Citrus Plant Disease Detection Using Deep Learning

A

Mini Project Report

Submitted

by

Mr. Prathamesh M More	**2241008**
Mr. Viraj Jalindar Raut	**2241025**
Ms. Vaishnavi Yogesh Devkate	**2241003**
Ms. Sakshi Amol Hingane	**2241034**

In partial fulfillment for the requirement of Fourth Year, Deep Learning

of

Bachelor of Computer Engineering

Under the guidance of

Prof. Sushma Nandgaonkar Mam

(Assistant Professor)

Department of Computer Engineering

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology

Bhigwan Road, Vidyanagari

Baramati-413133

2022-23

Vidya Pratishthan's

Kamalnayan Bajaj Institute of Engineering and Technology, Baramati

Department of Computer Engineering

Certificate

This is to certify that following students

Mr. Prathamesh Madhukar More	**2241008**
Mr. Viraj Jalindar Raut	**2241025**
Ms. Vaishnavi Yogesh Devkate	**2241003**
Ms. Sakshi Amol Hingane	**2241034**

have successfully completed their Mini Project work on

CITRUS PLANT DISEASE DETECTION USING DL

during the academic year **2022-23** in the partial fulfillment towards the completion of **Deep Learning** in **Computer Engineering.**

Mini Project Guide
(Prof. Sushma Nandgaonkar)

HoD Deptt. of Comp. Engg.
(Dr. C. S. Kulkarni)

Principal
(Dr. R. S. Bichkar)

Acknowledgments

The opportunity of discussing such informative things related to Deep Learning and such domains was a very interesting and quite useful task for us. It was a very great experience to come up with new ideas and technologies with respect to our vast growing society and the technologies too.

I would like to thank all the Senior faculties that they thought of growing students Practical Knowledge and for explaining the importance of knowing the practical issues and the thinking of way to solve them by conducting such discussions. I would like to thank Our Hon. Principle Dr. R.S. Bichkar Sir that he was always available for the students with his motivation and proper guidance.

Also, I would like to thank Our Guide Prof. Sushma Nandgaonkar Mam who personally looked over us regarding this activity, her time to time guidance was very much valuable for us that cant be mentioned in words. The suggestions given by them were much precious for the successful execution of the Mini Project. Also, they motivated us to work in more smarter ways.

At last, I would like to thank every person who were Directly or Indirectly involved in the Successful Execution of this work on time.

Mr. Prathamesh Madhukar More
Mr. Viraj Jalindar Raut
Ms. Vaishnavi Yogesh Devkate
Ms. Sakshi Amol Hingane

Abstract

Nowadays, with the development in machine learning and deep learning, it is possible to not only identify but control plant diseases by using computer devices instead of manual inspection. So that it helped in improving the quality as well as production of fruits. Citrus fruits are well known for their taste and nutritional values. They are one of the natural and well-known sources of vitamin C, so planted worldwide. Manual inspection of fruit diseases with the naked eye takes time and also it is difficult. Therefore, a computer-based method is always required for accurate recognition of plant diseases. So build a citrus image dataset including all common citrus diseases.

The deep learning network is used to train and learn these images, Which can effectively identify and classify plant diseases. The CNN model is used as the primary network. The results show that techniques of data augmentation and preprocessing have delivered promising insights to estimate citrus fruit's damages in percentage and also identify and classify agricultural diseases with relevant suggestions through which can avoid citrus plant disease with specific insecticides,Pesticides and fertilizers.

Contents

List of Figures

1

Introduction

There are many types of agricultural calamities with great impact and frequent disasters, which are not only cause losses to crop production, but also threaten food safety. Crop disease is one of the main types of "disasters" which affect on the citrus plant. Citrus disease is one of the most important trading crops all over the word. It brings great economic losses to farmers. There are some diseases that are the such as Greening, Canker, Scab, Black spot and Melanose. It takes selects several common citrus diseases as experimental objects disease problems certainty affect crops throughout their growing cycle. In agriculture field, disease detection is important step.

If disease is not detected at early stage will lead to farmer's economical loss. Machine vision framework uses image processing techniques to perform such specific work, that is why image processing assumes an exceptionally significant job in their capabilities. The system will take input as images of citrus plant and gives output as detected diseases for which plant is infected and provide the recommendation by proving required measurements of fertilizers or pesticides through CNN and classification algorithms. So system is proposed to classify the diseases using Convolution Neural Network.

