = bgr_hsv(rgb_image)
st.subheader("HSL Image")
fig_hsl = plt.figure()
plt.imshow(hsl_image)
plt.axis('off')
st.pyplot(fig_hsl)

# Perform image segmentation
segmented_image = img_segmentation(rgb_image, hsl_image)


# Display segmented images
st.subheader("Segmented Images")
fig_segmented = plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.imshow(segmented_image)
plt.axis('off')
plt.title("Segmented Image")
plt.subplot(1, 3, 2)
lower_green = np.array([25, 0, 20])
upper_green = np.array([100, 255, 255])
mask = cv2.inRange(hsl_image, lower_green, upper_green)
result = cv2.bitwise_and(rgb_image, rgb_image, mask=mask)
plt.imshow(mask, cmap="gray")
plt.axis('off')
plt.title("Healthy Mask")
plt.subplot(1, 3, 3)
lower_brown = np.array([10, 0, 10])
upper_brown = np.array([30, 255, 255])
disease_mask = cv2.inRange(hsl_image, lower_brown, upper_brown)

disease_result = cv2.bitwise_and(rgb_image, rgb_image, mask=disease_mask)
```

```
plt.imshow(disease_mask)
plt.axis('off')
plt.title("Disease Mask")
st.pyplot(fig_segmented)
# Calculate feature descriptors
hu_moments = fd_hu_moments(segmented_image)
haralick_texture = fd_haralick(segmented_image)
color_histogram = fd_histogram(segmented_image)

# Concatenate global features
global_features = np.hstack([color_histogram, haralick_texture, hu_moments])

# Load the selected model
model = models[model_name]
loaded_model = pickle.load(open("models/{}.pkl".format(model), "rb"))
# Predict the disease label
prediction = loaded_model.predict([global_features])
st.subheader("Prediction")
st.write("Prediction: ", prediction)
predicted_label = labels_mapping[prediction[0]]
# accuracy = loaded_model.score([global_features], [prediction])
st.subheader("Disease Prediction")
st.write("Predicted Label-: ", predicted_label)
if __name__ == '__main__':
    main()
```

3.3.3 System Requirements:

3.3.3.1 Hardware Requirements:

- Operating System : Windows Only
- Processor : i5 and above
- RAM : 4GB and above
- Speed : 2.30GHz
- Hard Disk : 50 GB

3.3.3.2 Software Requirements:

- Visual Studio Community Version
- Nodejs (Version 12.3.1)
- Python IDEL (Python 3.7)

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

The convolutional neural network (CNN) model proposed in this study demonstrates promising outcomes in the detection of plant diseases, particularly in the context of apple crops. By leveraging augmentation techniques such as shift, shear, scaling, zoom, and flipping, the model effectively expands the training dataset, thereby enhancing its ability to identify diseases using leaf images. Through experimentation with the publicly available PlantVillage dataset, the model showcased commendable classification accuracy, effectively discerning between Scab, Black rot, and Cedar rust diseases in apple leaves. The experimental results highlight the efficacy of the proposed CNN model, as it not only achieves good classification accuracy but also addresses key challenges associated with deep learning techniques in plant disease detection.

Notably, the model's utilization of augmentation techniques enables the generation of additional training samples without the need for capturing more images, thus mitigating the requirement for larger training sets. This approach not only enhances the model's performance but also alleviates issues related to overfitting, thereby contributing to its robustness in disease identification. Furthermore, the proposed CNN model offers practical advantages in terms of storage and computational resources. Compared to several existing deep CNN models, the proposed model demonstrates reduced resource requirements, making it more feasible for deployment in handheld devices. This aspect is particularly significant for real-world applications where accessibility and ease of use are paramount. By optimizing storage and computational efficiency, the proposed model presents a viable solution for on-the-go disease detection, empowering farmers and agricultural stakeholders with a convenient tool for monitoring crop health.

Overall, the experimental findings underscore the potential of the proposed CNN model as a reliable and efficient tool for plant disease detection. Its ability to achieve high classification accuracy, coupled with its resource-efficient nature, positions it as a promising solution for addressing the challenges associated with crop losses due to plant diseases. As advancements in deep learning continue to evolve, further refinements and optimizations of such models hold the potential to revolutionize agricultural practices, contributing to improved crop yield and food security on a global scale.The rigorous investigation manifests the proposed model to be better than various pre-trained CNN models. The method was also found better than some other existing methods on the basis of various parameters including accuracy and memory requirements.

