PLANT DISEASE PREDICTION

A Capstone Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

by

P. AJAY GOUD	**(2103A54008)**
D.POORNA CHANDAR	**(2103A54014)**
G. SAI PRIVIJITH	**(2103A54018)**
V. NATARAJ	**(2103A54041)**
K. SHIVA KUMAR	**(2103A54L02)**

Under the guidance of

Mr. Nafis Uddin Khan

Associate Professor, School of CS&AI.

sru SR UNIVERSITY

SR University, Ananthsagar,Warangal,Telagnana-506371

SR University

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this project entitled **"PLANT DISEASE PREDICTION"** is the bonafied work carried out by **POCHANAGARI AJAY GOUD, POORNA CHANDAR, SAI PRIVIJTH, VENAGANTI NATARAJ** as a Capstone Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **School of Computer Science and Artificial Intelligence** during the academic year 2024-2025 under our guidance and Supervision.

Mr. Nafis Uddin Khan
Associate Professor
School of CS&AI
SR University
Anathasagar,Warangal

Dr. M.Sheshikala
Professor & Head
School of CS&AI,
SR University
Ananthasaga, Warangal.

Reviewer-1
Name:
Designation:
Signature:

Reviewer-2
Name:
Designation:
Signature:

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our Capstone project guide **Mr. Nafis Uddin Khan, Assoc Professor** as well as Head of the School of CS&AI , **Dr. M.Sheshikala, Professor** and Dean of the School of CS&AI, **Dr.Indrajeet Gupta Professor** for guiding us from the beginning through the end of the Capstone Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project co-ordinators **Mr. Sallauddin Md, Asst. Prof., and R.Ashok Asst. Prof.** for their encouragement and support.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

Table of Contents

LIST OF FIGURES

I. INTRODUCTION

I.I Overview

Plant disease prediction is a study that includes analysing data linked to numerous illness that afflict Plants using machine learning techniques. The purpose of this research is to create an identification model that can effectively diagnose and categories plant diseases based on plant symptoms.

The initial stage in this endeavour is to collect information on the many illnesses that impact each of these plants. This might include gathering photographs of afflicted plants as well as information on the symptoms and environmental circumstances linked with disease.

After gathering this information, it may be utilized to train a machine learning model This model will learn to recognize patters in data and use these patterns to forecast the chance of a certain plant being impacted by a disease.

The model's accuracy is determined by the quality of the data used to train it. As a result, it is critical to verify that the data is correct and reflective of the various illnesses that might harm these plants.

Once trained, the model can be used to predict whether a specific plant will be affected by a specific disease. This may be accomplished by feeding data about the plant's symptoms and environmental variables into the model and then analysing the output to predict illness.

Overall, the plant disease identification project is an effective tool for detecting and treating plant illnesses in lemon trees, rose bushes, and cactus plants. It is conceivable to build a more accurate and effective way for diagnosing and treating plant illnesses by analysing data relating to these diseases using machine learning techniques.

I.II Problem Statement

Plant disease cause significant losses in crop yields, which ultimately affect the global food supply and economy. Early detection and accurate diagnosis of plant diseases are crucial for effective disease management. However, the traditional methods for identifying plant diseases are time-consuming, expensive, and require specialized skills.

To address this challenge, the problem statement is to develop a reliable and accurate system for predicting plant diseases based on leaf images. The system should be able to classify the type of disease and its severity with high accuracy, to facilitate early diagnosis and prompt management. Additionally, the system should be scalable and easily deployable to enable farmers and agricultural workers to use it in the field.

I.III. Existing System

There are now several approaches for predicting plant illness, especially for lemon plants, rose bushes, and cactus plants efc. These systems primarily rely on traditional illness detection methods, such as visual examination and expert manual diagnosis.

One of the major drawbacks of old approaches is that they are typically time-consuming and prone to mistakes owing to the subjective nature of human diagnosis. Furthermore, traditional methods may not always be accurate in identifying diseases in their carly stages, resuting in delayed treatment and potentially irreversible plant damage.

Recent improvements in machine learning and computer vision have resulted in the creation of automated plant discase identification systems to solve these difficulties. Image recognition algorithms are used in these systems to analyze plant photos and detect the presence of disease signs.

The Plant Village project, for example, employs a combination of machine learning algorithms and human experience to identify plant problems. This technology provides an online platform for farmers and other users to upload photographs of their plants and obtain a diagnosis of the illness that is harming them.

