A

Project Report

on

Foliar Disease Detection Using ML and Deep Learning

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2022-23 in

Computer Science and Engineering

By

Umang Pratap Singh(1900290100179)

Tushar Kundoo (1900290100174)

Saurabh Mandal(1900290100135)

Under the supervision of

Prof.Shalini Kapoor

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2023

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name:Umang Pratap Singh
Roll No:1900290100179
Date:
Signature:

Name:Tushar Kundoo
Roll No:1900290100174
Date:
Signature:

Name:Saurabh Mandal
Roll No:1900290100135
Date:
Signature:

CERTIFICATE

This is to certify that Project Report entitled "Foliar Diseases Detection using Machine Learning and Deep Learning" which is submitted by Umang Pratap Singh,Tushar Kundoo , Saurabh Mandal in partial fulfillment of the requirement for the award of degree B.Tech in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University,Lucknow is a reord of the candidates own work carried out by them under my supervision.The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date:

Supervisor Name:

Prof.Shalini Kapoor

(Designation)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to superisor name, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Name:Umang Pratap Singh
Roll No:1900290100179
Date:
Signature:

Name:Tushar Kundoo
Roll No:1900290100174
Date:
Signature:

Name:Saurabh Mandal
Roll No:1900290100135
Date:
Signature:

ABSTRACT

Foliar diseases are common ailments that affect the leaves of various plants, often caused by fungal or fungal-like organisms. These diseases can have a significant impact on plant health and productivity, making accurate diagnosis crucial for timely treatment and prevention of yield losses. Traditional methods of disease identification rely on labor-intensive field scouting, which can be inefficient and time-consuming. To address this, our project focuses on utilizing deep learning techniques, specifically convolutional neural networks (CNNs), to classify and identify tree diseases. By employing various deep learning and machine learning methods, we aim to determine the most effective approach for disease classification. The project specifically focuses on identifying the most prevalent diseases found in tomato, potato, and pepper plants, encompassing a total of 15 different illnesses. By allowing users to upload leaf images, our system provides real-time disease detection and offers the name of the identified disease, along with recommended pesticides for treatment. In cases where no disease is detected, a message indicating the absence of disease is displayed. Additionally, the system provides information on the proportion of the affected area and suggests suitable pesticides based on the extent of damage. As part of the research, we conducted a survey of various classification methods applicable to plant leaf diseases. The ultimate goal of this project is to contribute to the development of an efficient and user-friendly tool that enables early disease detection, mitigates crop losses, and promotes overall plant health.

In this project, we developed a system for the detection of foliar diseases in tomato, potato, and pepper plants using machine learning and image processing techniques. The system aims to accurately identify and classify 15 different types of diseases based on uploaded leaf images. By analyzing the color information and employing classification algorithms, the system can provide real-time disease diagnosis and recommend appropriate pesticides for treatment. The system also estimates the proportion of affected areas and provides insights for crop management. Through the implementation of deep learning models and extensive dataset analysis,we achieved high accuracy in disease detection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	**Full Forms**
ML	Machine Learning
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
KNN	K-nearest Neighbors
RF	Random Forest
RL	Reinforcement Learning
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
RNNLM	Recurrent Neural Network Language Model
CAE	Contractive Autoencoder

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Plant diseases cause yield reductions that have a direct influence on the domestic and international food production systems and lead to financial losses. About 20% to 40% of the world's food output is lost due to plant diseases and pests, according to the FAO of the United Nations has reported that 13% of global crop yield losses are due to plant diseases. This highlights the importance of identifying and preventing plant diseases to minimize these losses. One method for identifying plant diseases is by analyzing images of plant leaves, using a technique called "image processing" which falls under the field of signal processing. By leveraging the power of artificial intelligence, specifically machine learning, we can extract meaningful information from these images to accurately detect and diagnose plant diseases and thinking performs tasks itself or provides instructions on how to carry them out. Understanding the training data and incorporating it into models that should be helpful to humans is the basic goal of machine learning. Thus, it may help in making wise selections and forecasting the right output utilizing the vast training data. Leaf color, leaf damage level, leaf area, and leaf texture characteristics are utilized for classification. Several forms of plant diseases damage various plant organs. Plant pathologists can most easily identify foliar diseases, which are plant diseases that manifest symptoms on leaves. Fungal diseases are a major cause of yield losses, accounting for up to 50% of the total losses. As a result, many researchers are using computer vision, machine learning, and deep learning techniques to detect and diagnose plant diseases using images of plant leaves. Effective diagnosis of plant diseases involves early detection of diseases, identifying multiple diseases in different crops, estimating the severity of the disease, determining the appropriate amount of pesticide to apply, and taking practical measures to manage the disease and prevent its spread.

1.2 PROJECT DESCRIPTION

The primary objective of this project is to develop a system that can accurately identify common diseases in plant leaves, specifically focusing on tomato, potato, and pepper plants. The system will be capable of detecting and classifying 15 different types of diseases in these plants. By uploading a leaf image, the user will be able to receive real-time results, including the name of the detected disease and the recommended pesticides for treatment. In cases where no disease is found, a message stating "There is no disease on the plant" will be displayed. The system will also provide information on the proportion of the affected area and suggest appropriate pesticides based on the extent of the damage. The project includes a comprehensive survey of classification methods for plant leaf diseases, as addressing these diseases is crucial for achieving global food security. Modern farming techniques aim to feed the world's growing population, but plant diseases continue to pose a significant threat to crop yield and quality. Developing an image processing model for disease prediction and classification poses challenges, as visual recognition of disease symptoms can be difficult for farmers. Therefore, this project employs computerized image processing techniques that utilize color information from the leaves to facilitate the identification of unhealthy leaves and protect crops on a larger scale.

The project aims to develop an automated and accurate system for detecting and classifying plant diseases using machine learning and deep learning techniques. The primary objective is to minimize yield losses and optimize disease management strategies by enabling early detection and timely intervention. The project involves collecting a diverse dataset of plant leaf images, including both healthy and diseased samples, and implementing various deep learning models, such as convolutional neural networks (CNNs), for disease classification. We will explore transfer learning techniques to leverage pre-trained models and fine-tune them for our specific disease detection task. The project also includes rigorous evaluation and validation of the developed models using appropriate performance metrics. Additionally, we will address the interpretability aspect of the models by implementing visualization techniques, such as attention mechanisms, to provide insights into the decision-making process. Overall, this project aims to contribute to the field of foliar disease detection by developing accurate, efficient, and interpretable models that can be applied in practical agricultural settings.

CHAPTER 2

LITERATURE REVIEW

The literature on foliar disease detection using machine learning (ML) and deep learning techniques has witnessed significant advancements in recent years. Researchers have focused on developing accurate and automated methods for the early detection and diagnosis of plant diseases, aiming to minimize yield losses and optimize disease management strategies. Various approaches have been explored, encompassing traditional image processing techniques, statistical methods, and modern deep learning architectures for disease classification and detection.

One of the prominent advancements in the field is the application of convolutional neural networks (CNNs), a class of deep learning models that have demonstrated exceptional performance in image-based tasks. Previous studies have showcased the effectiveness of CNN architectures, such as AlexNet, VGG, and ResNet, in accurately identifying and classifying plant diseases based on leaf images. These models have the ability to learn discriminative features directly from raw image data, enabling them to capture intricate patterns and textures associated with different diseases. Transfer learning has also been widely employed in disease detection, allowing researchers to leverage pre-trained models on large-scale image datasets, such as ImageNet, and fine-tune them for specific disease detection tasks. This approach has shown promising results, especially when labeled training data is limited.

In addition to visual image analysis, researchers have explored the integration of multi-modal data sources to enhance disease detection capabilities. Hyperspectral imaging, for example, captures spectral information beyond the visible spectrum, enabling the detection of subtle biochemical changes associated with diseases. By combining hyperspectral imaging with ML algorithms, researchers have achieved improved accuracy in disease detection and classification. Spectroscopy, which measures the absorption, reflectance, or fluorescence of plant tissues, has also been employed to extract disease-related features and discriminate between healthy and infected leaves. The fusion of multi-modal data sources has the potential to enhance disease detection accuracy and robustness, allowing for more comprehensive analysis and interpretation of plant health.

While ML and deep learning techniques have shown great promise in foliar disease detection, several challenges and considerations remain. Dataset availability is a crucial aspect, as large and diverse datasets are required to train accurate and generalizable models. Collecting and annotating such datasets can be time-consuming and labor-intensive. Additionally, model generalization across different plant species and disease variations remains an important research area. Models trained on specific diseases or plant species may struggle to generalize to unseen or novel diseases, highlighting the need for more comprehensive and diverse training datasets.

Another challenge lies in the interpretability of deep learning models. Despite their impressive performance, deep learning models are often regarded as black boxes, making it difficult to understand the underlying reasoning behind their predictions. This limitation poses challenges in gaining the trust and acceptance of stakeholders in practical agricultural settings. Researchers are actively exploring methods for explaining and visualizing the decision-making process of deep learning models, including attention mechanisms, saliency maps, and gradient-based attribution methods.

Future research directions in foliar disease detection using ML and deep learning include the development of lightweight and real-time disease detection models suitable for deployment on edge devices, such as smartphones or low-power embedded systems. These models should balance accuracy and efficiency to enable on-site disease diagnosis and real-time monitoring. Additionally, exploring explainable AI techniques specific to plant disease detection can enhance the transparency and interpretability of the models, enabling farmers and agronomists to better understand and trust the automated diagnosis.

Collaboration with domain experts, such as plant pathologists and agricultural scientists, is crucial in ensuring the practicality and effectiveness of disease detection models. Field validation and integration of the developed models into existing plant disease management systems can provide valuable insights and contribute to the successful deployment of these technologies in real-world scenarios.

Transfer learning has emerged as a powerful technique in disease detection, especially in scenarios where labeled training data is limited. By leveraging pre-trained models on large-scale datasets, such as ImageNet, researchers can extract generic features from the early layers of the network and fine-tune the model on smaller disease-specific datasets. This approach allows for effective knowledge transfer and significantly improves the performance of disease detection models. Transfer learning has proved successful in various plant disease detection tasks, facilitating the development of accurate and efficient models even with limited annotated data.

In addition to accuracy, the interpretability of deep learning models is an area of active research in foliar disease detection. Deep learning models are often regarded as black boxes, making it challenging to understand the reasoning behind their predictions. This lack of interpretability can hinder the acceptance and trust of stakeholders, including farmers, agronomists, and plant pathologists, who require transparency and insights into the decision-making process.

Researchers have been exploring various methods to enhance the interpretability of deep learning models for disease detection. Attention mechanisms, for instance, provide insights into the regions of the image that contribute most to the model's prediction. By visualizing attention maps, users can understand which areas of the leaf are crucial for disease identification. This helps build trust in the model's decisions and allows domain experts to validate the relevance of identified regions.

CHAPTER 3

PROPOSED METHODOLOGY

In this section, we discuss pertinent initiatives in categorization problems utilizing deep learning architectures. Deep learning techniques have generally been the subject of much research for applications such as object recognition and image categorization. When used to solve recognition and classification issues, convolutional neural networks (CNNs), a deep learning technology, achieve state-of-the-art performance in picture classification.The first CNN architecture known as MobileNet for object recognition was evaluated using the dataset for tomato disease. To determine the degree of tomato leaf disease from photos of tomato leaves, pre-trained CNN architectures VGG16, MobileNet, and ResNet50 were implemented. Performance was improved by adding ResNet50 features to the traditional CNN model.