4.1 Output Screens:

Fig 4.1 Apple leaf is diagnosed as Rust

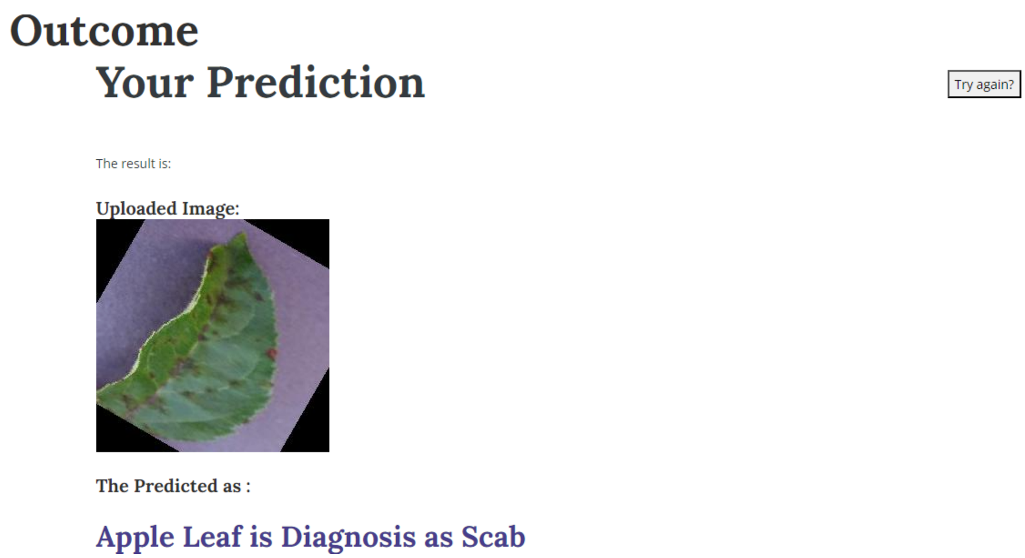


Fig 4.2 Apple leaf is diagnosed as Scab

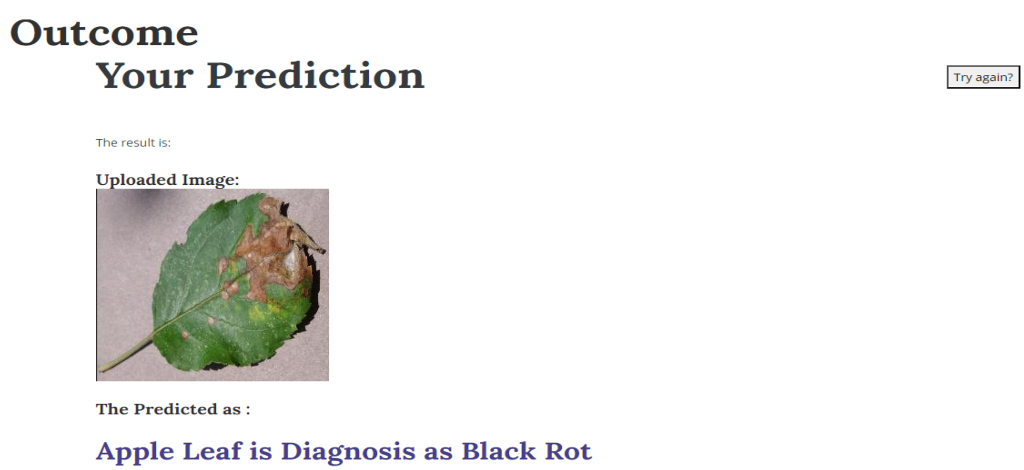


Fig 4.3 Apple leaf is diagnosed as Black Rot

Fig 4.4 Apple leaf is diagnosed as Healthy.

4.2 Performance Evaluation

SNO	ML Model	Accuracy	Precision	Recall	F1Score
0	VGG19	0.256	0.000	0.000	0.000
1	ResNet152	0.978	0.978	0.977	0.977
2	DenseNet201	0.981	0.981	0.981	0.981
3	MobileNetV2	0.962	0.962	0.962	0.962
4	ResNet50	0.939	0.939	0.939	0.939
5	VGG16	0.259	0.000	0.000	0.000
6	InceptionV3	0.898	0.900	0.897	0.898
7	Xecption	0.954	0.954	0.954	0.954
8	CNN	0.892	0.334	0.208	0.248
9	MobileNet	0.224	0.224	0.224	0.224

Table 4.2.1 Performance Evaluation

From the above table, we can observe that the pre-trained models gives the best accuracy results like MobileNetV2 as 96%, ResNet152 as 97% and DenseNet201 as 98%. Among these three DenseNet201 model has achieved high accuracy than other models and it is best choice for the prediction of apple leaf diseases.

4.3 Comparison of Graphs

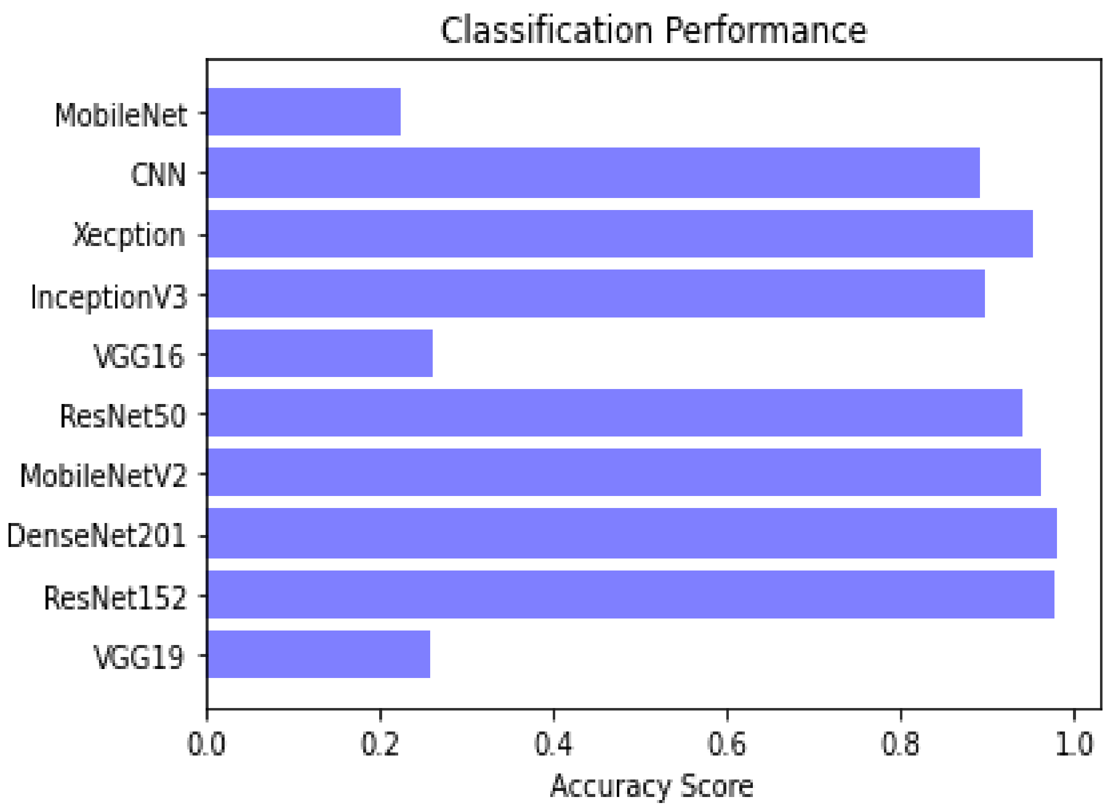


Fig 4.3.1 Accuracy Score of each model

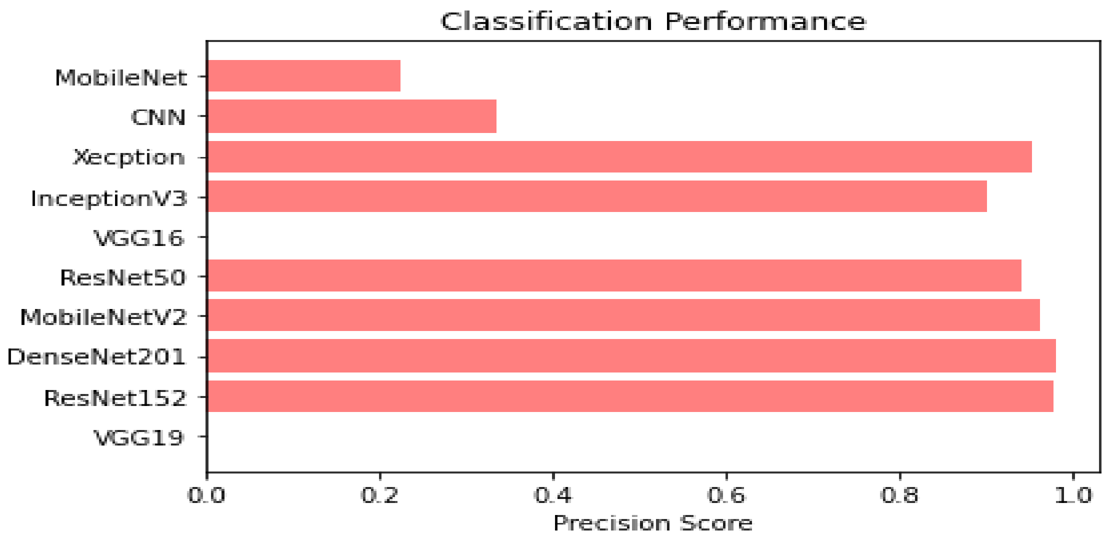


Fig 4.3.2 Precision Score of each model

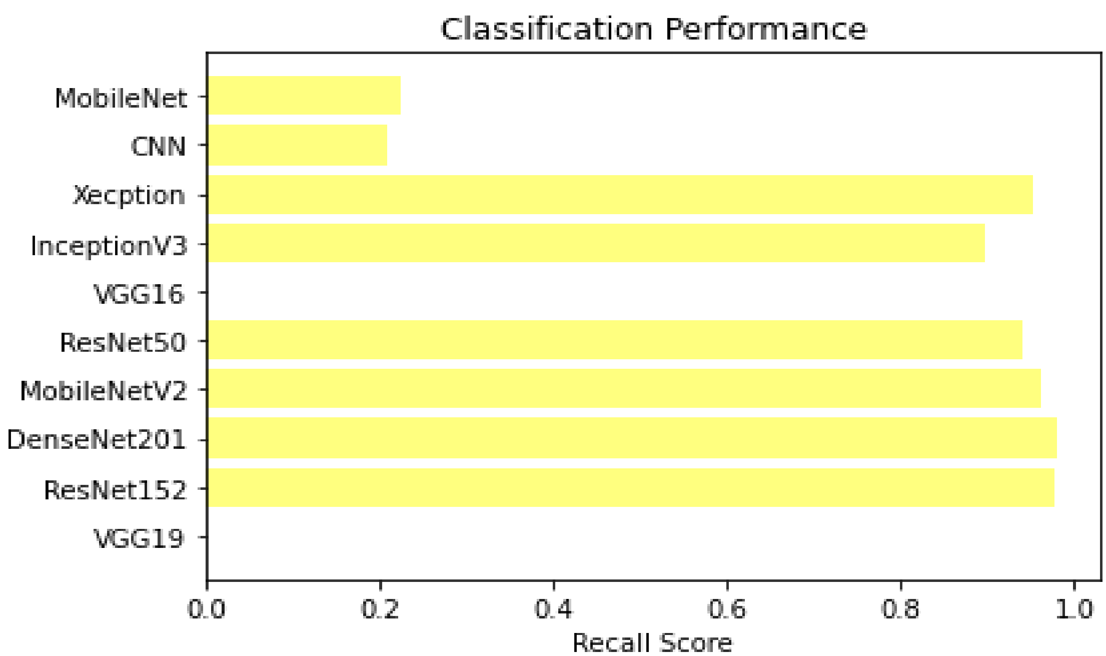


Fig 4.3.3 Recall Score of each model

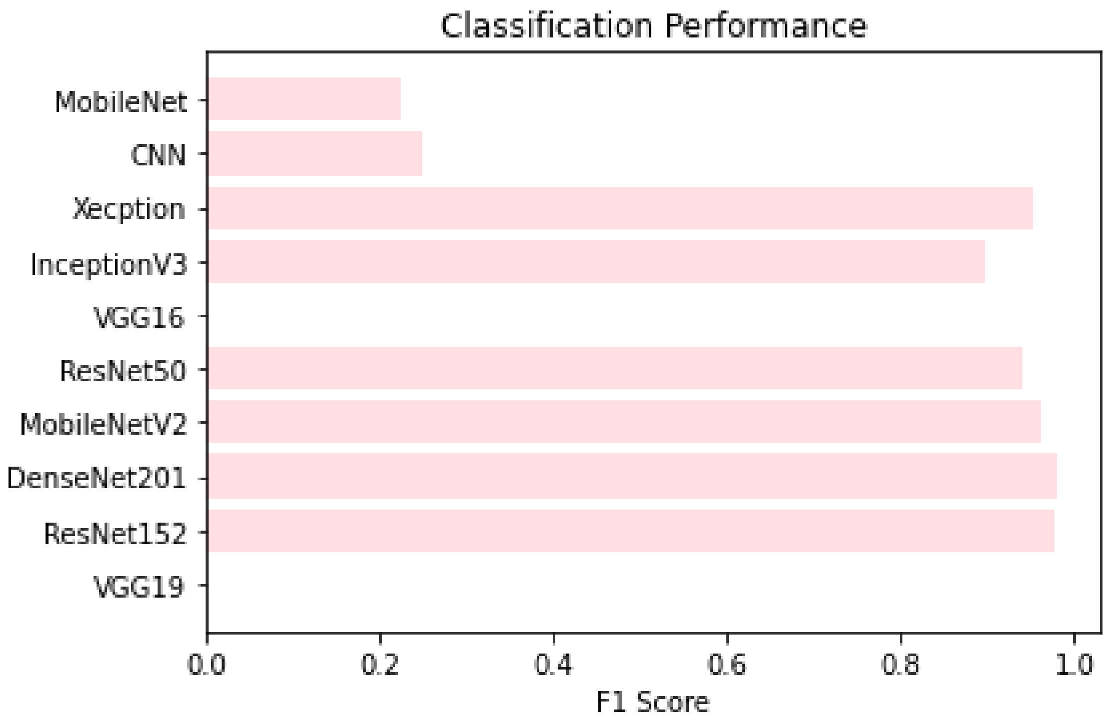


Fig 4.3.4 Recall Score of each model

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION

The conclusion of the research paper states that the proposed deep CNN model can assist non-expert farmers in apple orchards and lower the stress on plant pathologists by identifying diseases in apple crops with the help of leaf images. The model was trained on 3171 apple leaves for 1000 epochs and achieved good accuracy on the