Despite their advantages, automated plant disease identification systems have some limitations. These systems, for example, may be ineffective in identifying diseases that are not well-represented in the training data, or in cases where the plant is affected by multiple diseases at the same time.

Overall, while some systems for plant disease identification are already in place, more research and development in this area is needed to improve the accuracy and effectiveness of these systems.

I.IV. Proposed System

Based on current technologies, a suggested system for plant disease identification would entail the creation of a mobile app thar combines machine learning and image recognition to diagnose illnesses in lemon plants, rose bushes, and cactus plants etc.

The software would allow users to photograph their plant and submit it to the system. which would then analyze the image and deliver a diagnostic of the plant's illness. The app would also give information on treatment alternatives and illness preventive techniques.

The machine learning algorithms would be taught using a vast collection of photos of healthy and ill plants, spanning a wide spectrum of discases that might afflict these plants, to assure the system's accuracy. The system would also be continuously updated with fresh data to guarantee that it is up to date with the most recent plant disease information.

In addition to diagnostic capabilities, the app may offer users with plant care and maintenance information, as well as connect them with local experts and resources more assistance.

Overall, a smartphone app that combines machine learning and picture recognition to identify plant illnesses would provide a rapid, accurate, and accessible tool for farmers, gardeners and other plant lovers to diagnose and cure diseases in lemon plants, rose bushes, and cactus plants.

I.V Objectives

- Create a database with images and information on common diseases that affect lemon trees, rose bushes, cactus plants etc, as well as their symptoms and environmental requirements.
- Using the database, train machine learning models to effectively detect and identify the presence of illnesses in plants.
- Create a mobile or web-based application that allows users to upload images of their plants and receive a disease diagnosis.
- Improve the models' accuracy by frequently updating the database with fresh information on plant illnesses and symptoms.
- Provide consumers with information about illness treatment choices and preventative techniques.
- Conduct research to better understand plant diseases and their causes, as well as to find new ways to prevent and treat them.

The goals of a plant disease identification project for lemon plants, rose bushes, cactus plants etc. are to provide an accurate, accessible, and efficient tool for identifying and treating plant illnesses, as well as to increase plant health knowledge and understanding.