1.1 Overview

Deep Learning is an Sub part of Artificial Intelligence which deals which multiple computer operated applications in which a machine works with a Multiple Neurons and Hidden layers which helps us in better Classifications and Predictions. It is used to reduce the Errors using Weight Biases.

1.2 Motivation

The main motivation behind this project is that it Proposed CNN-based citrus plant disease identification model is capable of distinguishing between healthy and diseased citrus fruits and leaves.

We used the CNN model to tackle the problem of classifying diseases from citrus fruit and leaf images in this study.

The modules in our proposed model are as follows :
i) Data loading,
ii) Data Preprocessing,
iii) building CNN model.

Collects high-level attributes,yielding disease classification of citrus fruit/leaves into Black spot,Canker,Scab,Greening which are not human eye can't seen. It also not only detect the disease but also predict the required fertilizer's by which it prevent the disease the increase the citrus yielding.

2

Proposed Work

2.1 Problem Definition

In Agriculture field, Disease detection is important step. If disease is not detected at early stage will lead to farmer's economical loss. The System will take input as images of citrus plant and will give output as detected diseases through CNN and classification algorithms. So System is proposed to classify the Diseases.

2.2 Scope of Mini Project

- The proposed system is able to detect the different citrus plant diseases such as Black spot, canker, scab, greening, and Melanose.
- Citrus plant disease classification has practical applications in the field of plant disease recognition based on their visual symptoms.
- The system will be able to detect diseases which are at early stage of the disease.
- The proposed method contains five stages such as preproccessing, segmentation, feature extraction, reduction, fusion, and classification.

2.3 Objective of the Mini Project

- To develop system to detect plant disease.
- To create database of Plant Diseases for respective Image.

2.4 Project Constraints

The system has some limitations which are :-

- It detects only disease but it is unable to give the exact damage i.e. how much plant is infected with the disease.
- System is unable to show the position of the damage.
- We only used one deep learning-based CNN model in this system other deep learning models can be also used.

3

System Architecture

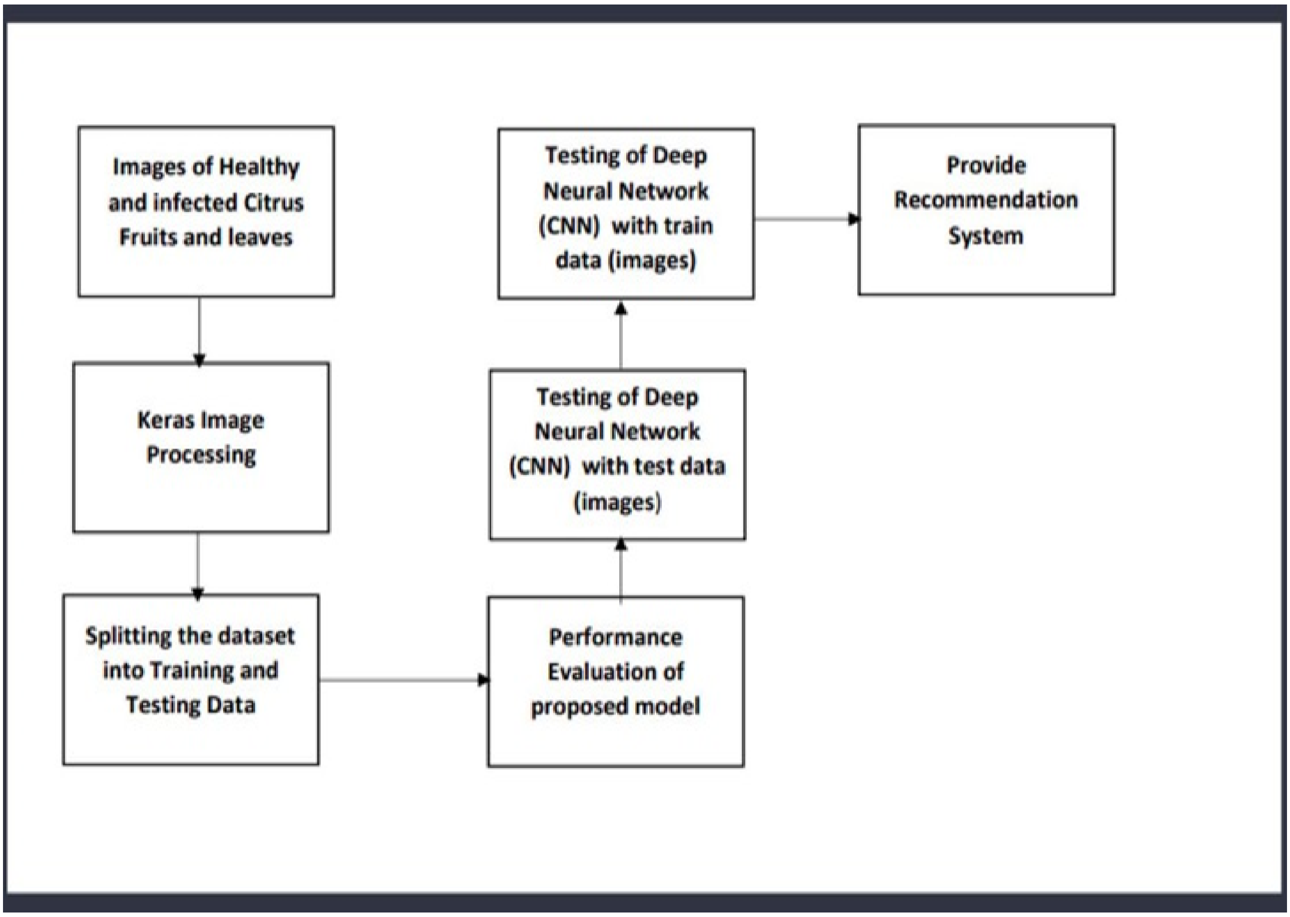


Figure 3.1: System Architecture

Firstly system will take input images of the citrus plant which contains leaves which are healthy as well as affected by the diseases. After that it will be given to the next step and data is downloaded from the keras. After that data is splitted into training data, testing data and validated data is preprocessed i.e. resized and rescaled.
Also data is augmented. Input layers are defined and hidden also defined. As output needed for data is data classified into five different classes.

So, five neurons are passed at the output layer. Here, CNN is used which contains five convolutional with simultaneous max-pooling layers. Model is build with the data as well as performance also evaluated. Different images were tested with random data of citrus plant and output is checked. Lastly disease is classified and specific fertilizer is recommended by the recommendation system.

4

Project Requirements Specification

4.1 Performance Requirements

For the Best Performance of the Software, user must follow the sequence of the activities to achieve the required results. Do not proceed to recognise text before the picture is captured. While using the software, users action must be consistent and unique. Input to the software must be in Proper Format.

4.2 Software Quality / Attribute Requirements

Some of the quality of attributes identified include:

- **Portability :** In API portability can be defined as compatibility of application with platform upgraded or downloaded versions.
- **Flexibility :** The architecture of the application will be flexible enough for some later requirements change or application enhancement.
- **Maintainability :** Whenever there is a change in requirement or bug found the application will be easily maintainable.

4.3 Safety Requirements

Safety requirements specify the needs that can help a system to keep intact after any problem takes place. The requirements are as follows.

- A backup of the database will be available on AWS.
- They should get accurate image otherwise it will anticipate the wrong disease and recommend the wrong fertilizer.

4.4 Security Requirements

Security requirements are needed to prevent any malicious attack that can take place on the project. These requirements are as follows:-

- The Information need be secured from ethical hackers.
- The type of data is produced at the website should not be exposed to fertilizers company, so that they could not manipulate the market.

4.5 Hardware Requirements

- **System Processor :** Core i5 processor 10th generation 8 GB RAM
- **SSD :** 512 GB

4.6 Software Requirements

- **Operating system :** Windows 10 (Intel/AMD 64-bit)
- **Coding Language :** Python (version 3.7.0)
- **Libraries :** Keras, Tensorflow, Matplotlib, Seaborn, Numpy, Pandas
- **IDE :** Jupyter Notebook, Anaconda, Google Colab

5

Tools Used

5.1 Overleaf

Overleaf is free to use. You Can Create, Edit and Share your projects with a sign up method. Overleaf is a real time editor for used to reasearch paper and projects. Oveleaf is a cloud based LaTeX editor used for writing, editing and publishing scientific documents. Overleaf can be a access multiple user at a time.

5.2 Anaconda Navigator

Anaconda is an open source tool. Anaconda Navigator also includes a graphical user interface. It can be used for python and R programming language for Data Science that aims to simplify package management and deployment. Anaconda Navigator can launch any applications and manage Anaconda package without using command line interface.

5.3 Jupyter Notebook

Jupyter Notebook is a widely based application for a creating and sharing any documents. Jupyter notebook is mostly used in Python Programming language related project. Jupyter Notebook can be support programming language such as R and python.

5.4 Diagrams.Net

Diagrams net is free online diagram software. It can be used for making flowchart, process diagram, DFD diagram, UML diagram and network diagram.

6

Project Stuctures

6.1 Project Breakdown Structure

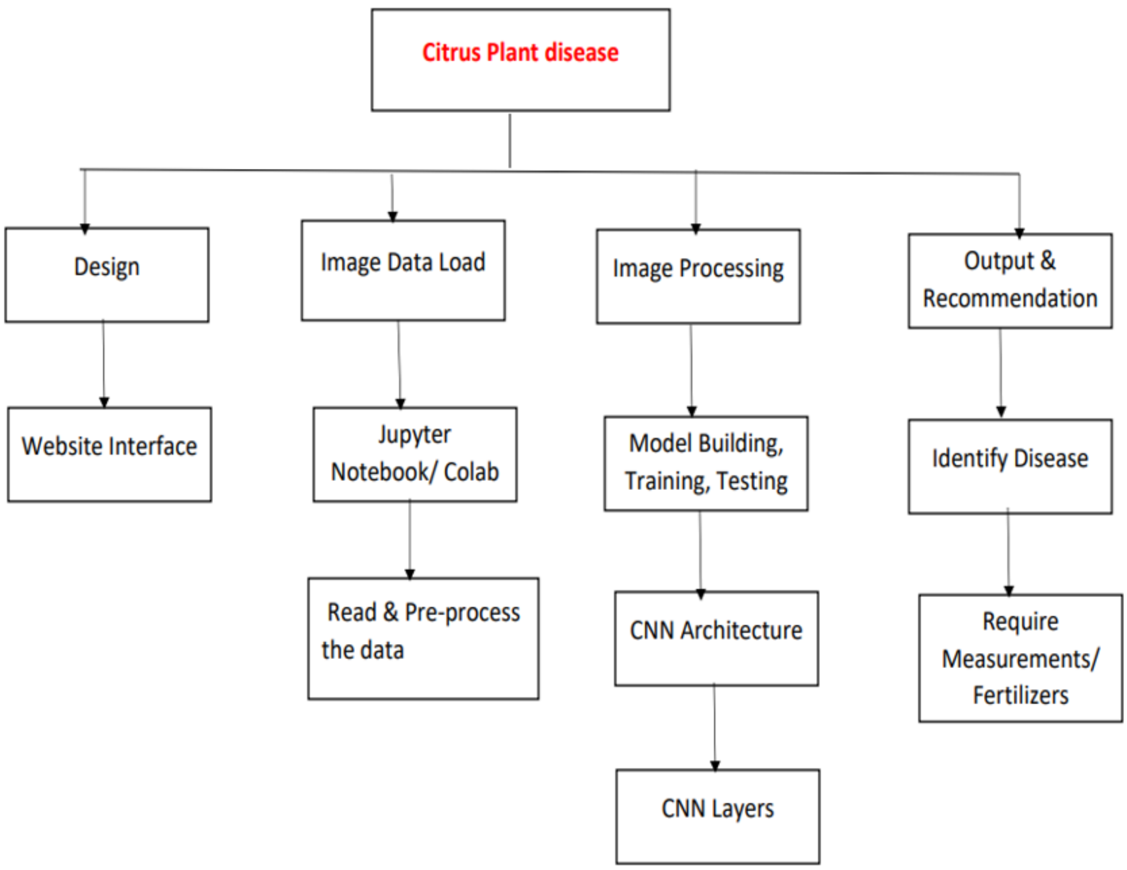


Figure 6.1: Project Breakdown

6.2 Task Network

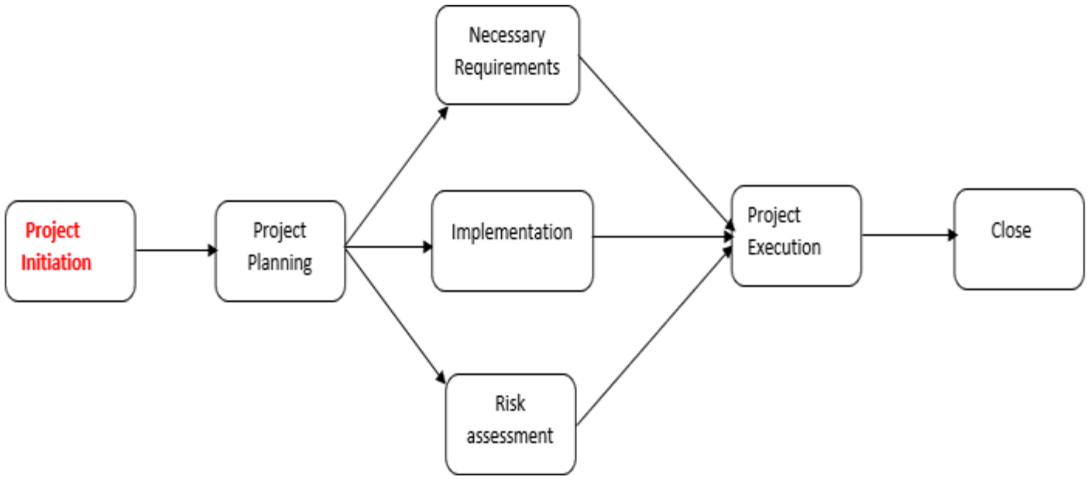


Figure 6.2: Task Network

6.3 Use Case Diagram

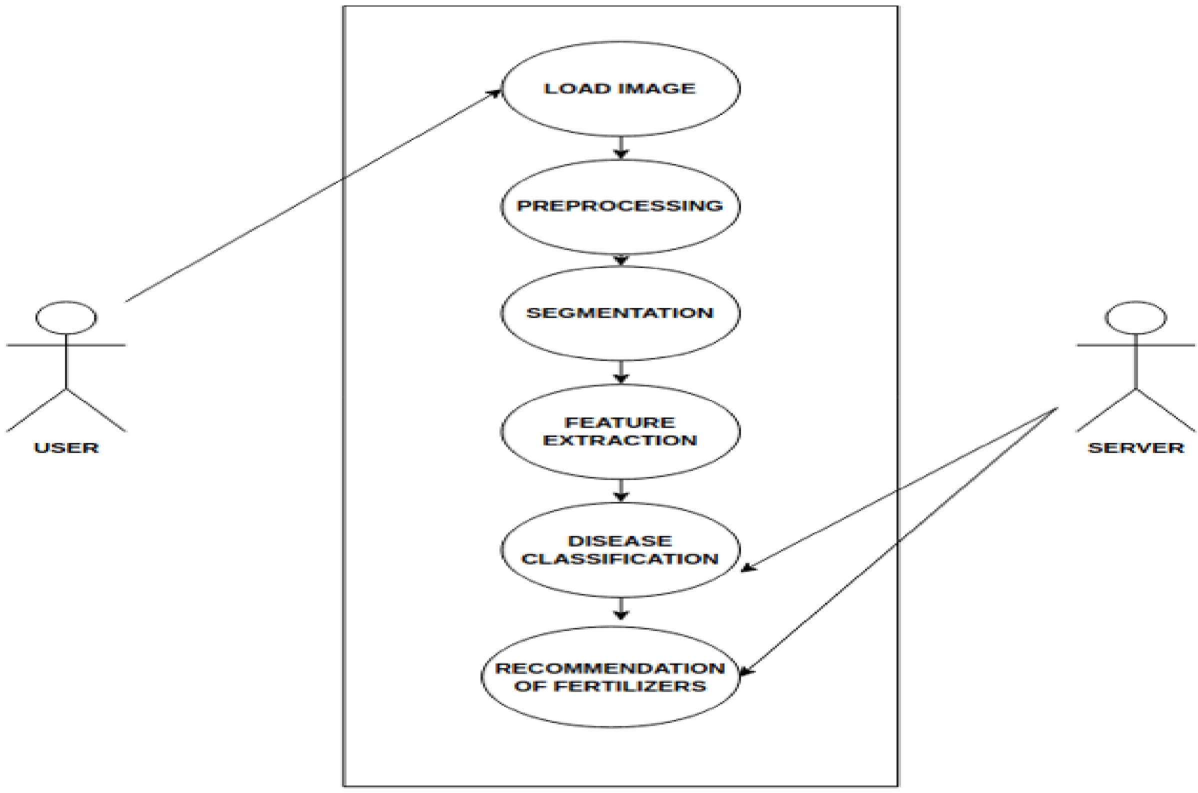


Figure 6.3: Use Case Diagram

7 Implementation

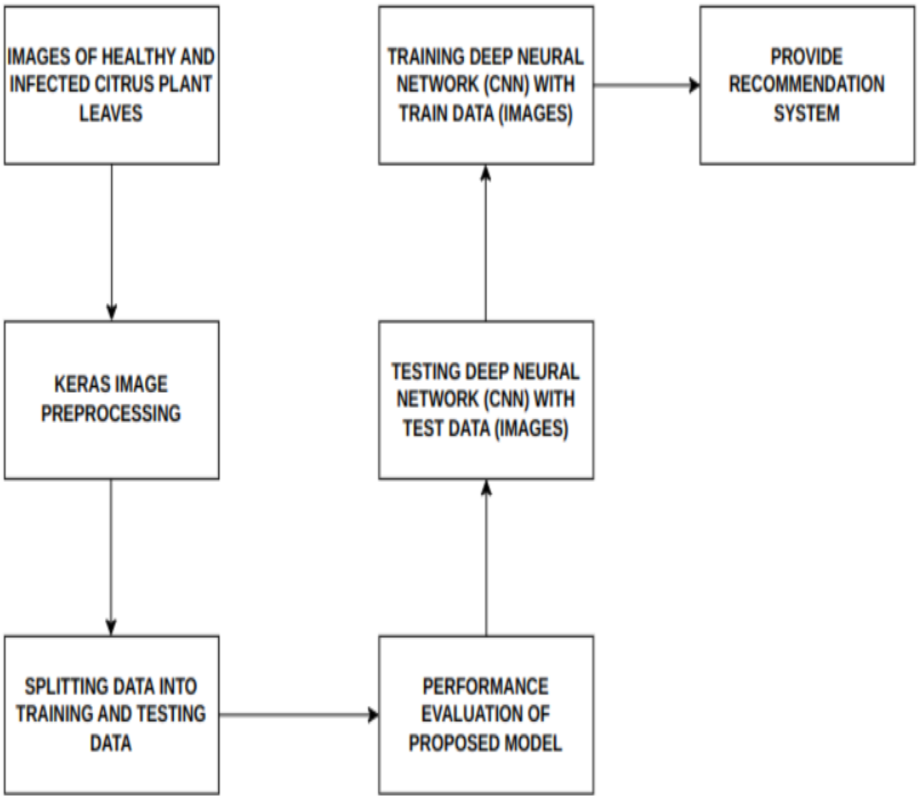


Figure 7.1: Layer wise process

Import Libraries

In [1]:
```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
```

1. CNN MODEL

UPLOADING DATA

In [2]:
```
dataset = tf.keras.preprocessing.image_dataset_from_directory("Leaves",shuffle =True,image_size = (255, 255), batch_size = 32)
```

```
Found 1036 files belonging to 5 classes.
```

CHECKING CLASSES PRESENT IN DATASET

In [3]:
```
len(dataset)
```

Out[3]:
```
33
```

In [4]:
```
class_name=dataset.class_names
class_name
```

Out[4]:
```
['Black spot', 'Melanose', 'canker', 'greening', 'healthy']
```

Figure 7.2: Initialization

PLOTTING THE IMAGES

In [6]:
```
plt.figure(figsize=(15,15))
for image_batch, label_batch in dataset.take(1):
  for i in range(30):
    ax = plt.subplot(6,5,i+1)
    plt.imshow(image_batch[i].numpy().astype("uint8"))
    plt.title(class_name[label_batch[i]])
    plt.axis("OFF")
```

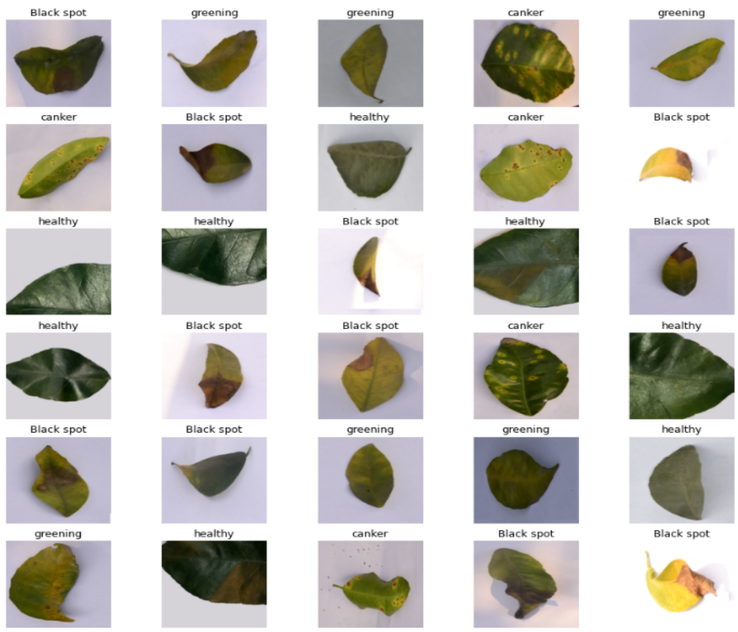


Figure 7.3: Images of the dataset

MODEL BUILDING

```
In [12]: resize_and_rescale = tf.keras.Sequential([
           layers.experimental.preprocessing.Resizing(255, 255),
           layers.experimental.preprocessing.Rescaling(1./255),
         ])
```

```
In [13]: data_augmentation = tf.keras.Sequential([
           layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
           layers.experimental.preprocessing.RandomRotation(0.2),
         ])
```

```
In [14]: input_shape = (32, 255, 255, 3)
         model = models.Sequential([
             resize_and_rescale,
             layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
             layers.MaxPooling2D((2, 2)),
             layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
             layers.MaxPooling2D((2, 2)),
             layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
             layers.MaxPooling2D((2, 2)),
             layers.Conv2D(64, (3, 3), activation='relu'),
             layers.MaxPooling2D((2, 2)),
             layers.Conv2D(64, (3, 3), activation='relu'),
             layers.MaxPooling2D((2, 2)),
             layers.Conv2D(64, (3, 3), activation='relu'),
             layers.MaxPooling2D((2, 2)),
             layers.Flatten(),
             layers.Dense(64, activation='relu'),
             layers.Dense(5, activation='softmax'),
         ])
```

Figure 7.4: Building Model

```
In [16]: model.summary()

Model: "sequential_2"
click to expand output; double click to hide output         Param #
=================================================================
 sequential (Sequential)     (32, 255, 255, 3)         0

 conv2d (Conv2D)             (32, 253, 253, 32)        896

 max_pooling2d (MaxPooling2D  (32, 126, 126, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 124, 124, 64)        18496

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 64)         0
 2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPooling  (32, 30, 30, 64)         0
 2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 64)          36928

 max_pooling2d_4 (MaxPooling  (32, 6, 6, 64)           0
 2D)

 conv2d_5 (Conv2D)           (32, 4, 4, 64)            36928

 max_pooling2d_5 (MaxPooling  (32, 2, 2, 64)           0
 2D)

 flatten (Flatten)           (32, 256)                 0

 dense (Dense)               (32, 64)                  16448

 dense_1 (Dense)             (32, 5)                   325

=================================================================
Total params: 183,877
Trainable params: 183,877
Non-trainable params: 0
_________________________________________________________________
```

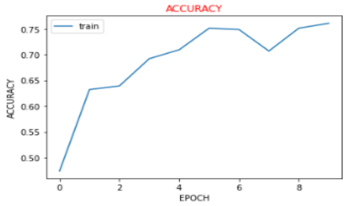
Figure 7.5: Summary of Model

MODEL EVALUATION

```
In [21]: scores = model.evaluate(test_ds)
```

```
4/4 [==============================] - 5s 461ms/step - loss: 0.4455 - accuracy: 0.8333
```

```
In [22]: import matplotlib.pyplot as plt
plt.plot(hist.history['accuracy'])
plt.title("ACCURACY",color="red")
plt.ylabel("ACCURACY")
plt.xlabel("EPOCH")
plt.legend(['train'],loc='upper left')
plt.show()
```




```
In [23]: plt.plot(hist.history['loss'])
plt.title("loss",color="red")
plt.ylabel("LOSS")
plt.xlabel("EPOCH")
plt.legend(['train'],loc='upper left')
plt.show()
```

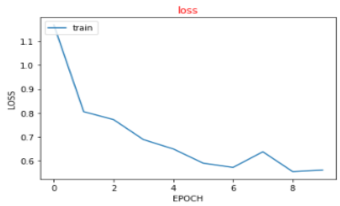


Figure 7.6: Model Evaluation

```
In [25]: plt.figure(figsize=(20, 20))
for images, labels in test_ds.take(1):
    for i in range(16):
        ax = plt.subplot(4, 4, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        predictions = model.predict(tf.expand_dims(images[i], 0))
        score = tf.nn.softmax(predictions[0])
        if(class_name[labels[i]]==class_name[np.argmax(score)]):
            plt.title("Actual: "+class_name[labels[i]])
            plt.ylabel("Predicted: "+class_name[np.argmax(score)],fontdict={'color':'green'})

        else:
            plt.title("Actual: "+class_name[labels[i]])
            plt.ylabel("Predicted: "+class_name[np.argmax(score)],fontdict={'color':'red'})
        plt.gca().axes.yaxis.set_ticklabels([])
        plt.gca().axes.xaxis.set_ticklabels([])
```

```
1/1 [==============================] - 0s 355ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 117ms/step
1/1 [==============================] - 0s 147ms/step
1/1 [==============================] - 0s 68ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 67ms/step
1/1 [==============================] - 0s 66ms/step
1/1 [==============================] - 0s 85ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 65ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 68ms/step
```

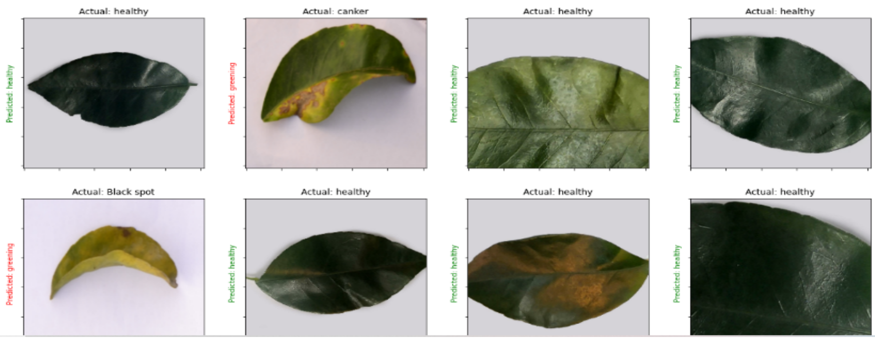


Figure 7.7: Prediction

8

Experimental Results

8.1 Disease Detection

There are some diseases that are the most dangerous, such as Black spot, canker, scab, greening and melanose classes. It selects several common citrus diseases as experimental objects Disease problems certainty affect crops throughout their growing cycle In agriculture field, disease detection is important step.If disease is not detected at early stage will lead to farmer's economical loss. Machine vision framework uses image processing techniques to perform such specific work, that is why image processing assumes an exceptionally significant job in their capabilities.

At First, model predict the disease due to which plant is infected and then by displaying not only disease name but also gives the recommendation that how to degrade this disease and which fertilizers, insecticides and pesticides that used to overcome this problem. It recommend the user that what they should be done.

Our Model works best for predicting Citrus disease detection with accuracy 83.33%

9

Conclusion

The proposed CNN-based leaf disease identification model is capable of differentiating between diseased and healthy Citrus leaves and fruits. In this study CNN model is used to tackle the problem of classifying disease from citrus leaf and fruit images. It contains modules as Data acquisition, Data preprocessing and CNN model application. In this first convolution layer low level features from the picture and second convolution layer collects high-level attributes, yielding disease classification of citrus leaves into Black spot, canker, scab, greening and melanose classes.

A

Glossary

A.1 List of Notations Used

- **CNN :** Convolutional Neural Network
- **DFD :** Data Flow Diagram
- **UML :** Unified Modelling Language
- **GUI :** Graphical User Interface
- **SSD :** Solid State Drive
- **HDD :** Hard Disk Drive
- **API :** Application Programming Interface
- **WOA :** Whale Optimization Algorithm