PlantVillage dataset. The proposed model was found to be better than various pre-trained CNN models in terms of storage and computational resources. The proposed model is highly suitable for deploying in handheld devices.The possible extension of this work include collection of more leaf images of apple plants from various geographically different areas with varying image quality at different plant growth stages in different cultivation conditions.

5.1 Future Scope:

In future research, expanding the dataset to include more diverse apple leaf images from various geographic locations and cultivation conditions will enhance the model's robustness and generalization capabilities. Additionally, capturing images at different growth stages and under varying environmental conditions will further enrich the dataset, enabling the model to effectively recognize disease symptoms across a broader spectrum of scenarios. Incorporating advanced techniques such as transfer learning and data augmentation could also improve model performance and efficiency. Furthermore, exploring the integration of real-time disease detection systems with precision agriculture technologies can revolutionize orchard management practices, providing timely interventions and optimizing resource allocation. Collaborating with agricultural experts and stakeholders to validate the model's effectiveness in real-world settings and integrating user-friendly interfaces for seamless deployment on handheld devices will be crucial steps towards facilitating widespread adoption and practical implementation in apple orchards worldwide.

CHAPTER 6
REFERENCES

REFERENCES

[1]. J. Arun Pandian,V. Dhilip Kumar et.al., “Plant Disease Detection Using Deep Convolutional Neural Network” published in MDPI, Link, available in https://www.mdpi.com/20763417/12/14/6982