I.VI. Architecture

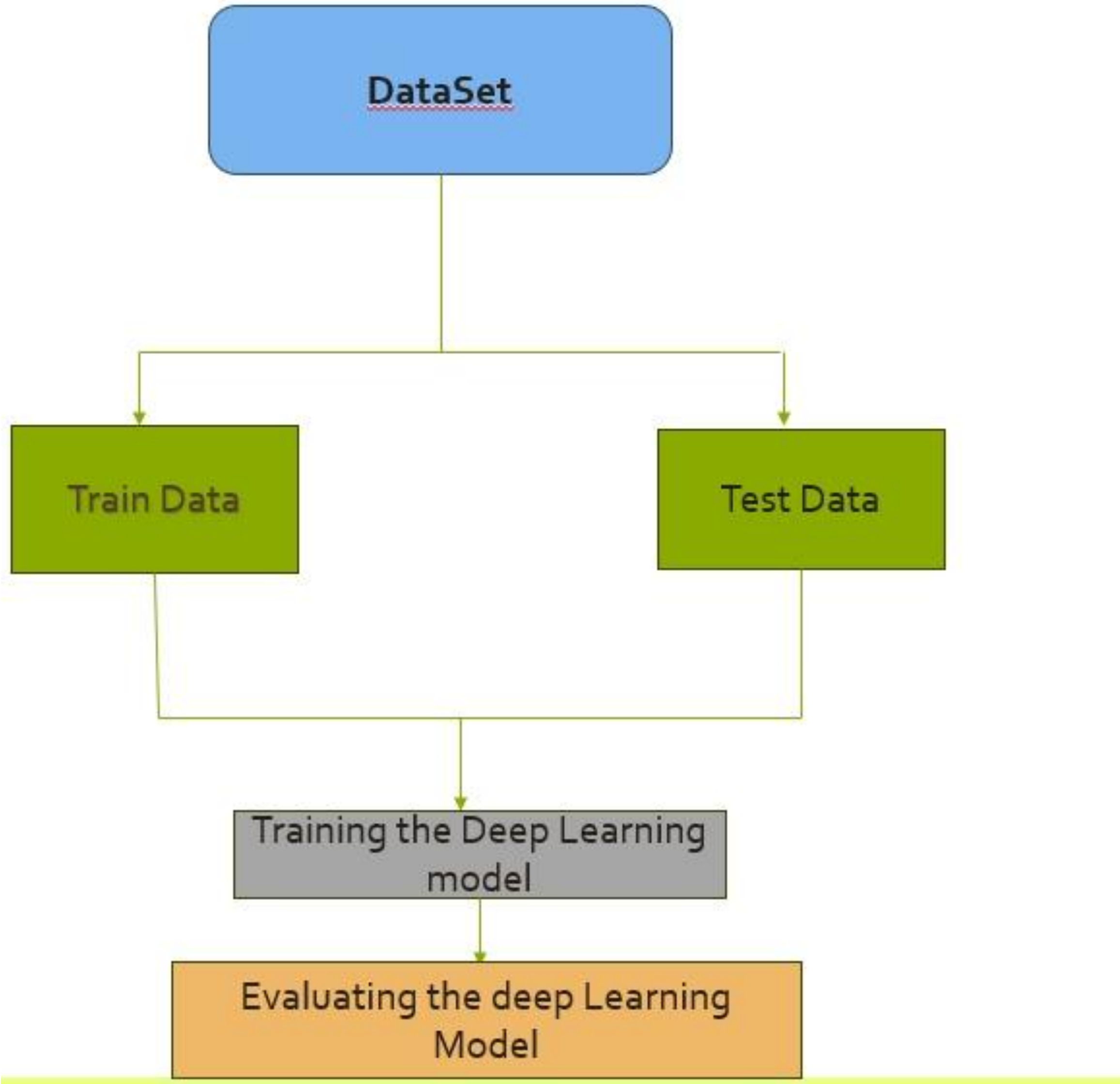

Fig:1.1

II. LITERATURE SURVEY

II.I. Survey Document

Plant disease identification is an essential topic in agriculture because it allows farmers to identify and control illnesses that might reduce crop output. Machine learning and artificial intelligence (Al) approaches have been applied to plant discase identification in recent years in order to increase the accuracy and efficiency of discase detection. We will examine some of the most current research findings in the subject of plant disease identification utilising AI and machine learning in this literature review.

- **P.T. Tran et al, "Deep Learning for Plant Disease Detection and Diagnosis" (2019).** This article presents a deep learning strategy for detecting and diagnosing plant diseases using Convolutional Neural Networks (CNNs). The scientists created a collection of S4,306 photos of healthy and ill plants to train and test their algorithm. The results indicated that their deep learning technique surpassed typical machine learning algorithms in terms of accuracy.
- **N.S. Sutar and S.A. Jadhav's "Automated Plant Disease Diagnosis: A Review Using Machine Learning Methods" (2021).** This review paper presents an overview of current advances in plant disease detection using machine learning approaches. The authors cover machine learning methods such as Decision Trees, Random Forests, and Support Vector Machines (SVMs) that are utilised for plant discase identification. They also emphasise the field's difficulties and prospects, as well as make recommendations for future research.
- **S. S. Mishra and S. K. Rout's "Deep learning for plant disease identification using transfer learning and data augmentation" (2021).**
  Using transfer learning and data augmentation approaches, this work suggests a deep learning solution for plant discase diagnosis. The scientists created a collection of 38,535 photos of healthy and ill plants to train and test their algorithm. The results indicated that their deep learning technique surpassed typical machine learning algorithms in terms of accuracy.

- **K. Ray e al (2019) published "Deep learning for plant disease diagnosis: A comparative study of Convolutional Neual Networks with transfer learning**
  "This study compares deep leaming techniques for plant disease diagnosis utilising CNNs and transfer learning, The scientists Created a collection of 7,631 photos of healthy and ill plants to train and test their algorithms. The deep learning strategy using transfer learning beat the other approaches and attained excellent accuracy according to the results.
- **N. Saravanan and R. Ramaswamy's (2021) paper, "Plant disease detection using image processing and machine learning techniques: A review**
  "The current state of plant discase detection utilising image processing and machine learning approaches is discussed in this review study. The authors present an overview of the different picture pre-processing, feature extraction, and classification approaches, They also examine the field's difficulties and prospects, as well as future research directions.