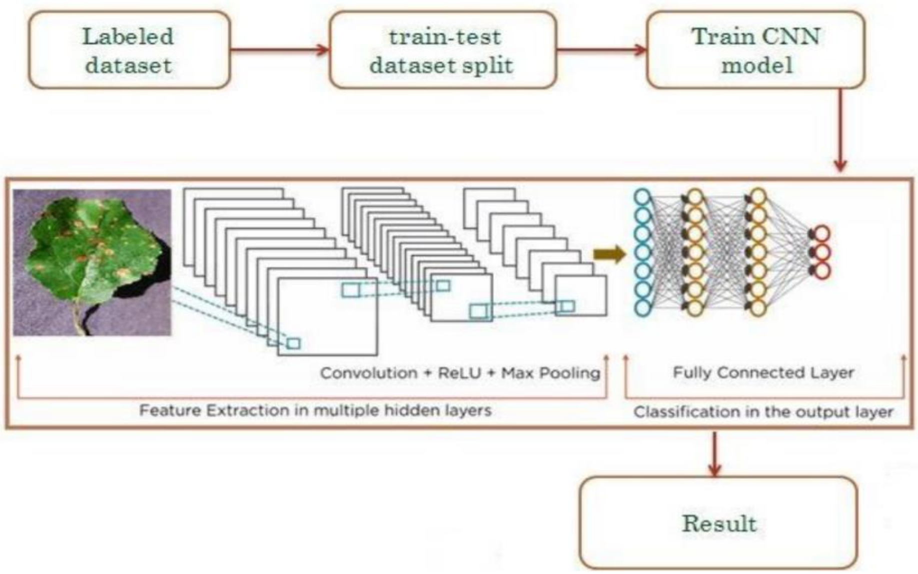


Fig 1: Proposed Workflow

The spatial links between the image's constituent parts are not taken into consideration by CNN architectures, making them ineffective for geometric transformations. By routing features from one layer to another in CNN, the max-pooling layer has a tendency to lose data. They are unable to model the rotational invariance of an item.The section presents a Capsule Network with Dynamic Routing algorithm to alleviate the shortcomings of CNN design. Capsule networks were used in the experiments to classify illnesses based on medical imaging, and they performed better than regular CNN in doing so.

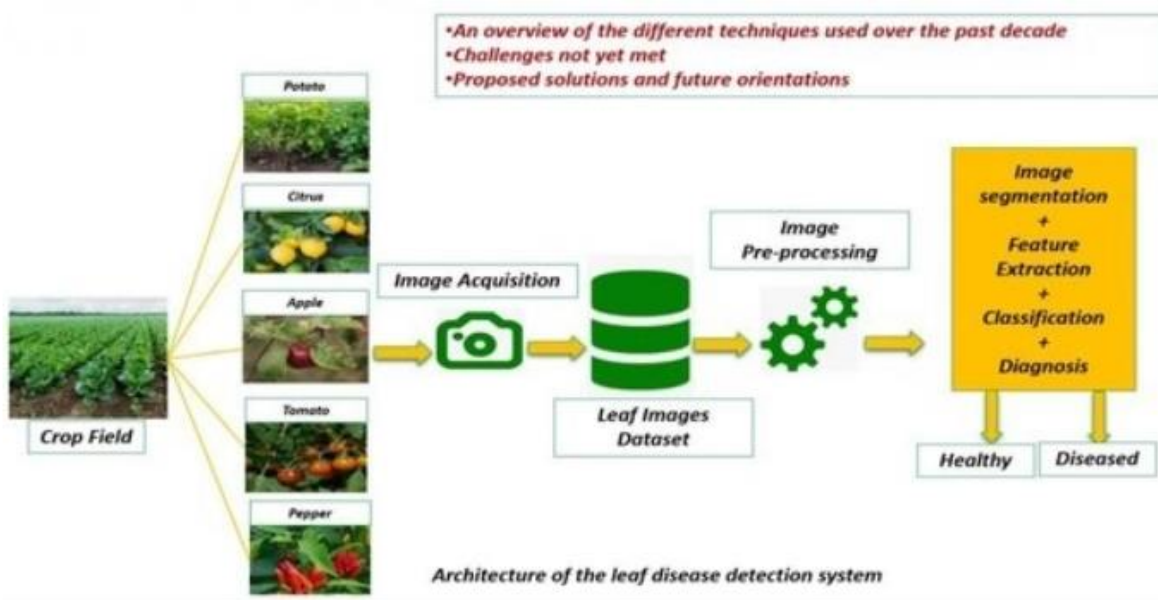


Fig 2: Architecture of the leaf disease detection system

- **Dataset Classification**
- **Building the CNN using transfer learning**
- **Factorizing Convolutions**
- **Testing**

3.1 DATASET CLASSIFICATION

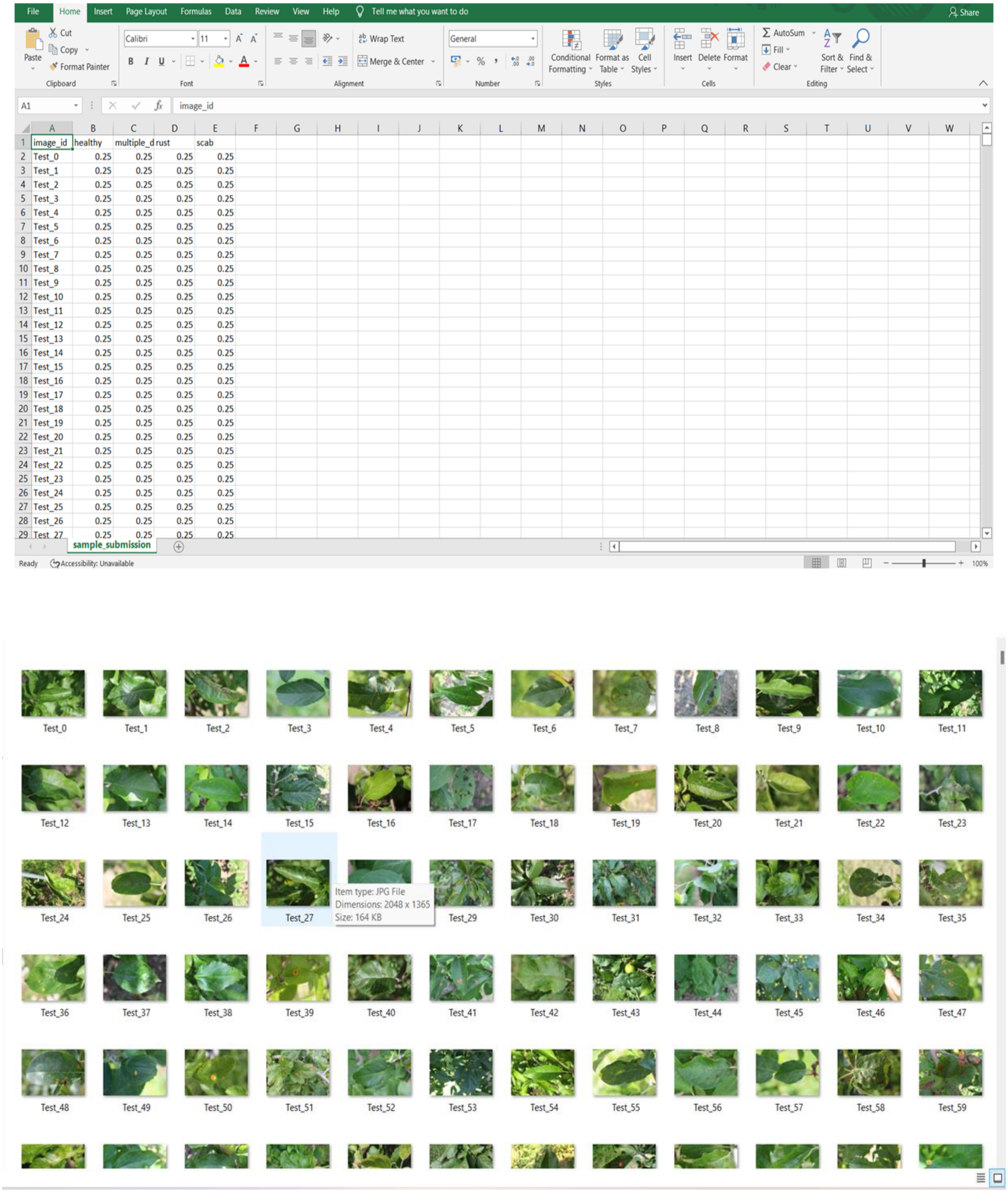
Selection of proper set of images for training of model is a significant task.Centroid of each image is calculated to retrieve select images.Centroid can be calculated by use of contours. Contour is a curve that joins all the points along the periphery of a shape. Contour scan much be detected much precisely on binary images. Hence, every image has to be converted to gray scale with a threshold applie do it.

Having found the contours, the image moments are calculated. Image moments are used to calculate the centre of mass or the centroid of an object.The function cv2.Moments return a dictionary of all moment values. From this moments one can extract data such as centroid , area, etc.As we only need centroid of the image, it is given by the relations,

Cx=(M[―m10‖]/M[―m00‖])and

Cy=(M["m01"] /M["m00"])

here,M is the dictionary of moments.



image_id	healthy	multiple_d	rust	scab
Test_0	0.25	0.25	0.25	0.25
Test_1	0.25	0.25	0.25	0.25
Test_2	0.25	0.25	0.25	0.25
Test_3	0.25	0.25	0.25	0.25
Test_4	0.25	0.25	0.25	0.25
Test_5	0.25	0.25	0.25	0.25
Test_6	0.25	0.25	0.25	0.25
Test_7	0.25	0.25	0.25	0.25
Test_8	0.25	0.25	0.25	0.25
Test_9	0.25	0.25	0.25	0.25
Test_10	0.25	0.25	0.25	0.25
Test_11	0.25	0.25	0.25	0.25
Test_12	0.25	0.25	0.25	0.25
Test_13	0.25	0.25	0.25	0.25
Test_14	0.25	0.25	0.25	0.25
Test_15	0.25	0.25	0.25	0.25
Test_16	0.25	0.25	0.25	0.25
Test_17	0.25	0.25	0.25	0.25
Test_18	0.25	0.25	0.25	0.25
Test_19	0.25	0.25	0.25	0.25
Test_20	0.25	0.25	0.25	0.25
Test_21	0.25	0.25	0.25	0.25
Test_22	0.25	0.25	0.25	0.25
Test_23	0.25	0.25	0.25	0.25
Test_24	0.25	0.25	0.25	0.25
Test_25	0.25	0.25	0.25	0.25
Test_26	0.25	0.25	0.25	0.25
Test_27	0.25	0.25	0.25	0.25

Fig 3: DataSet Examples

3.2 BUILDING THE CNN USING TRANSFER LEARNING

Image identification has become feasible with the advent of Convolutional Neural Networks.But designing a CNN that identifies objects and classifies them into distinct classes is a complextask. By making use of transfer learning it can be simplified. In transfer learning we have trained our model that has been trained on Plant Pathalogy dataset .Also Transfer learning significantly reduces training time and gives much better performance for relatively small dataset.

3.3 FACTORIZING CONVOLUTIONS

By means of factorizing convolutions the no. of connections and parameters are reduced to aconsiderable degree without adversly affecting the efficiency of the system. Factorization can be into smaller convolutions such as,two 3 by 3 convolutions replace one 5 by 5 convolution; or as symetric convolutions such as 3 by 1 convolution followed by 1 by 3 replaces 3 by 3 convolution.

3.3.1 AUXILARY CLASSIFIER

InInception-v3,auxiliary classifier isused as regularizer. Batch normalization,introduced in Inception v2,is also used in the auxiliary classifier.

3.3.2 EFFICIENT GRID SIZE REDUCTION

Usually feature map downsizing is done by maxpooling.But the approach either tends to be too greedy or too expensive. In inception v3 320 feature maps are obatined by max pooling and these are concatenated to obatin 640 feature maps. Efficient grid size reduction inInception v3 produces inexpensive yet effective network. 3) Training the network The deep convolutional model can be used to classify labels specific to the task at hand. For this theInception v3 model is loaded. New classes to be recognised are specified and Inception v3 model is trained over different batches for certain number of epoches, thus harnessing the image classifying abilities of Inception v3 to classify diseased plants.

3.4 TESTING

The trained model is tested on a set of images. Random images are introduced to thenetwork and output label is compared to the original known label of the image. Parameters used for evaluation are F1 score,precision and recall.Precision is the proportion of predicted positives that are truly positives. Recall gives the proportion of actual positives correctively classified.F1 score helps in maintaining a balance between precision and recall.