[2]. Muzafar Rasool Bhat, ‘‘Apple diseases: identification & classification using transfer learning’ 2022.[Online]. Available:https://qascf.com/index.php/qas/article/view/1167/1289.

[3]. Chongke Bil, “MobileNet Based Apple Leaf Disease Identification” 2020, Available:https://doi.org/10.1007/s11036-020-01640-1

[4]. T. Deshpande, ‘‘State of agriculture in India,’’ PRS Legislative Res., no. 113, pp. 1–29, 2017.[Online]. Available: https://www.prsindia.org/ policy/discussion-papers/state-agriculture-india

[5]. The State of Food Security and Nutrition in the World 2020. Transforming Food Systems for Affordable Healthy Diets, FAO and UNICEF, FAO and IFAD and UNICEF and WFP and WHO, Rome, Italy, 2020, Accessed: 25-June-2021. [Online]. Available: http://www.fao.org/documents/card/en/c/ca9692en/

[6]. World Production, Markets, and Trade Report–Fresh Apples, Grapes, and Pears: World Markets and Trade, United States Dept. Agricult. Foreign Agricult. Service, Foreign Agricultural Service/USDA, Washington, DC, USA, Accessed: Jul. 12, 2021. [Online]. Available:https://www.fas.usda.gov/data/fresh-apples-grapes-and-pears-world-markets-and-trade

[7]. National Horticulture Board. Report on Apple production in India. Accessed: Dec. 2020. [Online]. Available: http://nhb.gov.in/report_ files/apple/APPLE.html

[8]. V. K. Vishnoi, K. Kumar, and B. Kumar, ‘‘Plant disease detection using computational intelligence and image processing,’’ J. Plant Diseases Protection, vol. 128, no. 1, pp. 19– 53, Feb. 2021, doi: 10.1007/s41348-020- 00368-0.

[9]. S. Kaur, S. Pandey, and S. Goel, ‘‘Plants disease identification and classification through leaf images: A survey,’’ Arch. Comput. Methods Eng., vol. 26, no. 2, pp. 507– 530, Apr. 2018, doi: 10.1007/s11831-018-9255-6.