2.2 Conclusion:

To summarise, machine learning and Al techniques have shown considerable promise in plant discase identification, with recent research demonstrating their efficiency. However, challenges remain, such as the need for large and diverse datasets, model interpretability and robustness to environmental variations. Future research should concentrate on overcoming these obstacles and building more accurate and efficient plant disease identification algorithms.

III. DATA PRE-PROCESSING

III.I Description of Dataset

III.I.I Size of Data:

We gathered our image data from several sources, and the overall size of our dataset is roughly 500 photographs, which we tagged with two distinct categories. Each group has roughly similar proportion of data. We used 90% of the data for training and testing the model, and 10% for validation.

III.I.II Variety of plant species:

We gathered data from three distinct species: lemon plants, rose bushes, and cactus plants. These three plants belong to three separate families, and there are less datasets available on them.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator


train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
[ ] train_generator = train_datagen.flow_from_directory('/content/drive/MyDrive/archive/Train/Train',
                                                    target_size=(225, 225),
                                                    batch_size=32,
                                                    class_mode='categorical')

    validation_generator = test_datagen.flow_from_directory('/content/drive/MyDrive/archive/Validation/Validation',
                                                        target_size=(225, 225),
                                                        batch_size=32,
                                                        class_mode='categorical')
```

```
Found 1332 images belonging to 3 classes.
Found 60 images belonging to 3 classes.
```

Fig:3.1

III.I.III Data collection process:

We searched various websites for datasets but were unable to locate any accurate datasets. So, we went to a nearby nursery and worked on collecting real datasets. We used mobile cameras and went during the day to get clear image data.

III.I.IV Data labelling:

We have divided the data into 2 categories i.e .. images with healthy leaves and images with unhealthy leaves. The images which are labelled healthy is said to be as healthy leaves and the images with the unhealthy leaves are said to be as the unhealthy leaves.

III.I.V Data split:

we have split the entire data in to train and validation set. Which is the train data containing of 90% of dataset and the validation dataset containing of 10%, By the splitting of the dataset into these above-mentioned formats the model performance will be better.

III.II Data Cleaning:

There is no need of data cleaning process for our data set because its image data set and there will be no outliers present for the image dataset. so there is no need of dataset.

III.III Data Augmentation

Data augmentation is a frequent strategy used in machine learning, particularly plant' disease identification in agricultural Al. Data augmentation is applying different modifications to existing data in order to expand the training dataset and improve the model's capacity to generalise to new data.

Data augmentation in the context of plant disease identification might include techniques such as flipping, rotating, or cropping photographs of plants to produce fresh training instances. Colour modifications, blurring, and adding noise to photos can also be used to produce new variants of the original data.

The model may learn to recognise the patterns and properties of healthy and ill plants by increasing the quantity of raining data through data augmentation. This can lead 1o better crop management practises and increased accuracy in forecasting plant diseases.

Overall, data augmentation is a potent strategy for boosting the accuracy and robustness of AI models for plant disease identification, and it is widely employed in this sector to assist in the resolution of the pressing problem of crop disease detection and control.

Following Code:

```
from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

import numpy as np

import tensorflow as tf
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import os
def total_files(folder_path):
    num_files = len([f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path,
f))])
    return num_files
#getting path of the dataset

train_files_healthy = "/content/drive/MyDrive/archive/Train/Train/Healthy"
train_files_powdery = "/content/drive/MyDrive/archive/Train/Train/Powdery"
train_files_rust = "/content/drive/MyDrive/archive/Train/Train/Rust"

test_files_healthy = "/content/drive/MyDrive/archive/Test/Test/Healthy"
test_files_powdery = "/content/drive/MyDrive/archive/Test/Test/Powdery"
test_files_rust = "/content/drive/MyDrive/archive/Test/Test/Rust"

valid_files_healthy = "/content/drive/MyDrive/archive/Validation/Validation/Healthy"
valid_files_powdery = "/content/drive/MyDrive/archive/Validation/Validation/Powdery"
valid_files_rust = "/content/drive/MyDrive/archive/Validation/Validation/Rust"

print("Number of healthy leaf images in training set", total_files(train_files_healthy))
print("Number of powder leaf images in training set", total_files(train_files_powdery))
print("Number of rusty leaf images in training set", total_files(train_files_rust))
```