3.5 REQUIREMENT ANALYSIS

3.5.1 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements illustrate how a system must behave and create constraints of its functionality. This type of constraints is also known as the system's quality features. Attributes such as performance, security, usability, compatibility are notthe feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence wecan't compose a particular line of code to execute them. Any attributes required by the user are described by the specification.We must contain only those needs that are appropriate for our design.

Some Non-Functional Requirements areas follows:Availability, Maintainability, Performance, Portability Scalability, Flexibility.

3.5.3 AVAILABILITY

Availability is a general term used to depict how much an item, gadget, administration, or condition is open by however many individuals as would be prudent. In our venture individuals who have enrolled with the cloud can get to the cloud to store and recover their information with the assistance of a mystery key sent to their emai lids. UI is straight forward and productive and simple to utilize.

3.5.4 MAINTAINABILITY

In programming designing, viability is the simplicity with which a product item can be altered as:Correct absconds Meet new necessities New functionalities can be included in the task based the client necessities just by adding the proper documents to existing venture utilizing ASP. Net and C# programming dialects. Since the writing computer programs is extremely straight forward,it is simpler to discover and address the imperfections and to roll out the improvements in the undertaking.

3.5.5 SCALABILITY

Framework is fit for taking care of increment all out throughput under an expanded burden when assets(commonly equipment)are included.Framework can work ordinarily under circumstances, for example, low data transfer capacity and substantial number of clients.

3.5.6 PORTABILITY

Portability is one of the key ideas of abnormal state programming. Convenient is the product code base component to have the capacity to reuse the current code as opposed to making new code while moving programming from a domain to another. Venture can be executed under various activity conditions gave it meet its base setups.Just frame work records congregations would need to be designed in such case.

3.5.7 HARDWARE REQUIREMENTS

Processor	Any Processor Above 500 MHz
RAM	4GB
HardDisk	500 GB
System	Intel i3 6Gen 2.4 GHz

Table 1:Hardware Requirements

3.5.8 SOFTWARE REQUIREMENTS

Operating System	Windows
7/8/10/11 Programming language	Python,Machine Learning,CNN
IDE	Jupyter Notebook/Google Collab NoteBook
Tools	Anaconda,Pycharm

Table 2:Software Requirements

3.6 IMPLEMENTATION

The categorised leaves of tomato, potato, grape, and apple plants have 24 distinct types of labels. Information on Apple labels includes the following: healthy rust, scabs, and black rot. specifically: Cercospora of Corn Grey spot, healthy corn, corn blight, and corn rust. The individual grape labels are Leaf blight, Black rot, Esca, and healthy. The dataset consists of 31,119 images of various produce, including tomatoes, apples, maize, grapes, and potatoes. The images were downsized to 256 x 256 and divided into training and testing datasets with an 80-20 split. Out of the total dataset, 24,000 images were utilized for developing the CNN model.

The dataset includes images of plants affected by different pests and illnesses such as bacterial spot, early blight, healthy, late blight, leaf mould, septoria leaf spot, spider mite, target spot, mosaic virus, and yellow leaf curl virus. The objective of the model is to classify potato images into three categories: early blight, healthy, and late blight, which can help identify and manage diseases effectively. The convolution layer uses a convolution method to extract information. As the depth increases, the complexity of the recovered characteristics increases. The number of filters steadily rises as we move from one block to the next, but their size is constant at 5*5. There are 20 filters in the starting convolution block, 50 in the 2nd, and 80 in the 3rd.

The size of the feature maps was lowered as a result of the pooling layers being used in each of the blocks, which required more filters. After the convolution procedure is used, feature maps are null-padded to retain the dimesnions of the image. To shorten the length, utilise the max pooling layer.Transfer learning is a technique for sharing knowledge that employs 224*224 fixed-size pictures and requires the least amount of training data possible. Transfer learning is useful for transferring knowledge from one model to another. Sentiment analysis, activity recognition, software defect prediction, and plant categorization are just a few of the activities that have utilised transfer learning. In this study, the performance of the suggested Deep CNN model is compared to that of the well-liked VGG16 transfer learning technique. Three layers come after a stack of convolutional layers. The third device employs a 1000-way ILSVRC classification and has 1000 channels, compared to the preceding two devices' 4096 channels apiece. The last layer is the soft-max layer. The entirely connected layer design makes it easier to identify the leaf disease.

3.6.1 IMPORTING LIABRARIES

Import numpy as np, import pandas as pd,import os,from re import search, import shutil, import natsort, from PIL import Image, import matplotlib.pyplotasplt, from tqdm import tqdm, import cv2.

3.6.2 IMPORTING DATASET

df=r'D:\Python37\Projects\Foliardiseasesinappletrees\images\OriginalDataset'

train=pd.read_csv(r"D:\Python37\Projects\Foliardiseasesinappletree\labels\train.csv")

test=pd.read_csv(r"D:\Python37\Projects\Foliardiseasesinappletrees\labels\test.csv")

train.head()

```
train.head()
```

	image_id	healthy	multiple_diseases	rust	scab
0	Train_0	0	0	0	1
1	Train_1	0	1	0	0
2	Train_2	1	0	0	0
3	Train_3	0	0	1	0
4	Train_4	1	0	0	0

Fig 4: Train Head

test.head()

```
In [5]: test.head()
```

Out[5]:

	image_id
0	Test_0
1	Test_1
2	Test_2
3	Test_3
4	Test_4

Fig 5: Train Head

image1=Image.open(r'D:\Python37\Projects\Foliardiseasesinappletrees\images\OriginalDataset\Test_0.jpg')

plt.imshow(image1)

plt.show()

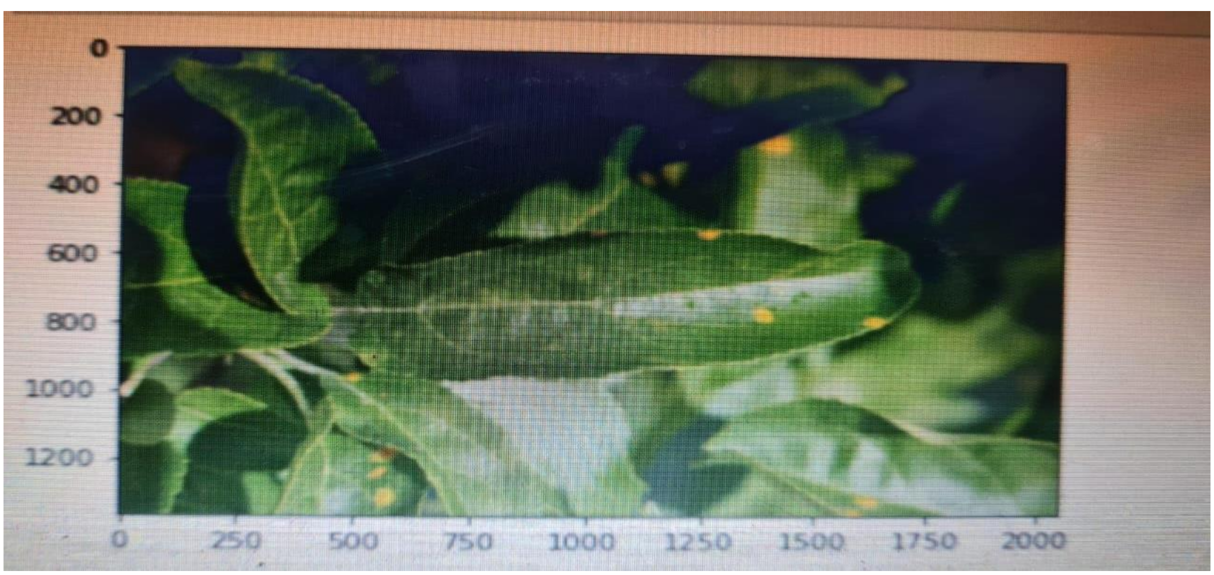


Fig 6: Image Showing

3.6.3 PREPARE THE TRAINING DATA

class_names = train.loc[:,'healthy':].columnsprint(class_names)

OUTPUT:Index(['healthy','multiple_diseases','rust','scab'],dtype='object') number=0

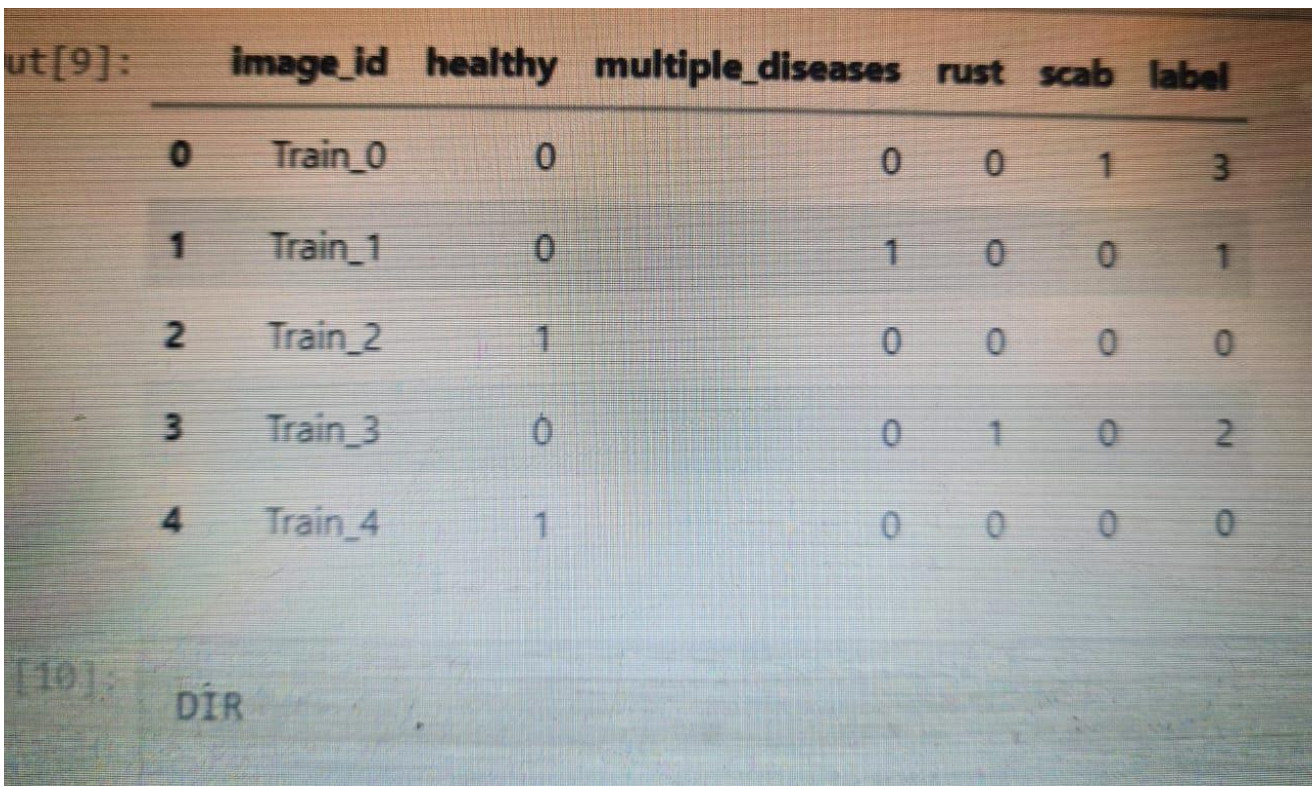
train['label']=0

for i in class_names:

train['label'] = train['label'] + train[i] * number

number = number+1

train.head()



	image_id	healthy	multiple_diseases	rust	scab	label
0	Train_0	0	0	0	1	3
1	Train_1	0	1	0	0	1
2	Train_2	1	0	0	0	0
3	Train_3	0	0	1	0	2
4	Train_4	1	0	0	0	0

Fig 7: Training Data

3.6.4 DIR

OUTPUT:'D:\\Python37\\Projects\\Foliar diseases in apple trees\\images\\Original Dataset' nat sort.nat sorted(os.listdir(DIR))

Output:

['Test_0.jpg', 'Test_1.jpg', 'Test_2.jpg', 'Test_3.jpg', 'Test_4.jpg', 'Test_5.jpg', 'Test_6.jpg', 'Test_7.jpg', 'Test_8.jpg', 'Test_9.jpg', 'Test_10.jpg', Test_11.jpg', 'Test_12.jpg','Test_13.jpg', 'Test_14.jpg', 'Test_15.jpg' ,'Test_16.jpg', 'Test_17.jpg' ,'Test_18.jpg', 'Test_19.jpg', 'Test_20.jpg', 'Test_21.jpg', 'Test_22.jpg', 'Test_23.jpg', 'Test_24.jpg', 'Test_25.jpg', 'Test_26.jpg', 'Test_27.jpg', 'Test_28.jpg', 'Test_29.jpg', 'Test_30.jpg', 'Test_31.jpg', 'Test_32.jpg', 'Test_33.jpg',]

3.6.5 CODE IMPLEMENTATION

```
def get_label_img(img):

if search("Train",img):

img=img.split('.')[0]

label=train.loc[train['image_id']==img]['label']

returnlabel

def create_train_data():

images=natsort.natsorted(os.listdir(DIR))

forimgintqdm(images):

label=get_label_img(img)

path=os.path.join(DIR,img)

ifsearch("Train",img):

if(img.split("_")[1].split(".")[0]) and label.item()==0:

shutil.copy(path,r'D:\Python37\Projects\Foliar diseases in apple trees\images\train\healthy')

elif(img.split("_")[1].split(".")[0]) and label.item()==1:

shutil.copy(path,r'D:\Python37\Projects\Foliardiseasesinapple trees\images\train\multiple_disease')

elif(img.split("_")[1].split(".")[0]) and label.item()==2:

shutil.copy(path,r'D:\Python37\Projects\Foliardiseasesinapple trees\images\train\rust')

elif(img.split("_")[1].split(".")[0]) and label.item()==3:

shutil.copy(path,r'D:\Python37\Projects\Foliardiseasesinappletrees\images\train\scab')

elif search("Test",img):
```

shutil.copy(path,r'D:\Python37\Projects\Foliardiseasesinappletrees\images\test')

shutil.os.mkdir(r'D:\Python37\Projects\Foliardiseasesinapple trees\images\train')

shutil.os.mkdir(r'D:\Python37\Projects\Foliar diseases in apple trees\images\train\healthy')

shutil.os.mkdir(r'D:\Python37\Projects\Foliardiseasesinapple trees\images\train\multiple_disease')

shutil.os.mkdir(r'D:\Python37\Projects\Foliar diseases in apple trees\images\train\rust')

shutil.os.mkdir(r'D:\Python37\Projects\Foliar diseases in apple trees\images\train\scab')

shutil.os.mkdir(r'D:\Python37\Projects\Foliardiseases inappletrees\images\test')

3.6.6 DATA PREPROCESSING

Train_DIR=r'D:\Python37\Projects\Foliardiseasesinapple trees\images\train'Categories=['healthy','multiple_disease','rust','scab']

for j in Categories:

path=os.path.join(Train_DIR,j)

for img in os.listdir(path):

old_image=cv2.imread(os.path.join(path,img),cv2.COLOR_BGR2RGB)

plt.imshow(old_image)

plt.show()

break

break

```
IMG_SIZE=224

new_image=cv2.resize(old_image,(IMG_SIZE,IMG_SIZE))

plt.imshow(new_image)

plt.show()
```

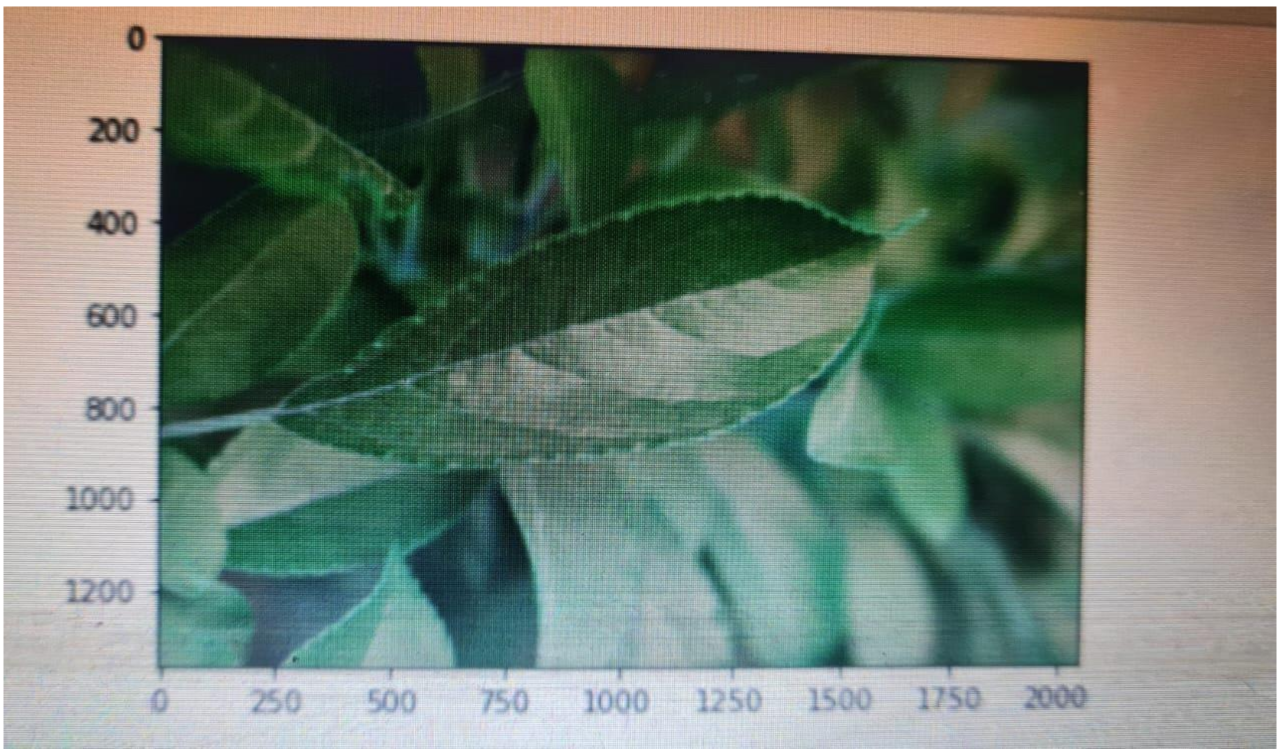


Fig 8: Data Preprocessing

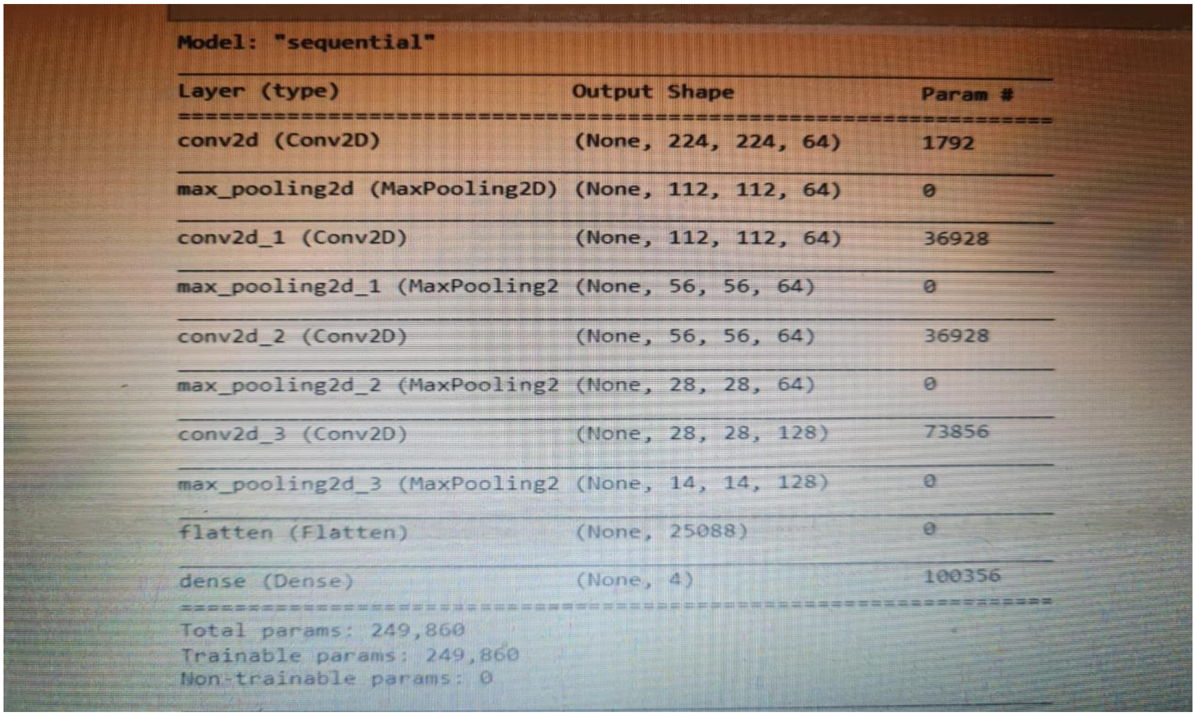
3.6.7 MODEL PREPRATION

```
Import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.callbacks

import ModelCheckpoint,EarlyStopping

from tensorflow.keras.preprocessing.image import Image DataGenerator

from tensorflow.keras.layers import Dense,Activation,Flatten,Conv2D,MaxPooling2D

datagen=ImageDataGenerator(rescale=1./255,

shear_range=0.2,

zoom_range=0.2,

horizontal_flip=True,

vertical_flip=True,

validation_split=0.2)

train_datagen=datagen.flow_from_directory(r'D:\Python37\Projects\Foliar diseases in apple trees\images\train',

target_size=(IMG_SIZE,IMG_SIZE),

batch_size=16,

class_mode='categorical',

subset='training')

val_datagen=datagen.flow_from_directory(r'D:\Python37\Projects\Foliar diseases in apple trees\images\train',
```

```
target_size=(IMG_SIZE,IMG_SIZE),

batch_size=16,

class_mode='categorical',

subset='validation')

model=Sequential()

model.add(Conv2D(64,(3,3),activation='relu',padding='same',input_shape=(IMG_SIZE,IMG_SIZE,3)))

model.add(MaxPooling2D(2,2))

model.add(Conv2D(64,(3,3),activation='relu',padding='same'))

model.add(MaxPooling2D(2,2))

model.add(Conv2D(64,(3,3),activation='relu',padding='same'))

model.add(MaxPooling2D(2,2))

model.add(Conv2D(128,(3,3),activation='relu',padding='same'))

model.add(MaxPooling2D(2,2))

model.add(Flatten())

model.add(Dense(4,activation='softmax'))
```

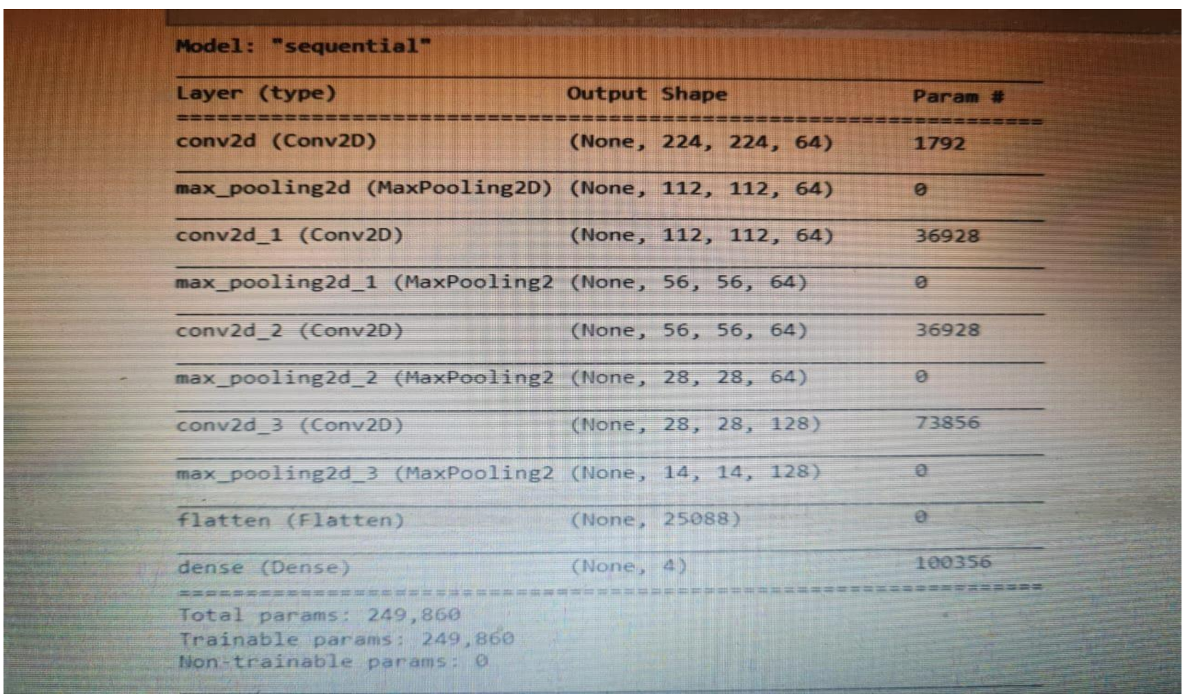
3.6.8 COMPILE THE MODEL

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
loss='categorical_crossentropy',

metrics=['accuracy'])

model.summary()
```



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 56, 56, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4)                 100356
=================================================================
Total params: 249,860
Trainable params: 249,860
Non-trainable params: 0
_________________________________________________________________
```

Fig 9: Model Preparation




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 56, 56, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4)                 100356
=================================================================
Total params: 249,860
Trainable params: 249,860
Non-trainable params: 0
_________________________________________________________________
```

3.6.9 TRAINING AND VALIDATION ACCURACY

```
acc_train=model_history.history['accuracy']

acc_val=model_history.history['val_accuracy']

epochs=range(1,31)

plt.plot(epochs,acc_train,'g',label='TrainingAccuracy')

plt.plot(epochs,acc_val,'b',label='ValidationAccuracy')

plt.title("TrainingandValidationAccuracy")

plt.xlabel("Epochs")

plt.ylabel("Accuracy")

plt.legend() plt.show()
```

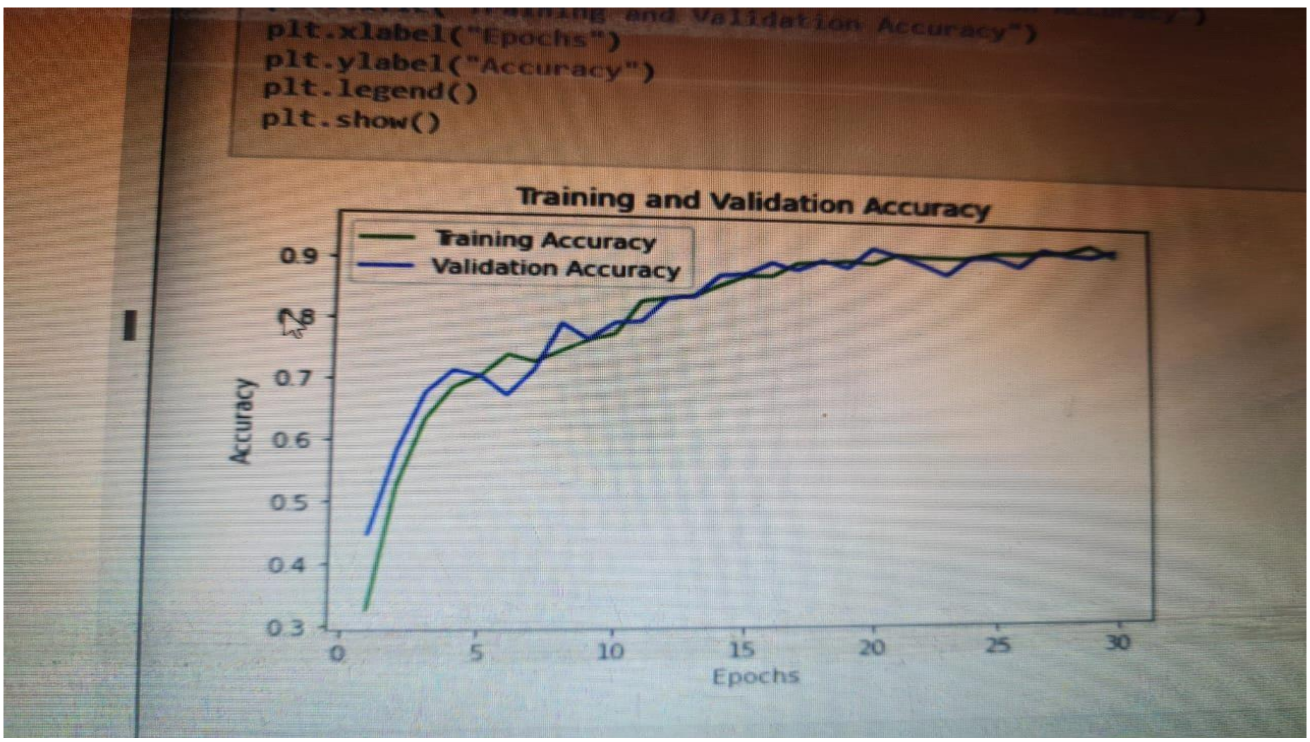


Fig 10: Training and Validation Accuracy

3.6.10 TRAINING AND VALIDATION LOSS

loss_train=model

history.history['loss']

loss_val=model_history.history['val__loss']

epochs=range(1,31)

plt.plot(epochs,loss_train,'g',label='TrainingLoss')

plt.plot(epochs,loss_val,'b',label='ValidationLoss')

plt.title("Training andValidation Loss")plt.xlabel("Epochs")plt.ylabel("Loss")plt.legend()

plt.show()

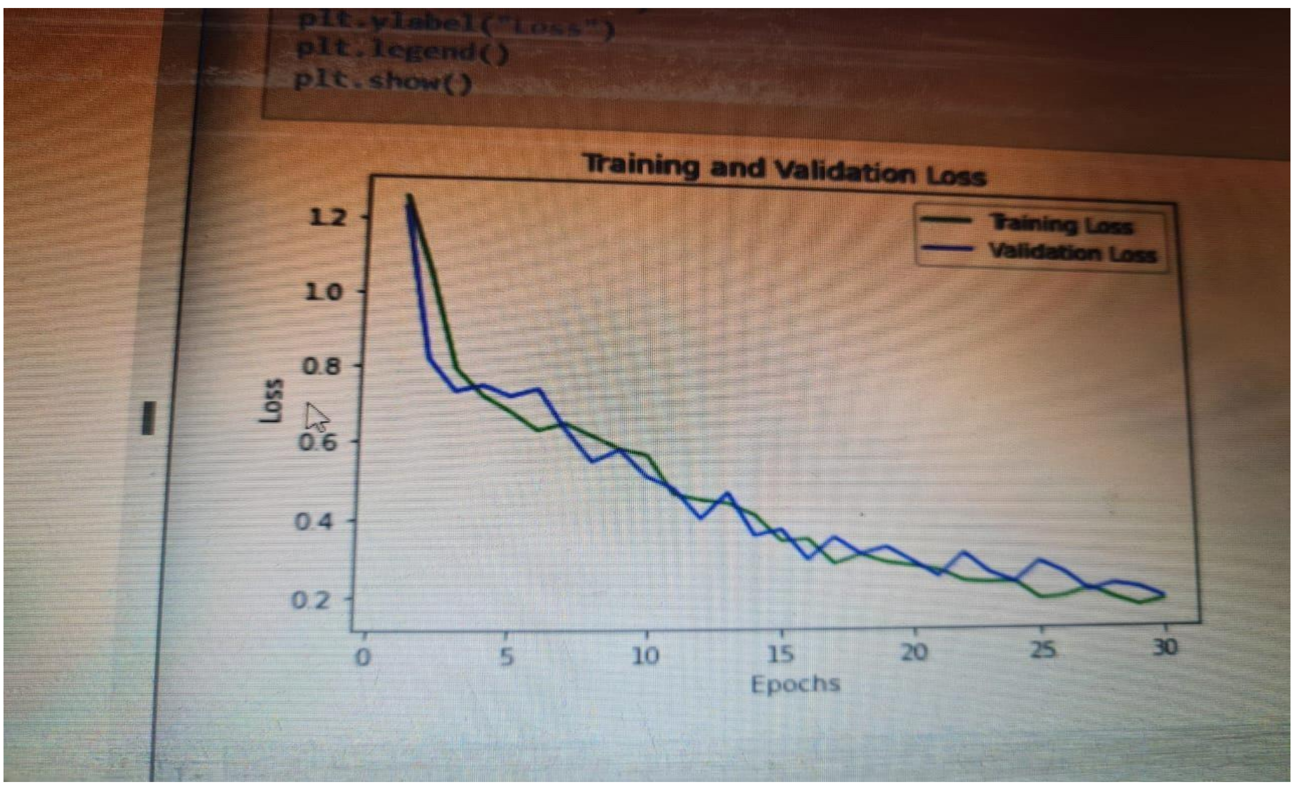


Fig 11: Training and Validation Loss

3.6.11 MAKING THE PREDICTIONON A SINGLE IMAGE

```
test_image=r'D:\Python37\Projects\Foliardiseasesinappletrees\images\train\rust\Train_3.jpg'

image_result=Image.open(test_image)

from tensorflow.keras.preprocessing

import imagetest_image=image.load_img(test_image,target_size=(224,224))

test_image=image.img_to_array(test_image)

test_image=test_image/255

test_image=np.expand_dims(test_image,axis=0)

result=model.predict(test_image)

print(np.argmax(result))

Categories=['healthy','multiple_disease','rust','scab']

image_result=plt.imshow(image_result) plt.title(Categories[np.argmax(result)])

plt.show()
```

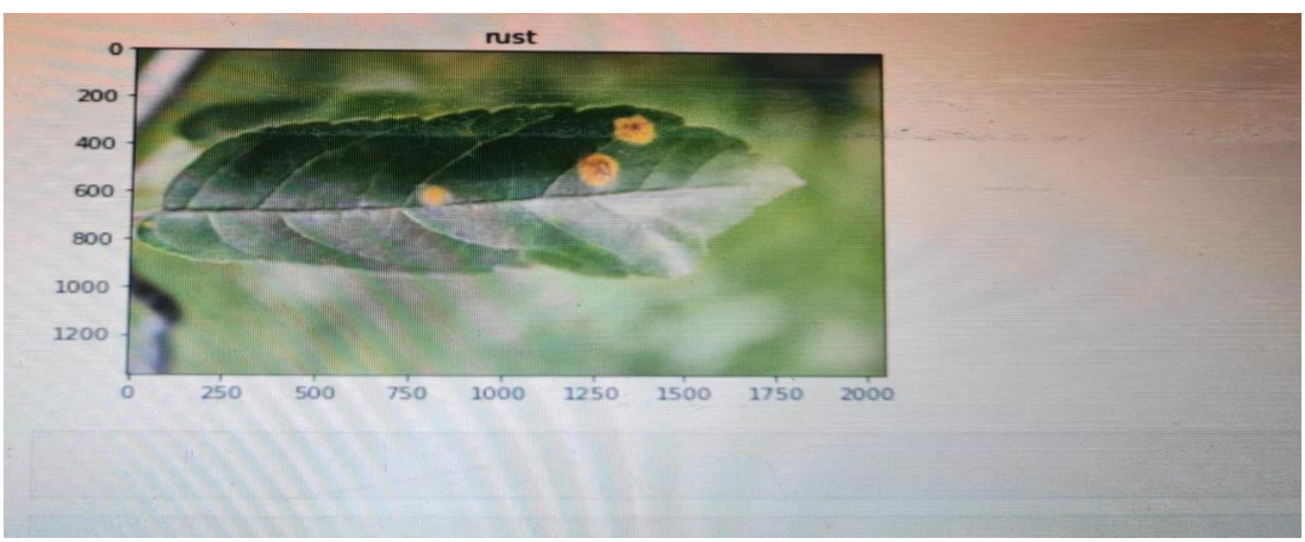


Fig 12: Prediction on Single Image

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION TO RESULTS

The table displays the performance metrics of each model developed for different plants, and it shows that the accuracy scores and f1 ratings are quite similar. However, a significant number of incorrect predictions, both positive and negative, contributed to this similarity. The average accuracy achieved was 93%, and confusion matrices were used to evaluate the number of true positives, true negatives, and accurate predictions. Additionally, the receiver operating characteristic (ROC) curve was plotted for each model to evaluate their performance at various classification thresholds.

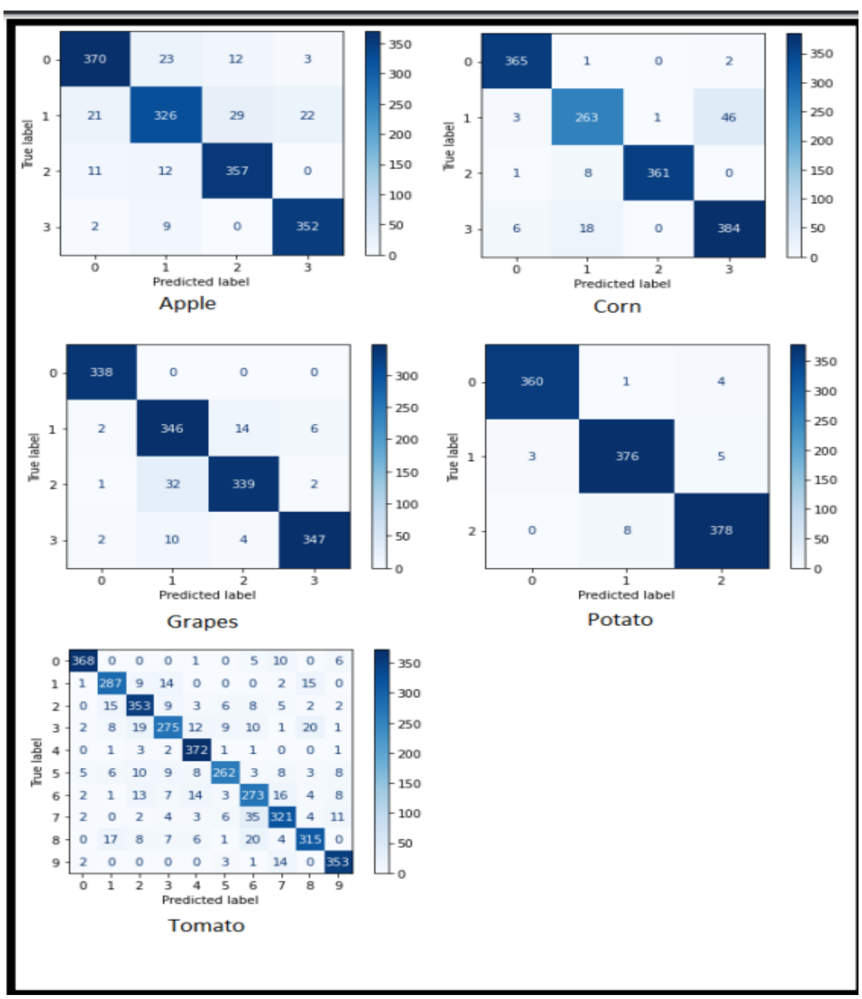


Fig 13: Prediction on Single Image

4.2 PERFORMANCE METRICS

Describe the evaluation metrics used to assess the performance of your disease detection models, such as accuracy, precision, recall, F1-score, and confusion matrix.Explain the significance of each metric and how they help evaluate the effectiveness of the models.

4.3 QUANTITATIVE RESULTS

Present the quantitative results of your models, including accuracy and other performance metrics.Compare the performance of different models or variations of your approach, highlighting any significant differences in performance.

4.4 QUALITATIVE RESULTS

Showcase visual examples of disease detection results, including images with correctly detected diseases and any challenging cases or false positives/negatives.Discuss the strengths and limitations of your models based on these examples, including instances where the model performed exceptionally well or struggled.

4.5 COMPARISON WITH BASELINES OR PRIOR WORK

If applicable, compare your results with baseline models or previous studies in the field of foliar disease detection.Discuss any improvements or advancements achieved by your models and the potential reasons behind them.

4.6 CONCLUSION

With the hope of increasing accuracy, the CNN (Alexnet) is being tested for the detection of leaf diseases. The database is split into two datasets, training and testing, using an 80/20 splitting ratio. CNN determines if a leaf is healthy or ill, and if so, it also forecasts the type of sickness. The CNN model was trained using a 10-epoch starting learning rate of 0.001. The activit of the CNN model on the testing dataset during training. Apple leaf confusion matrix illustration. 99% of the time, Apples leaves are accurate. Table 1 provides a summary of each plant's categorization accuracy. The total degree of accuracy is 97.71%. Examples of categorization made using convolutional neural networks on some randomly chosen photos from the testing dataset. The upper right corner of each image shows the accuracy percentage for the related plant leaves. By allowing for the early diagnosis of illnesses, this effort will help in the automatic identification of plant leaf disease and boost agricultural productivity. The accuracy of detecting tomato leaf disease may be improved by evaluating transfer learning and other CNN models.

Table 1: Classification Accuracy of leaves of plants

Plant name	Classification accuracy
Apple	99.0%
Cherry	99.4%
Corn	95.8%
Grape	99.7%
Peach	97.4%
Pepper bell	99.4%
Potato	98.7%
Strawberry	100%
Tomato	90.1%

Fig 14: Classification Accuracy of leaves of plants

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 **CONCLUSION**

This article examines the many paddy diagnosis techniques utilizing machine learning and image processing. Before using classification or image processing methods to detect the disease, the initial steps are characterized as picture acquisition, image processing, segmentation, and feature extraction. Several studies have changed pictures from the RGB color space to another color format or a grayscale since it is simple to obtain the threshold value of the histogram equation. In the process of processing images, noise has been recognized as a key issue that has to be resolved. It is essential for picture scaling and image improvement in machine learning classification research. The K-mean cluster approach is also utilized to remove the infected portions from the picture during the segmentation process. The Otsu method was used to choose the threshold value range. The three key characteristics that may be used to identify illnesses from a photograph of a paddy leaf are color, shape, and texture. SVM and ANN machine learning principles can be applied to diagnose problems using these features. The classification algorithm is mostly responsible for the diagnostic procedure' accuracy. A histogram equation with multiple values has provided some accuracy in differentiating rice leaf diseases in place of a machine learning process.

5.2 FUTURE SCOPE

This project will be very helpful to farmers in rural areasand also will help them in saving their yields from diseases as farmers lose a huge amount of their cultivated crops because of diseases and this system will help them to avoid the similar situation.Also we have tried to implement this project in a regional language so a stomake things more understandable for farmers.

Here are some points you can include:

5.2.1 IMPROVED MODELS AND ALGORITHMS

Explore advanced deep learning architectures specifically designed for plant disease detection, such as attention-based models or graph convolutional networks.Investigate the use of generative models, such as generative adversarial networks (GANs), for synthetic image generation and data augmentation in limited datasets.Research novel algorithms that can handle multi-class or multi-label classification for detecting multiple diseases in a single plant leaf.

5.2.2 TRANSFER LEARNING AND MODEL GENERALIZATION

Investigate the effectiveness of transfer learning by fine-tuning pre-trained models trained on large-scale image datasets, such as ImageNet, for foliar disease detection tasks. Explore methods to improve the generalization capabilities of disease detection models, allowing them to perform well on unseen plant species or disease variations.

5.2.3 REAL-TIME AND ON-DEVICE DISEASE DETECTION

Develop lightweight and efficient models suitable for deployment on edge devices, such as smartphones or low-power embedded systems, to enable real-time disease detection in the field.Investigate techniques like model compression, quantization, and knowledge distillation to reduce the model size and computational requirements while maintaining accuracy.

5.2.4 MULTI-MODAL APPROACHES

Explore the integration of multiple data sources, such as hyperspectral imaging, thermal imaging, or spectroscopy, along with visual images to improve disease detection accuracy and robustness.Investigate the fusion of plant physiological data, such as leaf temperature or chlorophyll content, with image-based approaches to enhance disease detection capabilities.

5.2.5 DATA COLLECTION AND ANNOTATION

Focus on creating larger and more diverse datasets that cover a wide range of plant species, disease types, and environmental conditions to improve the generalization and robustness of the models.Investigate semi-supervised or weakly supervised learning techniques to reduce the dependence on labor-intensive manual annotation by leveraging weak annotations or auxiliary information.

5.2.6 EXPLAINABILITY AND INTERPRETABILITY

Explore methods to provide interpretable explanations for disease detection models, enabling stakeholders to understand the decision-making process and build trust in the system.Investigate techniques, such as attention mechanisms or saliency maps, to visualize and highlight the regions of the image that contribute most to the disease detection.

5.2.7 DEPLOYMENT AND PRACTICAL IMPLEMENTATION

Conduct field trials and collaborate with agricultural experts to validate the performance of the disease detection models in real-world scenarios.Develop user-friendly software or mobile applications that can be used by farmers or agronomists to diagnose plant diseases on-site and provide appropriate recommendations.

REFERENCES

1.Literature Review of Diagnosis Rice Leaves Diseases Using Image Processing R.I.L.Jayasooriya, Samantha Mathara Arachchi University of Colombo School of Computing, 35, Reid Avenue, Colombo 7, Sri Lanka.

2.Plant Disease Detection Using Image Processing and Machine Learning Pranesh Kulkarni1,Atharva Karwande1,Tejas Kolhe1,Soham Kamble1,Akshay Joshi1,Medha Wyawahare1 1 Department of Electronics and Telecommunication,Vishwakarma Institute of Technology, Pune, India.

3.Image-Based Crop Leaf Disease Identification Using Convolution Encoder Network written by Indira Bharathi and Veeramani Sonai Reviewed: August 9th, 2022 Published: October 21st, 2022.

4.Plant Disease Detection using Machine Learning by Ms.Nilam Bhise1,Ms.Shreya Kathet2,Mast.Sagar Jaiswar3,Prof.Amarja Adgaonkar4 1 Student, Department of Information Technology,Excelsior Education Society's K.C. College of Engineering & Management Studies & Research, Thane, Maharashtra, India.

5.Plant Leaf Disease Detection using Deep Learning by Mr. Thangavel. M -AP/ECE ,Gayathri P K ,Sabari K R, Prathiksha V from Knowledge Institute of Technology,Salem.

6.Implementation of Improved Machine Learning Techniques for Plant Disease Detection and Classification Ibrahim M. Adekunle Department of ICT,Osun State University, Nigeria.

7.https://www.irjet.net/archives/V9/i4/IRJET-V9I4403.pdf.

8.https://www.researchgate.net/publication/327065422_Plant_Disease_Detection_Using_Machine_Learning.

9.https://www.researchgate.net/publication/344282301_Leaf_Disease_Detection_Using_Machine_Learning.

10.https://arxiv.org/ftp/arxiv/papers/2106/2106.10698.pdf

APPENDEX

FOLIAR DISEASE DETECTION USING ML AND DEEP LEARNING

KIET Group of Institutions, Ghaziabad

Prof.Shalini Kapoor

[1] Umang Pratap Singh,[2] Tushar Kundoo,[3] Saurabh Mandal

Abstract

The classification methods that may be used to classify plant leaf diseases are surveyed in this research. The 7.6 billion people on the planet may be fed with the help of contemporary farming techniques. People continue to experience malnutrition despite having access to enough food. Plant diseases affect both the quantity and the quality of the overall harvest. A number of challenges must be overcome when developing an image processing model for prediction or classification applications. For a farmer, it might be challenging to recognize illness signs visually. A computerized image processing technology is used for crop protection in big frames so that unhealthy leaves may be identified utilizing the color information of the leaves. There are several classification methods, including the SVM, Probabilistic Neural Network, k-Nearest Neighbor Classifier Genetic Algorithm and Principal Component Analysis. Because diverse input data might produce results of varying quality, choosing a classification technique is always a challenging undertaking. Classifications of plant leaf diseases are widely used in many industries, including agriculture, biotechnology, and scientific research.

Keywords: K- means clustering, ANN, SVM, Neural network.

Introduction

Plant diseases cause yield reductions that have a direct influence on the domestic and international food production systems and lead to financial losses. About 20% to 40% of the world's food output is lost due to plant diseases and pests, according to the FAO of the United Nations has reported that 13% of global crop yield losses are due to plant diseases. This highlights the importance of identifying and preventing plant diseases to minimize these losses. One method for identifying plant diseases is by analyzing images of plant leaves, using a technique called "image processing" which falls under the field of signal processing. By leveraging the power of artificial intelligence, specifically machine learning, we can *extract meaningful information from these images to accurately detect and diagnose plant diseases and thinking performs tasks itself or provides instructions on how to carry them out. Understanding the training data and incorporating it into models that should be helpful to humans is the basic goal of machine learning. Thus, it may help in making wise selections and forecasting the right output utilizing the vast training data. Leaf color, leaf damage level, leaf area, and leaf texture characteristics are utilized for classification. Several forms of plant diseases damage various plant organs. Plant pathologists can most easily identify foliar diseases, which are plant diseases that manifest symptoms on leaves. Fungal diseases are a major cause of yield losses, accounting for up to 50% of the total losses. As a result, many researchers are using computer vision, machine learning, and deep learning techniques to detect and diagnose plant diseases using images of plant leaves. Effective diagnosis of plant diseases involves early detection of diseases, identifying multiple diseases in different crops, estimating the severity of the disease, determining the appropriate amount of pesticide to apply, and taking practical measures to manage the disease and prevent its spread.*

Existing Work

Prior studies on detecting leaf damage using CNN provides an example of how to recognize ans classify leaf disease using image processing techniques like as

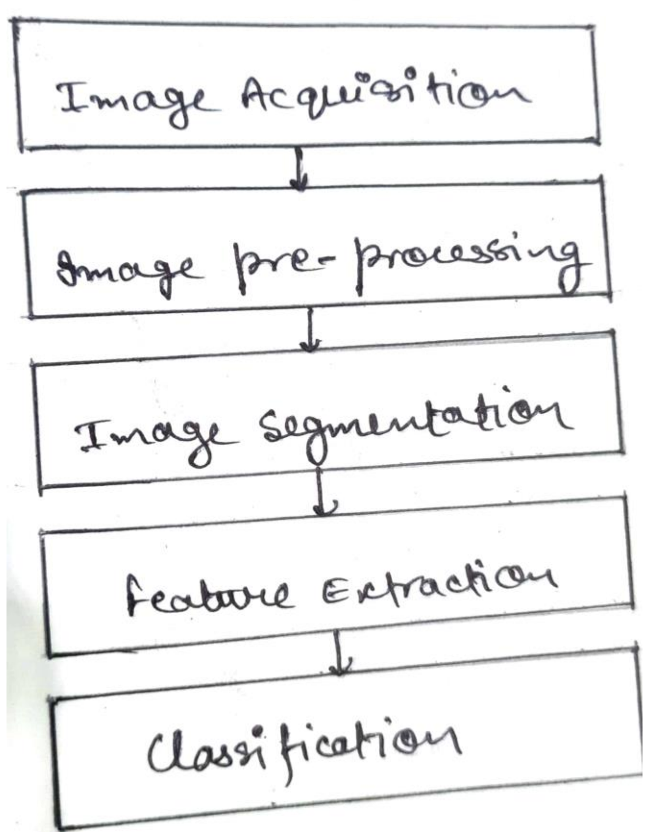


Fig: Block Diagram of Feature Based Approach

Image acquisition refers to the phenomena of capturing an image and storing it digitally, such as on a digital camera or other digital medium. Before any processing can be done on the image, it is necessary to pre-process it in order to improve its quality and remove any unwanted distortions. The important objective of image pre-processing is to enhance the desired characters of the image and improve the details contained within it, making it more suitable for subsequent processing and analysis using tools such as MATLAB. Many approaches are used in preprocessing, such as dynamic image size improving image and morphological processes, noise filtering, image conversion, and image enhancement. Image segmentation is employed in K-means clustering is a method for grouping several images so that at least one of the clusters has an image with a significant portion of an unhealthy region. Application of the k means cluster algorithmic method results in the classification of the objects into K different categories for every set of qualities. Following the formation of clusters, GLCM is used to extract texture characteristics.

Related Works

In this section, we discuss pertinent initiatives in categorization problems utilizing deep learning architectures. Deep learning techniques have generally been the subject of much research for applications such as object recognition and image categorization. When used to solve recognition and classification issues, convolutional neural networks (CNNs), a deep learning technology, achieve state-of-the-art performance in picture classification.The first CNN architecture known as MobileNet for object recognition was evaluated using the dataset for tomato disease. To determine the degree of tomato leaf disease from photos of tomato leaves, pre-trained CNN architectures VGG16, MobileNet, and ResNet50 were implemented. Performance was improved by adding ResNet50 features to the traditional CNN model.

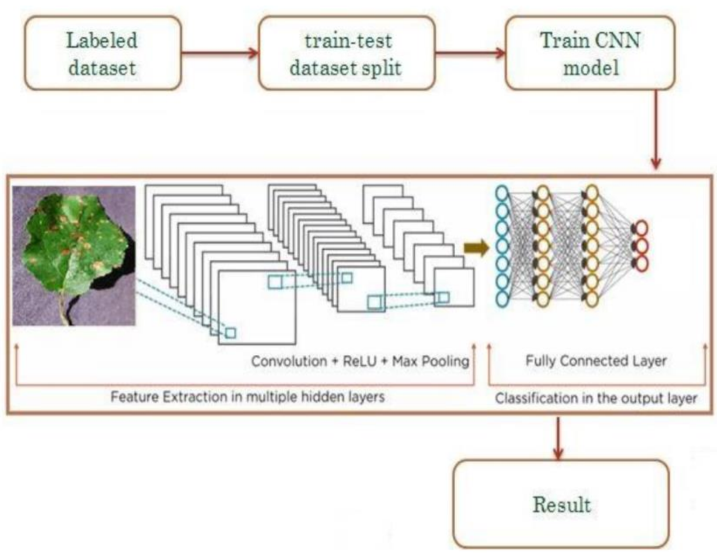


Fig: Proposed Workflow

The spatial links between the image's constituent parts are not taken into consideration by CNN architectures, making them ineffective for geometric transformations. By routing features from one layer to another in CNN, the max-pooling layer has a tendency to lose data. They are unable to model the rotational invariance of an item.The section presents a Capsule Network with Dynamic Routing algorithm to alleviate the shortcomings of CNN design. Capsule networks were used in the experiments to classify illnesses based on medical imaging, and they performed better than regular CNN in doing so.

Dataset

In this survey, we utilized the PlantVillage dataset, which is openly attainable collection of images for identifying plant leaf diseases. The dataset was curated and maintained by Sharada P. Mohanty and others, and contains over 87,000 RGB photos of both healthy and diseased plant leaves, with 38 different disease classes. However, for the purpose of our experiment, we selected only 25 disease classes to test our method, which are listed in the table.

Table - Dataset Specifications.

Plant	Disease Name	No. of Images
Apple	Healthy	2008
	Diseased Scab	2016
	Diseased: Black rot	1987
	Diseased: Cedar apple rust	1760
Corn	Healthy	1859
	Diseased: Cercospora leaf spot	1642
	Diseased: Common rust	1907
	Diseased: Northern Leaf Blight	1908
Grapes	Healthy	1692
	Diseased: Black rot	1888
	Diseased: Esca (Black Measles)	1920
	Diseased: Leaf blight (Isariopsis)	1722
Potato	Healthy	1824
	Diseased: Early blight	1939
	Diseased: Late blight	1939
Tomato	Healthy	1926
	Diseased: Bacterial spot	1702
	Diseased: Early blight	1920
	Diseased: Late blight	1851
	Diseased: Leaf Mold	1882
	Diseased: Septoria leaf spot	1745
	Diseased: Two-spotted spider mite	1741
	Diseased: Target Spot	1827
	Diseased: Yellow Leaf Curl Virus	1961
	Diseased: Tomato mosaic virus	1790

The model consists of 5 phases:

A.Feature Extraction and Data Preprocessing

For computer vision-based systems to yield accurate results, it is crucial to prepare the data correctly. One critical aspect of data preparation involves removing background noise from the image before extracting the essential features. By converting an RGB image to grayscale, the image is simplified, making it easier to process.The thresholding technique is then used to binaries the picture. The minor gaps in the foreground are then filled using morphological transform on the binarized picture. Texture, Shape and color attributes are now retrieved from the picture following segmentation. The parameter and area of a leaf are estimated using contours. A outline is a line that connects all the points on the boundaries of objects that share the equal hue or devotion. In addition, the standard and mean deviation of each RGB color channel are also calculated. The image is initially transformed from RGB to HSV color space, and the proportion of pixels with hue (H) channel pixel intensities ranging from 30 to 70 is calculated and divided by the total number of pixels in that channel. This is done to determine the quantity of green color present in the image. Calculating the non-green portion of a picture involves removing the green component from 1.

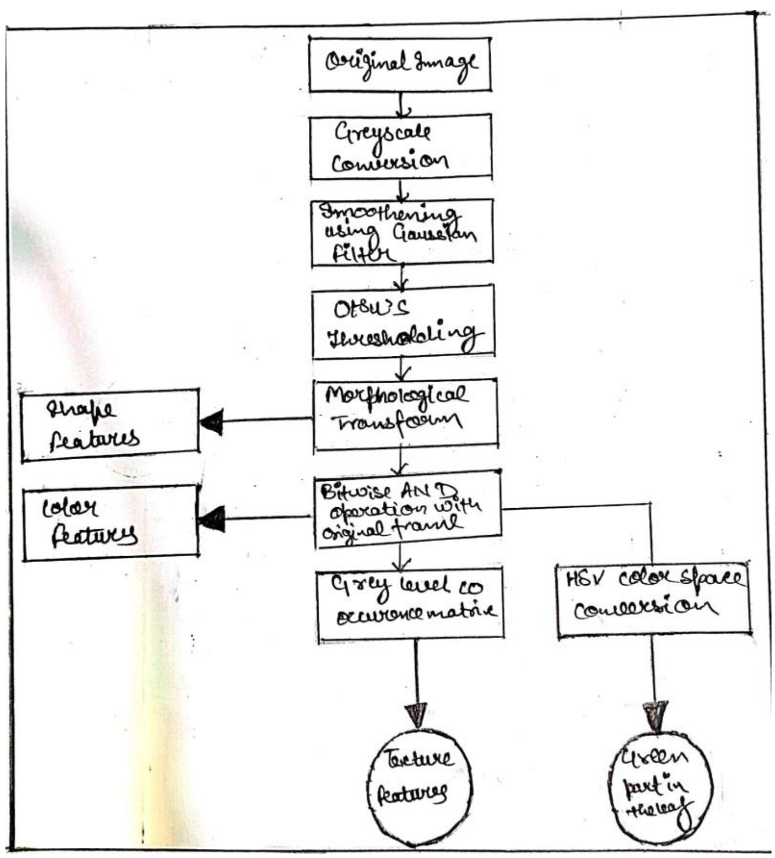


Fig: Steps for feature extraction and data processing

B.Image Pre-Processing

Unwanted noise has been eliminated from photos that have been gathered using image pre-processing. Research has proposed a number of concept preparation methods. An optical inspection's dependability can be enhanced by pre-processing of image.A more simple or quick review consists of many filter processes that highlight or diminish certain visual elements. With a few clicks, users may easily improve a camera image. Many graphics processes, such as cropping, rotating, normalizing, contrast boosting, filtering, and angle correction, are involved. Digital photographs can include noises like dust, dewdrops, and insect faces that can be removed using image preprocessing. Distortion and Noises from shadow effects and water drops can also be removed using various types of noise reduction filters. The real image was converted into a new color space using primarily three image pre-processing techniques. This new color space is basically similar to the original image but varies in certain ways. As previously said, picture resizing, image restoration, and image enhancement include: Steps in this process of engagement.

1. Resize

Original photos were downsized to a fixed resolution of 640 × 480 pixels to better fit the available processing and memory resources.

2. Noise Restoration

When a camera and an object move, the shutter opens improperly, the environment is disturbed, and the focus is off, all of these things can produce noise.

3. Image Enhancement

For the purpose of improving digital photographs for presentation or subsequent study utilized image enhancement.

$(x) = 0.114 * B + 0.2989*R + 0.5870*G$

Eqn: 1 RGB to grey conversion equation

C.Disease detection and classification

Disease identification is carried out in two phases, namely the kind of crop and the type of illness. Convolutional Neural Network is used to facilitate this. Transfer learning will be used to develop the model. It is a method in which the existing models are used to build the new ones. Classification also functions as fully linked classifiers that are created utilizing a variety of model learning. By flattening the photos, we achieve the following by creating vectors with a single dimension from the pooled images. It becomes much simpler to categorize the photographs once they have been turned to vectors. We obtain *specific numerical values in relation to distinct classes using the trained model. If a leaf is healthy, it will be labeled as such without any further classification. However, if there is a disease present, black dots on a gray scale will indicate it and the disease will be classified with a certain level of confidence. This classification process uses two numerical arrays and determines whether the leaf is healthy or sick based on the dataset provided. Identifying plant diseases through classification is a crucial and efficient process that provides accurate results.*

D.Image Segmentation

The photos have been divided into parts for examination using image segmentation. In accordance with the necessary characteristics, images have been transformed into another format. The steps involved in segmenting an image include background removal, and picture analysis. In favour to distinguish the contaminated region from the background, image partition is carried out by choosing the proper threshold range. The bottom and top of the picture histogram are used to select the threshold values. One method at the threshold has been shown to be unsuccessful due to the non-uniform distribution of the illness zone. As a result, a partition-level entry depend on partition of pixels was suggested.The query image is separated into a division for disease and a section for health based on a fuzzy logical grading design. The threshold has been set at the valley's bottom if the histogram shows a severe and deep valley between two peaks. If not, it is impossible to apply this technique to make items stand out from the backdrop. As a result, the Otsu technique has automatically chosen the best threshold value.

E.Image Analysis and Diagnosis

In photos of paddy leaves, the histogram as well as the intensity and saturation parts of the green, blue and red components are determined. Nevertheless, the only factor that affects how accurate the results are is the Hue components.As a result judgements were solely based on Hue histogram analysis.With the help of the extracted characteristics, the image is examined to see if any of the three disorders mentioned above are present. In order to segregate paddy leaf photos depending on the infected illness, color histogram and pixel layout were utilized. To extract contours from photos, utilize the Border tracing technique. Before examining an image's characteristics, the image has been turned into a histogram.

Implementation work

The categorised leaves of tomato, potato, grape, and apple plants have 24 distinct types of labels. Information on Apple labels includes the following: healthy rust, scabs, and black rot. specifically: Cercospora of Corn Grey spot, healthy corn, corn blight, and corn rust. The individual grape labels are Leaf blight, Black rot, Esca, and healthy. The dataset consists of 31,119 images of various produce, including tomatoes, apples, maize, grapes, and potatoes. The images were downsized to 256 x 256 and divided into training and testing datasets with an 80-20 split. Out of the total dataset, 24,000 images were utilized for developing the CNN model. The dataset includes images of plants affected by different pests and illnesses such as bacterial spot, early blight, healthy, late blight, leaf mould, septoria leaf spot, spider mite, target spot, mosaic virus, and yellow leaf curl virus. The objective of the model is to classify potato images into three categories: early blight, healthy, and late blight, which can help identify and manage diseases effectively.

*The convolution layer uses a convolution method to extract information. As the depth increases, the complexity of the recovered characteristics increases. The number of filters steadily rises as we move from one block to the next, but their size is constant at 5*5. There are 20 filters in the starting convolution block, 50 in the 2nd, and 80 in the 3rd. The size of the feature maps was lowered as a result of the pooling layers being used in each of the blocks, which required more filters. After the convolution procedure is used, feature maps are null-padded to retain the dimesnions of the image. To shorten the length, utilise the max pooling layer.*

*Transfer learning is a technique for sharing knowledge that employs 224*224 fixed-size pictures and requires the least amount of training data possible. Transfer learning is useful for transferring knowledge from one model to another. Sentiment analysis, activity recognition, software defect prediction, and plant categorization are just a few of the activities that have utilised transfer learning. In this study, the performance of the suggested Deep CNN model is compared to that of the well-liked VGG16 transfer learning technique. Three layers come after a stack of convolutional layers. The third device employs a 1000-way ILSVRC classification and has 1000 channels, compared to the preceding two devices' 4096 channels apiece. The last layer is the soft-max layer. The entirely connected layer design makes it easier to identify the leaf disease. All hidden layers have the ability to rectify. Re-Lu It should be noted that none of the networks use Local Response Normalization (LRN), which has no positive effects on the dataset's performance. Network nonlinearity exists in repaired linear units..*

Results and discussion

The table displays the performance metrics of each model developed for different plants, and it shows that the accuracy scores and f1 ratings are quite similar. However, a significant number of incorrect predictions, both positive and negative, contributed to this similarity. The average accuracy achieved was 93%, and confusion matrices were used to evaluate the number of true positives, true negatives, and accurate predictions. Additionally, the receiver operating characteristic (ROC) curve was plotted for each model to evaluate their performance at various classification thresholds. The ROC curve is a graphical representation of a classification model's performance, with the true positive rate and false positive rate being the two key parameters used to assess the model's efficacy.

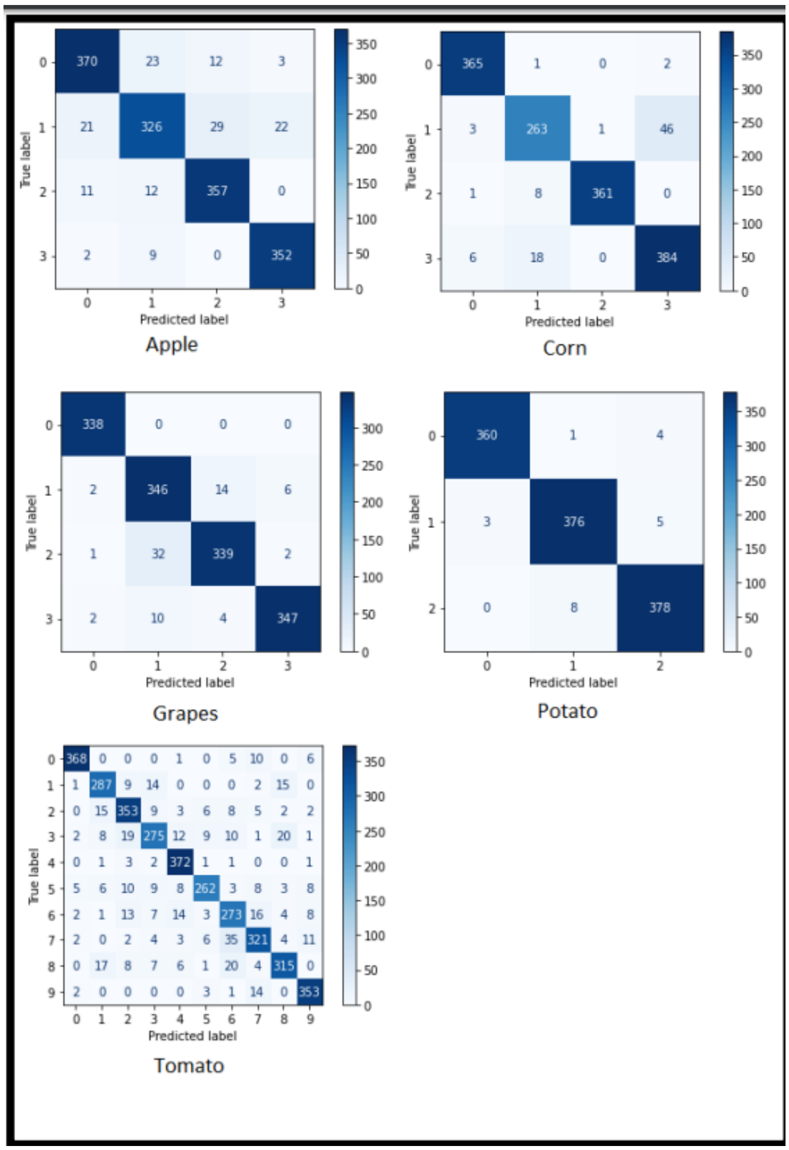


Fig: Confusion matrices for all the models

With the hope of increasing accuracy, the CNN (Alexnet) is being tested for the detection of leaf diseases. The database is split into two datasets, training and testing, using an 80/20 splitting ratio. CNN determines if a leaf is healthy or ill, and if so, it also forecasts the type of sickness. The CNN model was trained using a 10-epoch starting learning rate of 0.001. The activit of the CNN model on the testing dataset during training. Apple leaf confusion matrix illustration. 99% of the time, Apples leaves are accurate. Table 1 provides a summary of each plant's categorization accuracy. The total degree of accuracy is 97.71%. Examples of categorization made using convolutional neural networks on some randomly chosen photos from the testing dataset. The upper right corner of each image shows the accuracy percentage for the related plant leaves. By allowing for the early diagnosis of illnesses, this effort will help in the automatic identification of plant leaf disease and boost agricultural productivity. The accuracy of detecting tomato leaf disease may be improved by evaluating transfer learning and other CNN models.

Plant name	Classification Accuracy
Apple	99.0%
Cherry	99.4%
Corn	95.8%
Grape	99.7%
Peach	97.4%
Pepper bell	99.4%
Potato	98.7%
Strawberry	100%
Tomato	90.1%

<ins>**Table: Classification Accuracy of leaves of plants**</ins>

Conclusion

This article examines the many paddy diagnosis techniques utilizing machine learning and image processing. Before using classification or image processing methods to detect the disease, the initial steps are characterized as picture acquisition, image processing, segmentation, and feature extraction. Several studies have changed pictures from the RGB color space to another color format or a grayscale since it is simple to obtain the threshold value of the histogram equation. In the process of processing images, noise has been recognized as a key issue that has to be resolved. It is essential for picture scaling and image improvement in machine learning classification research. The K-mean cluster approach is also utilized to remove the infected portions from the picture during the segmentation process. The Otsu method was used to choose the threshold value range. The three key characteristics that may be used to identify illnesses from a photograph of a paddy leaf are color, shape, and texture. SVM and ANN machine learning principles can be applied to diagnose problems using these features. The classification algorithm is mostly responsible for the diagnostic procedure' accuracy. A histogram equation with multiple values has provided some accuracy in differentiating rice leaf diseases in place of a machine learning process.

References

1.Literature Review of Diagnosis Rice Leaves Diseases Using Image Processing R.I.L.Jayasooriya, Samantha Mathara Arachchi University of Colombo School of Computing, 35, Reid Avenue, Colombo 7, Sri Lanka.

2.Plant Disease Detection Using Image Processing and Machine Learning Pranesh Kulkarni1,Atharva Karwande1,Tejas Kolhe1,Soham Kamble1,Akshay Joshi1,Medha Wyawahare1 1 Department of Electronics and Telecommunication,Vishwakarma Institute of Technology, Pune, India.

3. Image-Based Crop Leaf Disease Identification Using Convolution Encoder Network written by Indira Bharathi and Veeramani Sonai Reviewed: August 9th, 2022 Published: October 21st, 2022.

4.Plant Disease Detection using Machine Learning by Ms.Nilam Bhise1,Ms.Shreya Kathet2,Mast.Sagar Jaiswar3,Prof.Amarja Adgaonkar4 1 Student, Department of Information Technology,Excelsior Education Society's K.C. College of Engineering & Management Studies & Research, Thane, Maharashtra, India.

5.Plant Leaf Disease Detection using Deep Learning by Mr. Thangavel. M -AP/ECE ,Gayathri P K ,Sabari K R, Prathiksha V from Knowledge Institute of Technology,Salem.

6.Implementation of Improved Machine Learning Techniques for Plant Disease Detection and Classification Ibrahim M. Adekunle Department of ICT,Osun State University, Nigeria.