GitHub Link
https://github.com/Ravi8295

DOI LINK: https://doi.org/10.22214/ijraset.2024.59800

RESEARCH PAPER

AND

CERTIFICATION

iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59800

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Prediction of Disease in Apple Leaf Using Convolution Neural Networks

B. Ravichandra[1], P. Shirisha[2], U. Harshith[3], Dr. S. Kirubakaran[4]
[1, 2, 3]UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana
[4]Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

***Abstract:** Apple tree diseases can significantly impact fruit quality and yield, making early detection and intervention crucial for orchard management. To avoid the impact on apple production a novel approach is required for the early and accurate prediction of apple leaf diseases through Convolutional Neural Networks (CNNs). Our proposed system contains a comprehensive dataset of high-resolution pictures of apple leaves exhibiting various disease symptoms, including common issues like apple scab, apple rust, and powdery mildew. The dataset was carefully annotated to train and validate the CNN model effectively. The proposed CNN model makes use of its ability to automatically learn applicable features from images, making it highly suitable for the task of disease prediction in apple leaves. This model contributes to the field of precision agriculture by offering a cost-effective and efficient tool for apple disease detection.*

***Keywords:** Apple diseases, convolutional neural network, classification, deep learning, disease identification, image processing.*

I. INTRODUCTION

In the last few years, the agricultural sector has witnessed a surge in the utilization of advanced technologies, particularly in disease identification and management. Among various crops, apple cultivation holds significant economic importance worldwide. However, one of the most difficult tasks for apple growers is the early detection and control of diseases that can have a noteworthy effect on yield and quality. Apple is among the globally consumed fruits among the four after bananas, grapes, and oranges. Globally, using pests and fertilizers can affect the apple plant and also the standard of fruit which should show an influence on the overall production of the apple crop In India, fungal diseases are the major ones that affect the apple fruit quality in particular apple producer state i.e. Himachal Pradesh, which is the second-biggest apple producer in India.

Infections in plants include diseases such as biotic and abiotic. Biotic diseases like scab, cedar rust, leafy blotch, powdery mildew, blight, mosaic, black rot, etc are very dangerous compared to abiotic diseases. It is crucial to increase apple production and early identification of disease. There are some serological method-based techniques for disease identification such as Fluorescence in-situ hybridization (FISH), polymer chain reaction (PCR), immunofluorescence (IF), and flow cytometry used by experts and can only be performed in a laboratory.

Machine learning-based approaches such as K-nearest Neighbor (KNN), Artificial Neural Networks (ANN), and Support Vector Machine (SVM) are mostly employed for disease identification. Convolution Neural Networks (CNN), one type of deep learning technique, have efficiency and it has also proved their suitability for plant disease identification. A deep CNN model consisting of three convolution layers is developed for disease detection based on apple leaf images. This model can self-learn hierarchical features, making it ideal for tasks such as image categorization and object detection. The objective is to create a robust and accurate a model capable of identifying common diseases affecting apple trees such as scab, rust (or) black rot.

II. LITERARATURE REVIEW

A. Transfer Learning Approach

[1] In this research study, a transfer learning strategy was employed to identify apple leaf disease using the CNN model. Transfer learning is a method of using a pre-trained model as a starting point for a new task. This study shows that a transfer learning strategy can accomplish high detection of apple leaf disease using a relatively small dataset (consisting of 2897 images) of training data.

First, you need to choose a pre-trained CNN model trained on a large dataset such as image-Net. At the top of the trained network, remove the fully linked layer. Freeze the convolutional layer weights if necessary. This prevents updates during training and preserves previously learned features. Add a new fully connected layer above the convolutional base. Train the model using the collected dataset. Experimentation results on a gathered dataset of 2897 images with data augmentation demonstrated that AppleNet can be effectively used to detect apple diseases on apple leaves with a classification accuracy of 96.00%.

B. Mobilenet CNN Model

[2] The study proposed a MobileNet CNN model for identifying disease in apple leaves.MobileNet is a lightweight CNN architecture that is designed for mobile devices. This study found that MobileNet can achieve high accuracy in apple leaf disease identification using a small model size and low computational cost.

MobileNet has several variants, such as MobilenetV1, and MobileNetV3. Each has improved performance, efficiency, and capabilities. We practiced and examined the MobileNet model using a dataset containing 334 images. Finally, the highest accuracy achieved by this model is 74.

C. Lightweight and Efficient CNN Model

[3] In this study, we suggested a lightweight and efficient CNN model for the identification of apple leaf disease. The model was trained using a collected dataset of apple leaf pictures labeled with different disease types. This study found that the model succeeded in achieving 95.5% accuracy in identifying the apple leaf disease while maintaining a small model size and low computational requirements. It provides a more systematic and practical solution for identifying illnesses of the apple leaf on mobile devices.

To make a lightweight and efficient CNN model, Use fewer layers and fewer filters in each layer. This reduces the complexity of the model. Instead of big filters, use smaller ones. Instead of always reducing the size of the image, find ways to do it only when necessary. This saves computing power. And pick a model that's designed to be efficient, like MobileNet or SqueezeNet. They are built to be fast and use less memory.

D. Comparison of Plant Leaf Classification Using Modified AlexNet and Support Vector Machine

[4 Farmers find it challenging to detect leaf diseases early. As technology advances, many models for detecting diseases have been introduced that are superior to older methods such as apple leaf collection and laboratory process validation. This article uses a pre-trained Model Alexnet using Transfer Learning to classify leaves. A dataset is collected which consists of 3200 images with the measurements of 226 x 226 x 3. Pre-process the collected data and perform the training and evaluation with different variations. Then, modify and apply deep learning techniques like Alexnet which can classify 1000 classes and calculate the performance parameters. The classification can be done based on accuracy and confusion matrix. For this model, the obtained accuracy is 91.15%.