```
print("===========================================================")


print("Number of healthy leaf images in test set", total_files(test_files_healthy))
print("Number of powder leaf images in test set", total_files(test_files_powdery))
print("Number of rusty leaf images in test set", total_files(test_files_rust))


print("===========================================================")


print("Number of healthy leaf images in validation set", total_files(valid_files_healthy))
print("Number of powder leaf images in validation set", total_files(valid_files_powdery))
print("Number of rusty leaf images in validation set", total_files(valid_files_rust))
from PIL import Image
import IPython.display as display


image_path = '/content/drive/MyDrive/archive/Test/Test/Healthy/8ddaa5a5caa5caa8.jpg'


with open(image_path, 'rb') as f:
    display.display(display.Image(data=f.read(), width=500))
image_path = '/content/drive/MyDrive/archive/Test/Test/Rust/82add70df6ab2854.jpg'


with open(image_path, 'rb') as f:
    display.display(display.Image(data=f.read(), width=500))
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

from keras.models import Sequential

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

```
[ ]  from matplotlib import pyplot as plt
     from matplotlib.pyplot import figure

     import seaborn as sns
     sns.set_theme()
     sns.set_context("poster")

     figure(figsize=(6, 6))

     plt.plot(history.history['accuracy'])
     plt.plot(history.history['val_accuracy'])
     plt.title('model accuracy')
     plt.ylabel('accuracy')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```

Fig:3.2

model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(225, 225, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

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

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

```
model.add(Dense(3, activation='softmax'))
history = model.fit(train_generator,
          batch_size=32,
          epochs=20,

          validation_data=validation_generator,
          validation_batch_size=32
          )
model.save("disease_disease_recognition.h5")
from matplotlib import pyplot as plt
from matplotlib.pyplot import figure

import seaborn as sns
sns.set_theme()
sns.set_context("poster")

figure(figsize=(6, 6))

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
```

```
plt.show()

from tensorflow.keras.preprocessing.image import load_img, img_to_array

import numpy as np


def preprocess_image(image_path, target_size=(225, 225)):

    img = load_img(image_path, target_size=target_size)

    x = img_to_array(img)

    x = x.astype('float32') / 255.

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

    return x


x                                                                                    =
preprocess_image('/content/drive/MyDrive/archive/Train/Train/Powdery/8192743f44f1d9ad.j
pg')


# displaying the testing image

image_path = '/content/drive/MyDrive/archive/Train/Train/Powdery/8192743f44f1d9ad.jpg'


with open(image_path, 'rb') as f:

    display.display(display.Image(data=f.read(), width=500))

predictions = model.predict(x)

predictions[0]

labels = train_generator.class_indices

labels = {v: k for k, v in labels.items()}

labels
```

predicted_label = labels[np.argmax(predictions)]

print(predicted_label)

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory('/content/drive/MyDrive/archive/Train/Train',
                                                    target_size=(225, 225),
                                                    batch_size=32,
                                                    class_mode='categorical')

validation_generator = test_datagen.flow_from_directory('/content/drive/MyDrive/archive/Validation/Validation',
                                                        target_size=(225, 225),
                                                        batch_size=32,
                                                        class_mode='categorical')
```

```
Found 1332 images belonging to 3 classes.
Found 60 images belonging to 3 classes.
```

Fig:3.3

```
history = model.fit(train_generator,
                    batch_size=32,
                    epochs=20,

                    validation_data=validation_generator,
                    validation_batch_size=32
                    )
model.save("disease_disease_recognition.h5")
```

```
Epoch 1/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 230s 5s/step - accuracy: 0.8410 - loss: 0.3980 - val_accuracy: 0.8667 - val_loss: 0.5410
Epoch 2/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 219s 5s/step - accuracy: 0.9337 - loss: 0.2171 - val_accuracy: 0.8333 - val_loss: 0.4566
Epoch 3/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 230s 5s/step - accuracy: 0.9425 - loss: 0.1977 - val_accuracy: 0.8667 - val_loss: 0.3149
Epoch 4/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 217s 5s/step - accuracy: 0.9065 - loss: 0.2396 - val_accuracy: 0.8000 - val_loss: 0.4550
Epoch 5/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 216s 5s/step - accuracy: 0.9104 - loss: 0.2508 - val_accuracy: 0.8667 - val_loss: 0.4042
Epoch 6/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 225s 5s/step - accuracy: 0.9610 - loss: 0.1330 - val_accuracy: 0.8833 - val_loss: 0.3551
Epoch 7/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 222s 5s/step - accuracy: 0.9743 - loss: 0.0807 - val_accuracy: 0.8667 - val_loss: 0.4837
Epoch 8/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 227s 5s/step - accuracy: 0.9647 - loss: 0.1036 - val_accuracy: 0.9000 - val_loss: 0.3694
Epoch 9/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 252s 5s/step - accuracy: 0.9807 - loss: 0.0648 - val_accuracy: 0.9333 - val_loss: 0.2276
Epoch 10/20
42/42 ━━━━━━━━━━━━━━━━━━━━ 260s 5s/step - accuracy: 0.9738 - loss: 0.1007 - val_accuracy: 0.9167 - val_loss: 0.4084
```

Fig:3.4

```
[ ] from matplotlib import pyplot as plt
    from matplotlib.pyplot import figure

    import seaborn as sns
    sns.set_theme()
    sns.set_context("poster")

    figure(figsize=(6, 6))

    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('model accuracy')
    plt.ylabel('accuracy')
    plt.xlabel('epoch')
    plt.legend(['train', 'val'], loc='upper left')
    plt.show()
```

Fig:3.5

Fig:3.6

```
[ ] with open(image_path, 'rb') as f:
        display.display(display.Image(data=f.read(), width=500))
```

```
predictions = model.predict(x)
predictions[0]
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 187ms/step
array([1.4180335e-04, 9.9984300e-01, 1.5176247e-05], dtype=float32)
```

Fig:3.7

```python
with open(image_path, 'rb') as f:
    display.display(display.Image(data=f.read(), width=500))
```

```python
predictions = model.predict(x)
predictions[0]
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 187ms/step
array([1.4180335e-04, 9.9984300e-01, 1.5176247e-05], dtype=float32)
```

```python
labels = train_generator.class_indices
labels = {v: k for k, v in labels.items()}
labels
```

```
{0: 'Healthy', 1: 'Powdery', 2: 'Rust'}
```

```python
predicted_label = labels[np.argmax(predictions)]
print(predicted_label)
```

```
Powdery
```

Fig:3.8

IV. METHODOLOGY

IV.I Procedure to solve the given problem

We have followed following steps in our project to get our ultimate goal of Predicting plant disease:

1. **Define the scope and objectives of the project:**
   Determine the ypes of diseases and plants to be included, the target users, and the expected outcome.
2. **Collect data on plant diseases;**

   Gather information on common diseases affecting lemon plants, rose bushes, and cactus plants, including their symptoms, causes, and treatments.
3. **Gather image data:**
   Collect a large dataset of images of healthy and diseased plants, covering a wide range of diseases that can affect these plants.
4. **Importing necessary libraries:**

   Import the necessary libraries such as Pandas, NumPy, Matplot, Open vision(cv2), Keras, TensorFlow, Os, Seaborn etc, for reading and visualizing the data.
5. **Label the image data:**
   Annotate the images with labels indicating the presence of diseascs, as well as anyother relevant information.
6. **Train machine learning models:**
   Develop and train machine learning algorithms using the labelled image data to accurately predict and diagnose the presence of diseases in the plants.
7. **Test and validate the models:**

   Evaluate the performance of the models on a separate test dataset to assess their accuracy and effectiveness.
8. **Improve the accuracy of the models:**

   Regularly update the database with new information on plant discases and their symptoms to improve the accuracy of the models.

V.II MODEL ARCHITECTURE

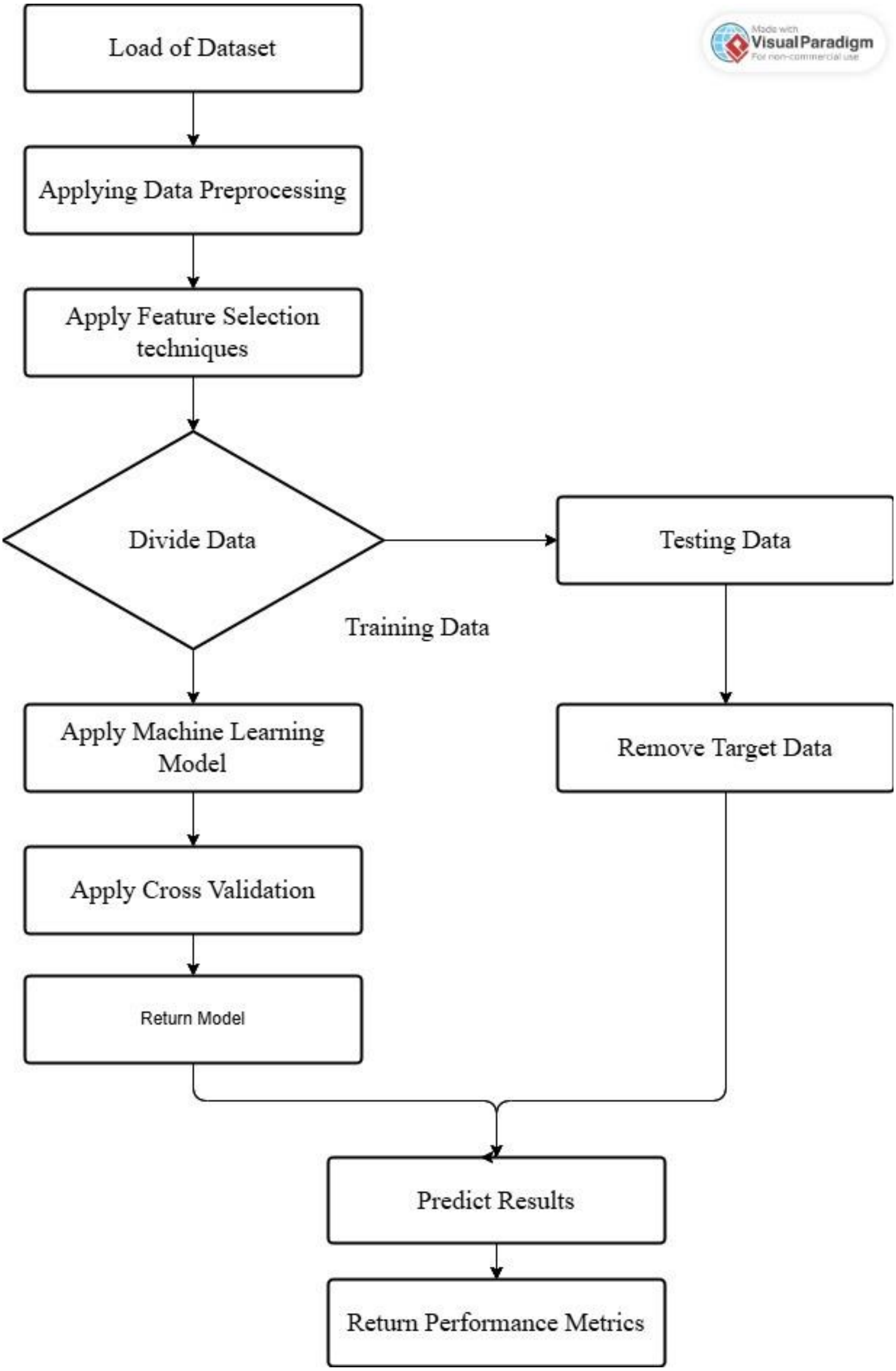

Fig 4.1

IV.III Software Description

1. Python:- Python is an interpreted, high-level, general-purpose programming language. Python design philosophy emphasizes code readability with its notable use of significance whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large scale projects.
   Python is dynamically typed and supports multiple programming paradigms, including procedural, object-oriented and functional programming.

2. GOOGLE COLAB: Colaboratory or Colab for short from Google Research. Colab allows anybody io write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

3. VS CODE -Microsoft Visual Studio Code (VS Code) is a popular, free and open- source code editor that supports a broad range of programming languages and frameworks. It has extensive developer capabilities including as debugging, syntax highlighting, auto-completion, version control, and extensions. It is available for Windows, macOS, and Linux, and it has a sizable and active user and developer community that contributes to its continuous development and improvement

V. RESULTS AND DISCUSSION

We Utilized CNN model to asses the accuracy value as well as the visulaization samples. We choose CNN because it is reasonable simple to comprehend and use and we can readily obtain reliable value samples of the visualised data

Sample images of the output:

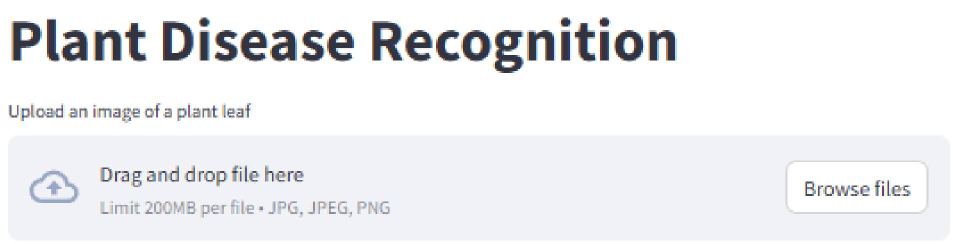

Fig: 5.1

Upload an image of a plant leaf

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

9e84aee68691979c.jpg 0.8MB

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Uploaded Image

Prediction: Powdery

Prediction probabilities:

Healthy: 0.00

Powdery: 1.00

Rust: 0.00

Fig:5.2

Plant Disease Recognition

Upload an image of a plant leaf

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Leaf spot disease.jpg 81.5KB

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.

Uploaded Image

Prediction: Rust

Prediction probabilities:

Healthy: 0.00

Powdery: 0.00

Rust: 1.00

Fig:5.3

Plant Disease Recognition

Upload an image of a plant leaf

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

8f83c2ee646433e3.jpg 0.8MB

×

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Uploaded Image

Prediction: Healthy

Prediction probabilities:

Healthy: 0.54

Powdery: 0.00

Rust: 0.46

Fig:5.4

Healthy Leaf **Powdery Leaf** **Rusty Leaf**

Fig: 5.5

VI. CONCLUSION AND FUTURE SCOPE

In the current day scenario, plant disease diagnosis using machine learning models has the potential to revolutionise agriculture by allowing farmers to diagnose and treat plant diseases rapidly and reliably. This technique can improve the efficiency and efficacy of plant disease control by utilising the power of image recognition algorithms and massive databases.

In the future, there is a lot of room for more study and growth in this field. Continuously expanding the information used to train the models, for example, can enhance their accuracy and capacity to identify a wider spectrum of illnesses. Additionally incorporating other data sources, such as weather patterns and soil conditions, can improve model accuracy. Furthermore, robust and dependable hardware systers that can be deployed in the field to capture high-quality images of plants and transmit them to machine learning models for analysis are needed. Furthermore, the use of mobile and web-based applications can make this technology more accessible to farmers and other users.

VII. REFERNCES

1) Acharya, A., et al., 2020, November. Plant disease detection for paddy crop using ensemble of CNNs. 2020 IEEE International Conference for Innovation in Technology (INOCON). IEEE, pp. 1–6.
2) Akhter, M.S., et al., 2019. Plant virus diseases and their management in Bangladesh. Crop Prot. 118, 57–65.
3) Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. August). Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET). Ieee, pp. 1–6.
4) Alegbejo, M.D., et al., 2006. Rice yellow mottle virus disease, a new disease of rice in Zamfara, Nigeria. Intern. Rice Res. Notes 31 (39), 39.
5) Alfred, R., et al., 2021. Towards paddy rice smart farming: a review on big data, machine learning, and rice production tasks. IEEE Access 9, 50358–50380.
6) Al-Shemarry, Meeras Salman, Li, Yan, Abdulla, Shahab,Wen, Peng, 2019. An efficient texture descriptor for the detection of license plates from vehicle images in difficult conditions. IEEE Transactions on Intelligent Transportation Systems. Vol 99, pp. 1–12.
7) Arnal Barbedo, J.G., 2013. Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus 2 (1), 1–12.
8) Asfarian, A., et al., 2014. A computer vision for rice disease identification to support integrated pest management. Crop Prot. 61, 103–104.
9) Atila, Ü., et al., 2021. Plant leaf disease classification using EfficientNet deep learning model. Ecol. Inform. 61, 101182.
10) Springer, Cham, pp. 638–645. Kumar, J.P., Domnic, S., 2019. Image based leaf segmentation and counting in rosette plants. Inform. Proc. Agricult. 6 (2), 233–246.
11) Chen, J., et al., 2020. Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimed. Tools Appl. 79 (41), 31497–31515.
12) Chollet, F., 2017. Xception: deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258.
13) https://www.bing.com/ck/a!&&p=51537602d80999c5JmltdHM9MTY4MiY0MDAWMCZpZ3VpZD0xNTNmOTBkYS01NzZiLTY0N2QtMDA0Yy04MiBlNTYzOTY1NWUmaW5zaWO9NTIyMg&ptn=3&hsh=3&fclid=153f90da-576b-647d-004c