III. MATERIALS AND METHOD

A. Dataset and Pre-processing

The proposed deep Convolution Neural Network model is to predict diseases based on apple leaves, such as apple scab, black rot, cedar rust, and unaffected leaves. A total of 3171 images are collected including healthy and unhealthy leaves of black rot, rust, and powdery apple diseases. A total of 4 labels are created.

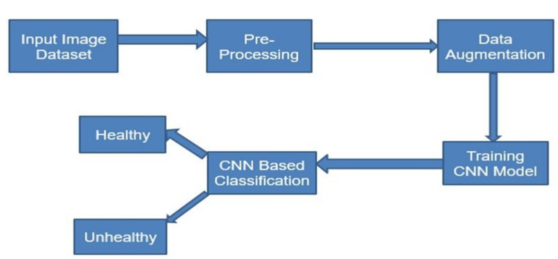


Fig 1: Block Diagram

The apple leaf images dataset is taken from the Kaggle website which dataset is composed of 3171 images of apple plant leaves.

This model contains four classes of which the three classes correlate with 3 apple diseases namely black rot, scab, and apple cedar rust while the other class contains healthy apple leaf images. The apple leaf images of size 256 × 256 of all four categories were captured with a simple background at various plant development stages.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

It is crucially important to have a wide variety of leaf photos within the collection. Hence, the model can learn important variations throughout the instruction phase. It helpfully assists in enhancing the generalizability of the CNN model. Augmentation presents itself as an approach to artificially create various variations of the image. In this work, amazing transformations like panning, shearing, scaling, zooming, and flipping are used to transform images. These transformations add minor variations in images, which inherently help in introducing variety in the training set. It turns out to assist in reducing overfitting and helps the model achieve better tolerance and an enhanced ability to generalize.

B. Convolution Neural Network

CNN is a feed-forward ANN-based deep learning methodology and is an attractive method. CNN is employed to display more layers within the CNN. Here, we construct a deep CNN by stacking several constructing elements such as convolutional layers, pooling layers, and fully connected levels with typical nonlinear activation units. It has exhibited certain advantages over state-of-the-art ML-based methods like magic! It doesn't require any additional effort for feature engineering. It is successfully used in several applications including image/text classification, NLP, precision agriculture, etc.

1) Convolution Layer

Convolutional layers are in charge of performing the convolution of a filter (kernel) on an input image. Feature maps are produced via convolutional layers by finding local connections that appeared in previous layers. Fundamentally, the first layer (convolutional layer) comprises two components: a linear convolution operation and a nonlinear activation unit. Convolution operations are performed on image volumes that have multiple channels, such as RGB images. Convolutional layers are a crucial component of the picture-processing process and play a key role in identifying patterns and features in visual data. By applying filters to input images, the layer can extract key information and generate feature maps that aid in subsequent analysis tasks. The process of convolution is expressed as in (1):

$$Conv(I,K)_{x,y} = \sum_{i=1}^{nH} \sum_{j=1}^{nW} \sum_{k=1}^{nC} Ki,j,kI_{x+i-1\,,y+j-1\,,k} \qquad (1)$$

Where the kernel K ($f_{□}$, f_w, n_C) convolves with the picture I (n_H, n_W, n_C) of different sizes but of similar no. of channels n_C and generate a feature map O (o_H, o_W, z). The $f_{□}$ represents the height and breadth (width) of the kernel. And, n_H, n_W denote the height and breadth(width) of the specified image. Conventionally, the kernel is considered as an odd-dimensional square window, i.e., $f_H = f_W = f$. The generated feature map dimension is defined as in (2):

$$Feature\_map\ (O_H, O_W, z) = \left(\left\lfloor\frac{n_H + 2p - f}{s}\right\rfloor + 1,\ \left\lfloor\frac{n_W + 2p - f}{s}\right\rfloor + 1\ ,\ z\right) \qquad (2)$$

where symbol p indicates the padding value, s indicates the stride, and z is the no. of kernels convolved with the input image.

Rectified linear unit (ReLU) is the most widely used activation function. Rectified linear unit do not activate all neurons simultaneously. When the yield or other direct change of the convolution unit rises to or is more prominent than zero, the neurons are essentially activated. It is expressed as in (3):

$$f(z) = \max(0, z) \qquad (3)$$

2) Pooling layer

A pooling layer is accustomed to downsampling the feature map produced by convolution. It can lower the dimensions of activation maps that contain a larger number of parameters. Hence, it helps in lowering the computational burden, controls the process-related overfitting, and ultimately reduces the time required for training. The major pooling operations include max, min, and average. However, the most popular method of pooling resources is max-pooling, which selects the highest value from each input patch. The max pooling procedure is given in (4):

$$Max_{Pooling}: y_j = \max_{i \in R_j}(P_i) \qquad (4)$$

where R indicates a receptive field containing P pixels. The dimension of the feature map that was produced is defined as in (5):

$$Feature\_map\ (O_H, O_W, n_C) = \left(\left\lfloor\frac{n_H + 2p - f}{s}\right\rfloor + 1,\ \left\lfloor\frac{n_W + 2p - f}{s}\right\rfloor + 1,\ n_C\right) \qquad (5)$$

The pooling operation only modifies the dimensions nH and nW whereas nC remains unchanged.

3) *Fully Connected Layer*

The fully connected (FC) or dense layers of a CNN are almost identical to the layers of a traditional neural network. They are typically connected at the final stage of CNN to form an output layer with the desired number of outputs. The FC layers operate on 1-D data. The flattened layer arranges the 2-D output of the preceding layers in a 1-D representation. The FC layers conduct two types of functioning: linear and, nonlinear transformations. These transformations can be expressed as in (6):

$$Z = W^T.X + b$$
$$O = f(z) \tag{6}$$

where X represents the input feature map, W denotes weight, b denotes bias terms, and O represents the output of the fully connected layer. For better prediction, optimal weights are eventually needed to lower the loss function. The Gradient descent is the most frequently utilized technique for determining optimization weights. Adam algorithm is another technique to get a less noisy and smoother path while optimizing the gradients. It performs learning rate annealing based on finding the adaptive estimates of the lower-order moments.

C. *Proposed Method*

The proposed deep CNN (Conv-3 DCNN) model consists of three collaborative layers and 2 fully connected layers after the three max-pooling units. ReLU is explored as a nonlinear activation function at each convolution zone and the first dense layer. The function of Softmax is employed at the layer of output to classify apple plant diseases. The softmax function is responsibly responsible for multiclass classification and assumes that each sample belongs to exactly one class.

The developed deep CNN model uses different layers along with activation functions. A dropout layer is as additionally employed as well at the third max pool layer to cause overfitting. The dropout unit eliminates some randomly selected neurons. The network could not rely on any one feature. Therefore, some neurons are ignored to spread out the weights for better generalization.

Initially, at the first convolution level, 32 filters (3 × 3) with valid padding and stride (1, 1) are selected to convolute over RGB images of size 256 × 256. It produces 32 feature maps of size 254 × 254. In the resulting feature map, the no. of channels corresponds to the no. of filters applied. At the first pooling level, the foregoing produced feature maps are downsampled by a kernel of size 2 × 2, and 32 feature maps of size 127 × 127 are generated. The same kernels are used at respective higher layers. The proposed deep CNN model will show promising results in the classification of apple plant diseases.

IV. RESULTS AND DISCUSSION

Fig 2: Home Page

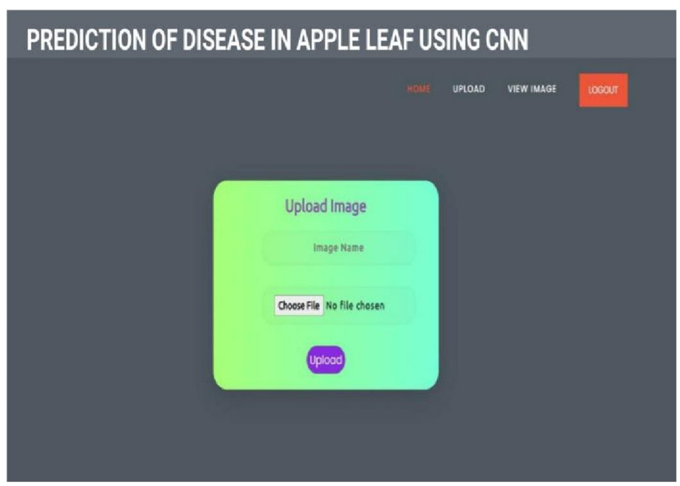


Fig 3: Select the input picture

Fig 2 shows the Home page which is the main web page of a website. It may also refer to the start page which will shown in a web browser when the application first opens.

Fig 3 selects the Input picture from the dataset. Drag and drop to upload the dataset we click the button named Upload. The dataset is uploaded successfully**.**

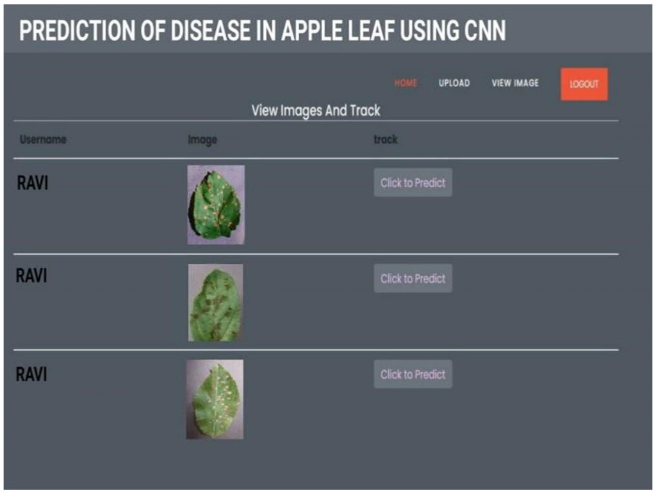


Fig 4: Predict Disease

Fig 5: Predicted Result

In Fig 4 dataset is uploaded successfully, we view images and track then click to predict. It verifies the image and predicts the diseases.

Fig 5 shows the predicted disease. It will display the result of which disease is predicted.

V. CONCLUSION

In this work, a CNN model was developed to recognize diseases in apple crops based on apple leaf images. It can assist non-expert farmers in apple orchards and lower the pressure on plant pathologists. Over 1000 epochs were used to train the model on 3171 apple leaves. On the PlantVillage dataset, the model's accuracy is evaluated to 98%. The rigorous investigation manifests the proposed model as much better than various pre-trained CNN models. The method was also found superior to other existing models based on various parameters including accuracy and memory requirements. For several diseases, this model achieves good accuracy, ranging from 97% to 99%. The model will successfully balance the accuracy and precision. The AUC-ROC curve shows that the proposed approach is reliable and consistent.

REFERENCES

[1] Muzafar Rasool Bhat, ‘‘Apple diseases: identification & classification using transfer learning’ 2022.[Online]. Available: https://qascf.com/index.php/qas/article/view/1167/1289.

[2] Chongke Bil, “MobileNet Based Apple Leaf Disease Identification” 2020, Available:https://doi.org/10.1007/s11036-020-01640-1

[3] Lili Li;Shujuan Zhang ; Bin Wang; “ Apple Leaf Disease Identification using Lightweight Convolution Networks” 2021, Available:https://doi.org/10.3390/s22010173

[4] Shivali Amit Wagle, Harikrishnan R, “Comparison of Plant Leaf Classification Using Modified AlexNet ” 2021, Available: https://doi.org/10.18280/ts.380108

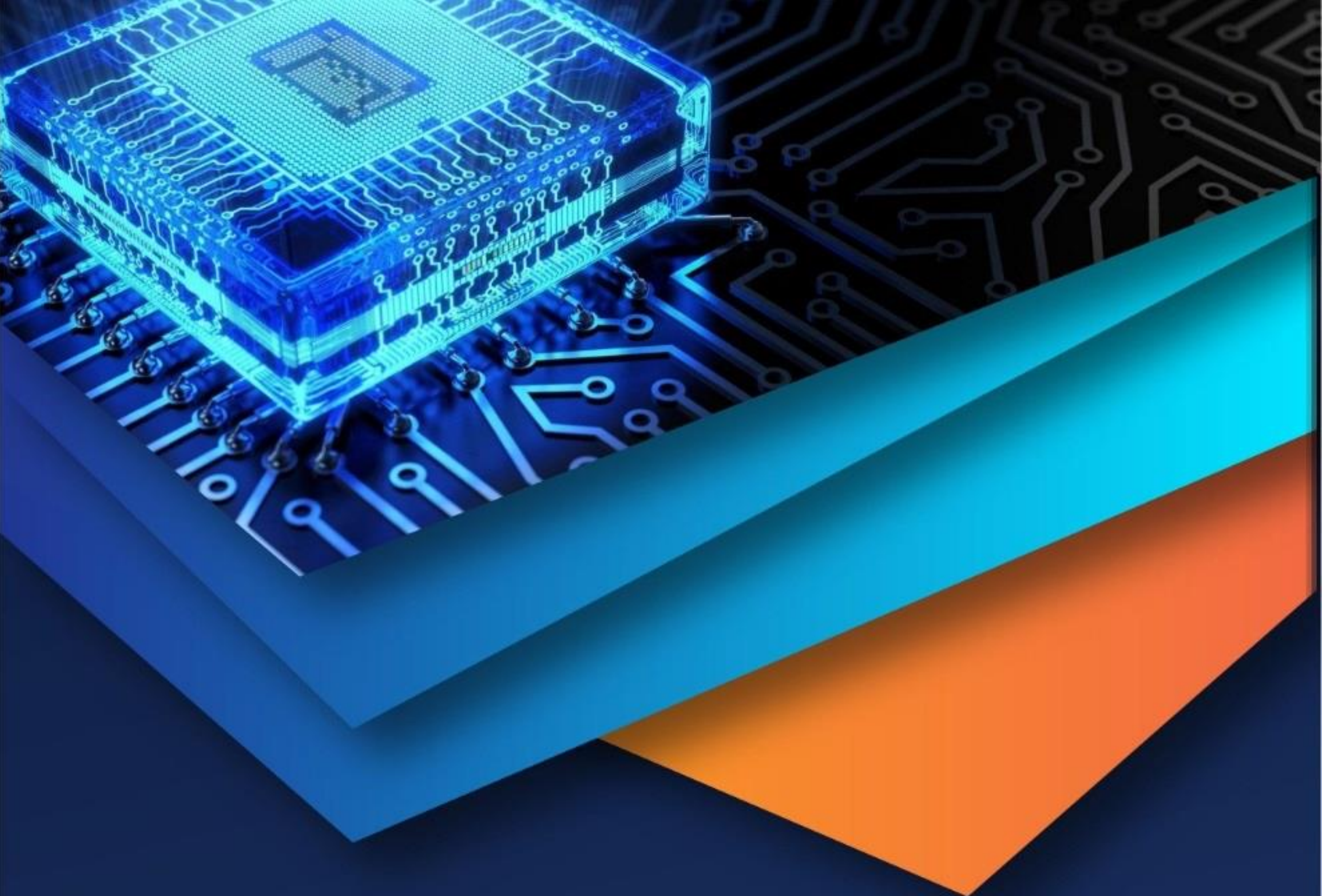
[5] T. Deshpande, ‘‘State of agriculture in India,’’ PRS Legislative Res., no. 113, pp. 1–29, 2017. [Online]. Available: https://www.prsindia.org/ policy/discussion-papers/state-agriculture-India

[6] National Horticulture Board. Report on Apple production in India. Accessed: Dec. 2020. [Online]. Available: https://nhb.gov.in/report_files/apple/APPLE.html.

[7] G. Huang, Z. Liu, L. van der Maaten, & K. Q. Weinberger, ‘‘Densely connected convolutional networks,’’ 2016, arXiv: 1608.06993.

[8] V. K. Vishnoi, K. Kumar, and B. Kumar, ‘‘Crop disease classification through image processing & machine learning methods using leaf images,’’ in Proc. 1st Int. Conf. Adv. Comput. Future Commun. Technol. (ICACFCT), Dec. 2021, pp. 27–32.

[9] A. N. I. Masazhar and M. M. Kamal, ‘‘Digital image processing technique for palm oil leaf disease identification through multiclass SVM classifier,’’ in Proc. IEEE 4thInt. Conf. Smart Instrum., Meas. Appl. (ICSIMA), Nov. 2017, pp. 1–6.



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59800, entitled

Prediction of Disease in Apple Leaf Using Convolution Neural Networks

by

B. Ravichandra

after review is found suitable and has been published in

Volume 12, Issue IV, April 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59800, entitled

Prediction of Disease in Apple Leaf Using Convolution Neural Networks

by

P. Shirisha

after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: **7.429**

45.98 INDEX COPERNICUS

THOMSON REUTERS
Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59800, entitled

Prediction of Disease in Apple Leaf Using Convolution Neural Networks

by

U. Harshith

after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS
Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59800, entitled

Prediction of Disease in Apple Leaf Using Convolution Neural Networks

by

Dr. S. Kirubakaran

after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET