A

Major Project

On

RECOGNITION MODEL FOR CROP DISEASES AND INSECT PESTS BASED ON DEEP LEARNING IN HARSH ENVIRONMENT

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

P. NARESH

(167R1A05A3)

B. VINAY KUMAR

(157R1A0570)

A. ABISAINATH REDDY

(157R1A05J3)

Under the Guidance of

Mrs. U. KRISHNAVENI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2015-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**RECOGNITION MODEL FOR CROP DISEASES AND INSECT PESTS BASED ON DEEP LEARNING IN HARSH ENVIRONMENT**" being submitted by **P. Naresh, B. Vinay Kumar , A. Abisainath Reddy** bearing the **167R1A05A3, 157R1A0570, 157R1A05J3** roll number in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, during the year 2021-22. It is certified that they have completed the project satisfactorily.

INTERNAL GUIDE
Mrs. U. krishnaveni
Assistant Professor

DIRECTOR
Dr. A. Raji Reddy

Head of Department
Dr. Srujan Raju

EXTERNAL EXAMINER

Submitted for viva voce Examination held on ____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Mrs. U. Krishnaveni , Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. B P Deepak Kumar, Mr. J. Narasimha Rao** and **Ms. V. Sandya** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

P. Naresh (167R1A05A3)

B. Vinay Kumar (157R1A0570)

A. Abisainath Reddy (157R1A05J3)

TABLE OF CONTENTS

ABSTRACT

Agricultural diseases and insect pests are one of the most important factors that seriously threaten agricultural production. Early detection and identification of pests can effectively reduce the economic losses caused by pests. In this paper, convolution neural networks are used to automatically identify crop diseases. The data set comes from the public data set of the AI Challenger Competition in 2018,with 27 disease images of 10 crops. In this paper, the Inception-ResNet-v2 model is used for training. The cross-layer direct edge and multi-layer convolution in the residual network unit to the model. After the combined convolution operation is completed, it is activated by the connection into the ReLu function. The experimental results show that the overall recognition accuracy is 86.1% in this model, which verifies the effectiveness. After the training of this model, we designed and implemented the We chat applet of crop diseases and insect pests recognition. Then we carried out the actual test. The results show that the system can accurately identify crop diseases, and give the corresponding guidance.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

1. INTRODUCTION

As a superpower with more than 20% of the world's total population, China has been facing the problem of insufficient arable land resources. According to the survey data of the Ministry of Agriculture, the proportion of cultivated land in China is even less than 10% of China's land area. According to statistics, the mountainous area accounts for about two-thirds of the total land area in China, while the plain area accounts for only one-third. About one third of the country's agricultural population and arable land are in mountainous areas. This situation has resulted in the rela-tively poor production conditions of agriculture, forestry and animal husbandry in China. According to the statistics of the Food and Agriculture Organization of the United Nations, the per capita cultivated land area in China is less than half of the world average level, and shows a decreasing trend year by year. Once natural disasters cause agricultural production reduction, it will seriously affect the output of agricultural products and agricultural development. So how to develop agriculture stably, especially in the complex environment, is extremely important for China. Although with the development of science and technology, agricultural production is progressing. But due to various natural factors and non-natural factors, the yield of crops has not been greatly improved. Among the various factors, the largest proportion is the problem of crop diseases and insect pests. According to statistics, the area of crops affected by pests and diseases in China is as high as 280 million km2 every year, and the direct yield loss is at least 25 billion kg. In recent years, this problem is on the rise and seriously threatens the development of the planting industry. Timely diag- nosis and prevention of crop diseases has become particularly important. At present, agricultural workers often use books and network, contact local experts and use other methods to protect and manage crop diseases. But for various reasons, misjudgments and other problems often occur, resulting in agricultural production being deeply affected. At present, the research on crop diseases is mainly divided into two directions. The first one is the traditional physical method, which is mainly based on spectral detection to iden-tify different diseases. Different types of diseases and insect pests cause different leaf damage, which leads to different spectral absorption and reflection of leaves eroded by dis-eases and healthy crops. The other one is to use computer vision technology to identify images. That is to say, the char-acteristics of disease images are extracted by using computer related technology, and the recognition is carried out through the different characteristics of diseased plants and healthy plants. In recent years, the rapid development of artificial intelligence has made life more convenient, and AI has become a well-known technology. For example, AlphaGo defeated the world champion of Go. Siri and Alexa as voice assistants of Apple and Amazon are all applications of arti-ficial intelligence technology represented by deep learning in various fields. As the key research object of computer vision and artificial intelligence, image recognition has been greatly developed in recent years. In agricultural applications, the goal of image recognition is to identify and classify different types of pictures, and analyze the types of crops, disease types, severity and so on. Then we can formulate corresponding countermeasures to solve various problems in agricultural production in a timely and efficient manner. So as to further ensure and improve the yield of crops and help the better development of agriculture. With the rapid development of deep learning especially in image recognition, speech analysis, natural language processing and other fields, it shows the uniqueness and efficiency of deep learning. Compared with the traditional methods, deep learning is more efficient in the diagnosis of crop diseases in the field of agricultural production. The deep learning model can monitor, diagnose and prevent the growth of crops in time. Image recognition of crop diseases and insect pests can reduce the dependence on plant protection tech-nicians in agricultural production, so that farmers can solve the problem in time. Compared with artificial identification, the speed of intelligent network identification is much faster than that of manual detection. And the recognition accuracy is getting higher and higher in the continuous development. The establishment of a sound agricultural network and the combination of Internet and agricultural industry can not only solve the problems related to crop yield affected by diseases and insect pests, but also be conducive to the development of agricultural informatization. However, due to the rugged terrain of the mountain envi-ronment, the surrounding interference factors are greater. Therefore, image acquisition is more difficult than the general environment. In addition, the camera and network transmission needed for image recognition and processing will also have a certain impact. Therefore, it is more difficult to carry out intelligent recognition in mountainous areas. This paper tries to build the Internet of Things platform in the complex environment of mountainous areas, and carry out the research on the identification model of crop diseases and insect pests. The purpose of this model is to improve agricultural information, deal with the harm of pests and diseases to crops, and improve crop yield.

1.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining privacy. Input Design considered the following things:

- What data should be given as input?
- How should the data be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

1.2 OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system
- It is achieved by creating user-friendly screens for the data entry to handle large volumes of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize instant. Thus the objective of input design is to create an input layout that is easy to follow

1.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other systems through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use it easily and effectively. When analyzing computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create documents, reports, or other formats that contain information produced by the system.

2. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 OBJECTIVE

A complex Internet of Things environment of crop diseases and insect pests identification model is established. Through the deployment of sensors and cameras in a complex mountainous environment, the environmental information and image information of the scene are collected, and the basic database of crop pest identification is established. Through the deep learning network model, the image information is learned and recognized, which is used to identify and collect leaf images, and then identify pests and diseases.

2.2 EXISTING SYSTEM

Although with the development of science and technology, agricultural production is progressing. But due to various natural factors and non-natural factors, the yield of crops has not been greatly improved. Among the various factors, the largest proportion is the problem of crop diseases and insect pests. According to statistics, the area of crops affected by pests and diseases in China is as high as 280 million km2 every year, and the direct yield loss is at least 25 billion kg. In recent years, this problem is on the rise and seriously threatens the development of the planting industry. Timely diagnosis and prevention of crop diseases has become particularly important. At present, agricultural workers often use books and network, contact local experts and use other methods to protect and manage crop diseases. But for various reasons, misjudgments and other problems often occur, resulting in agricultural production is deeply affected.

.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- But for various reasons, misjudgments and other problems often occur, resulting in agricultural
- Agricultural production is deeply affected.

2.3 PROPOSED SYSTEM

In this project we are applying deep learning convolutional neural network (CNN) to predict crop disease and its pests to reduce economical loss in crop business. To build disease recognition model author is applying RESNET CNN model which consists of 3 parts

1) Feature Extraction: CNN composed of multiple layers and first layer defined for feature extraction and these features will be extracted from a given input image dataset or any other multidimensional dataset.
2) Feature Selection: Using this layer features will be selected by applying a layer called pooling or max polling.
3) Activation module: using this module RELU will be applied on input features to remove out unimportant features and hold only relevant important features
4) Flatten: This layer will be define to convert multidimensional input features into single dimensional input array
5) Dense: This layer can be used to connect one layer to another layer to receive input features from the previous layer to the new layer to further filter input features in the next layer to get the most important features from the dataset to have the best prediction result.

2.3.1 ADVANTAGES

- The experimental results show that the overall recognition accuracy is 86.1% in this model.
- The results show that the system can accurately identify crop diseases, and give the corresponding guidance.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- System : Intel I3
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 1GB.

2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating system : Windows 10
- Coding Language : Python
- Tool : Anaconda

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

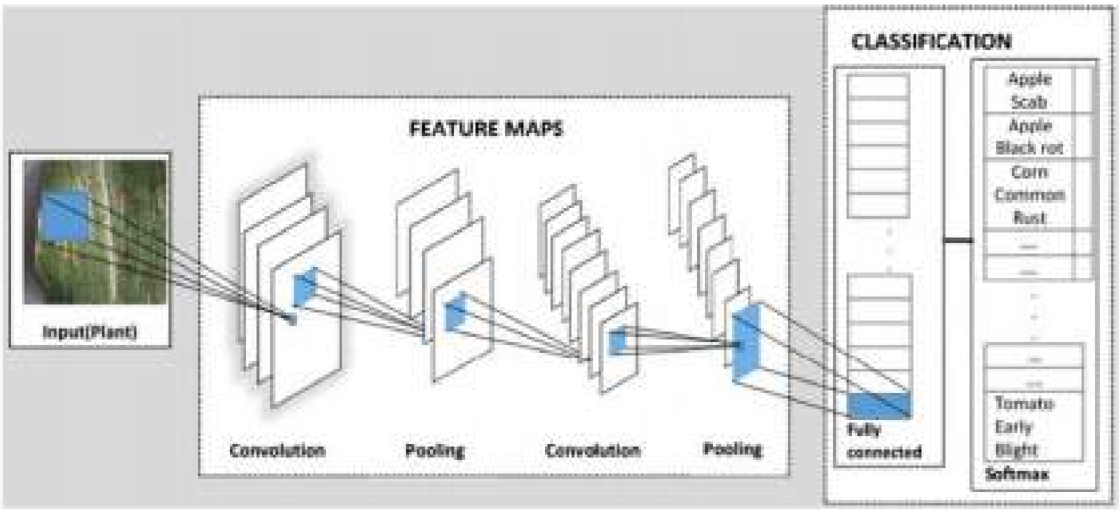


Fig 3.1 Project Architecture

3.2 DATA FLOW DIAGRAM

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

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

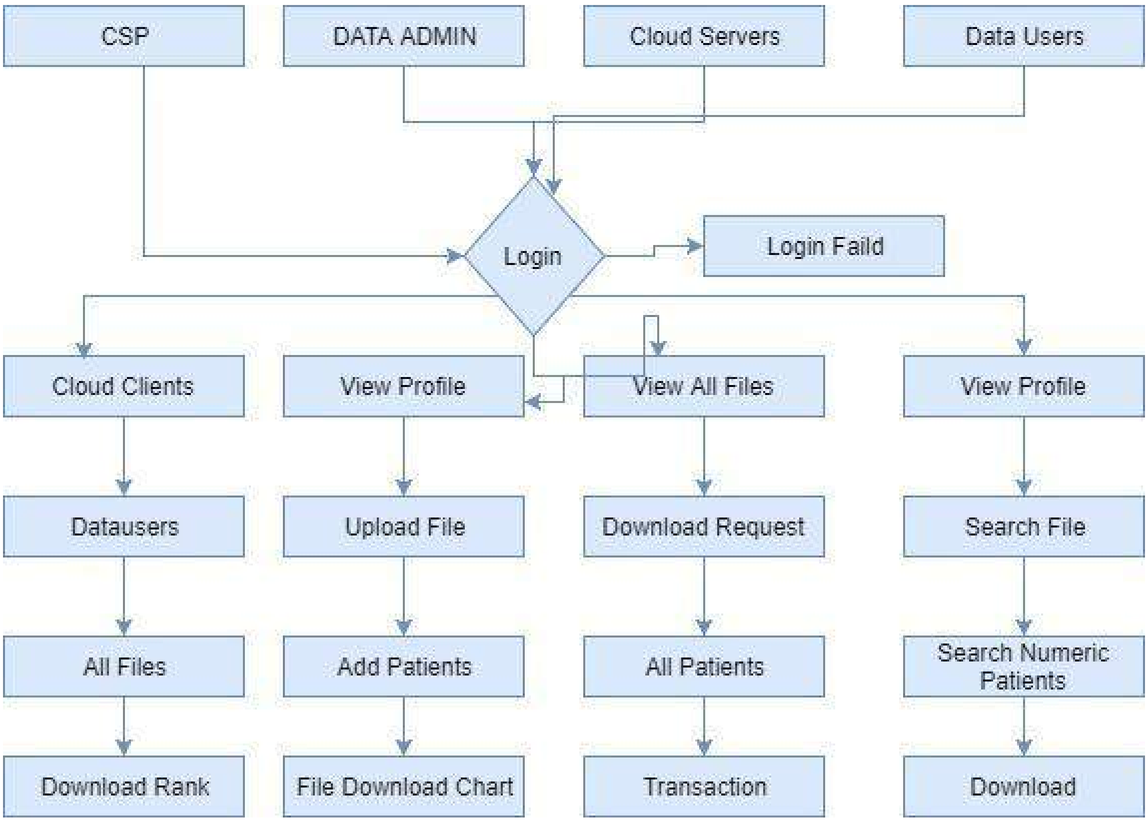


Fig 3.2 Data Flow Diagram

3.3 MODULES DESCRIPTION

Modules

- Feature Extraction
- Feature Selection
- Activation module
- Flatten
- Dense

3.3.1 FEATURE EXTRACTION

CNN consists of multiple layers and the first layer is defined for feature extraction and these features will be extracted from a given input image dataset or any other multidimensional dataset.A CNN model can be thought as a combination of two components: feature extraction part and the classification part. The convolution + pooling layers perform feature extraction. For example, given an image, the convolution layer detects features such as two eyes, long ears, four legs, a short tail and so on.

3.3.2 FEATURE SELECTION

Using this layer features will be selected by applying a layer called pooling or max polling.Feature selection is an important technique to improve neural network performances due to the redundant attributes and the massive amount in original data sets. ... A CNN is chosen for the image recognition task and one dropout is applied to reduce the overfitting of training data.

3.3.3 ACTIVATION MODULE

Using this module RELU will be applied on input features to remove unimportant features and hold only relevant important features The activation function is a node that is put at the end of or in between Neural Networks. They help to decide if the neuron would fire or not. We have different types of activation functions, but for this post, my focus will be on Rectified Linear Unit (ReLU).

3.3.4 FLATTEN

This layer will be defined to convert multidimensional input features into a single dimensional input array. Flatten is the function that converts the pooled feature map to a single column that is passed to the fully connected layer. Dense adds the fully connected layer to the neural network.

3.3.5 DENSE

This layer can be used to connect one layer to another layer to receive input features from the previous layer to the new layer to further filter input features in the next layer to get the most important features from the dataset to have the best prediction result.

.

3.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML consists of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

3.5 USE CASE DIAGRAMS

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

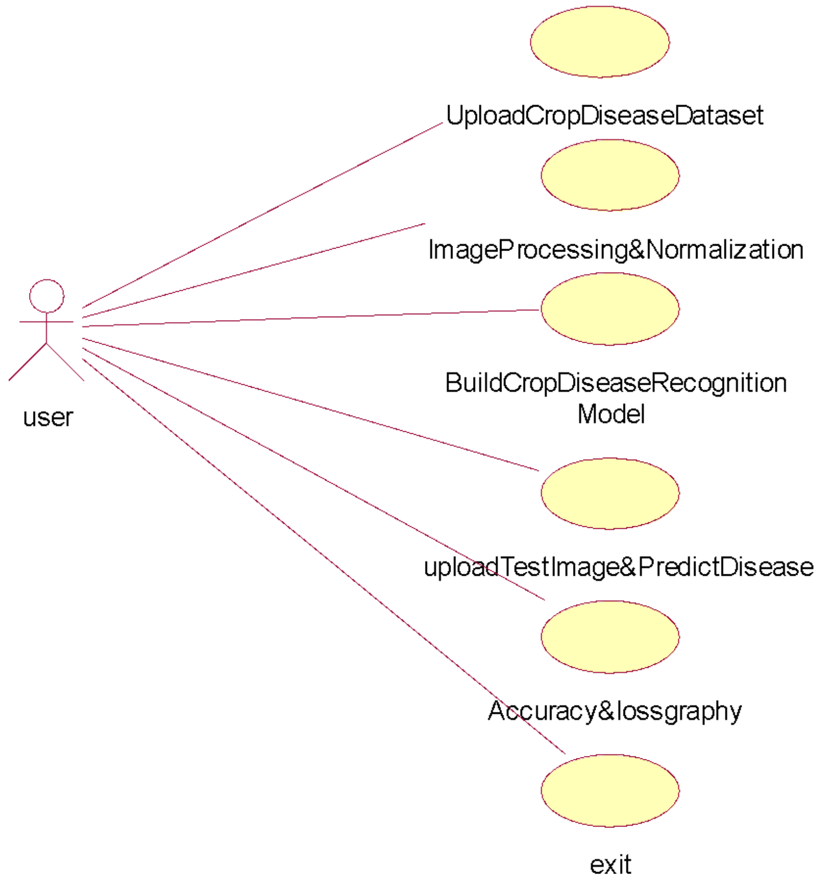


fig 3.5.1 use case diagram

3.6 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

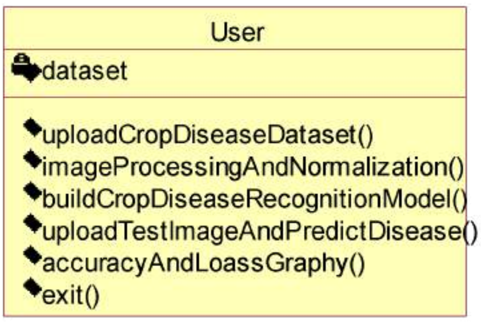


fig 3.6 Class Diagram

3.7 SEQUENCE DIAGRAM

fig 3.6 Sequence diagram

3.8 Collaboration

Collaboration is the process of two or more people, entities or organizations work ing together to complete a task or achieve a goal. Collaboration is similar to cooperation. Most collaboration requires leadership, although the form of leadership can be social within a decentralized and egalitarian group

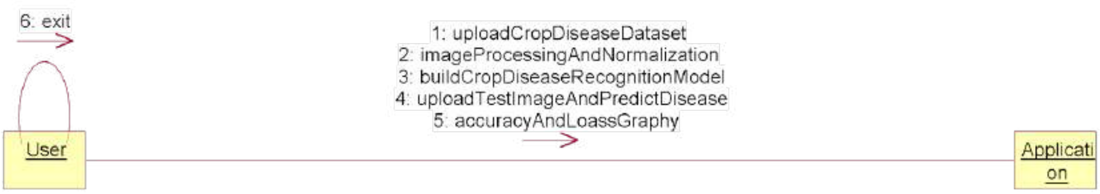


fig 3.8.1 Activity diagram for User

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from tkinter import filedialog
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.layers.core import Dense,Activation,Dropout, Flatten
from sklearn.metrics import accuracy_score
import os
import cv2
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
import pickle
from keras.models import model_from_json
main = Tk()
main.title("Research on Recognition Model of Crop Diseases and Insect Pests Ba
sed on Deep Learning in Harsh Environments")
main.geometry("1300x1200")
global filename
global X, Y
global model
global accuracy
plants = ['Pepper__bell___Bacterial_spot', 'Pepper__bell___healthy', 'Potato__
_Early_blight', 'Potato___healthy', 'Potato___Late_blight', 'Tomato__Target_Sp
ot',
'Tomato__Tomato_mosaic_virus', 'Tomato__Tomato_YellowLeaf__Curl_Viru
s', 'Tomato_Bacterial_spot', 'Tomato_Early_blight', 'Tomato_healthy',
'Tomato_Late_blight', 'Tomato_Leaf_Mold', 'Tomato_Septoria_leaf_spot
', 'Tomato_Spider_mites_Two_spotted_spider_mite']
def uploadDataset():
global X, Y
global filename
text.delete('1.0', END)
filename = filedialog.askdirectory(initialdir=".")
text.insert(END,'dataset loaded\n')
def imageProcessing():
text.delete('1.0', END)
```

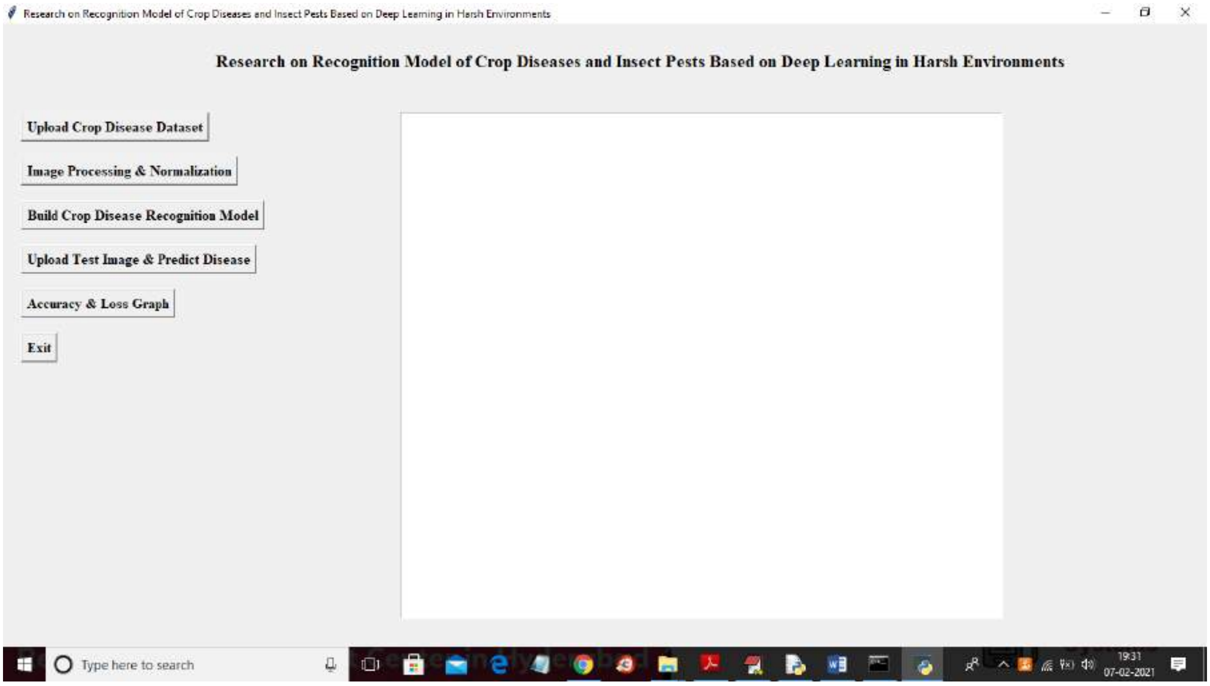
```
global X, Y
X = np.load("model/myimg_data.txt.npy")

Y = np.load("model/myimg_label.txt.npy")
Y = to_categorical(Y)
X = np.asarray(X)
Y = np.asarray(Y)
X = X.astype('float32')
X = X/255
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
text.insert(END,'image processing completed\n')
img = X[20].reshape(64,64,3)
cv2.imshow('ff',cv2.resize(img,(250,250)))
cv2.waitKey(0)
def cnnModel():
global model
global accuracy
text.delete('1.0', END)
if os.path.exists('model/model.json'):
with open('model/model.json', "r") as json_file:
loaded_model_json = json_file.read()
model = model_from_json(loaded_model_json)
json_file.close()
model.load_weights("model/model_weights.h5")
model._make_predict_function()
print(model.summary())
f = open('model/history.pckl', 'rb')
accuracy = pickle.load(f)
f.close()
acc = accuracy['accuracy']
acc = acc[9] * 100
text.insert(END,"CNN Crop Disease Recognition Model Prediction Accurac
y = "+str(acc))
else:
model = Sequential() #resnet transfer learning code here
model.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activatio
n = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Convolution2D(32, 3, 3, activation = 'relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Flatten())
model.add(Dense(output_dim = 256, activation = 'relu'))
model.add(Dense(output_dim = 15, activation = 'softmax'))
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', m
etrics = ['accuracy'])
```

```
print(model.summary())

hist = model.fit(X, Y, batch_size=16, epochs=10, validation_split=0.2,
shuffle=True, verbose=2)
model.save_weights('model/model_weights.h5')
model_json = model.to_json()
with open("model/model.json", "w") as json_file:
json_file.write(model_json)
json_file.close()
f = open('model/history.pckl', 'wb')
pickle.dump(hist.history, f)
f.close()
f = open('model/history.pckl', 'rb')
accuracy = pickle.load(f)
f.close()
acc = accuracy['accuracy']
acc = acc[9] * 100
text.insert(END,"CNN Crop Disease Recognition Model Prediction Accurac
y = "+str(acc))
def predict():
global model
filename = filedialog.askopenfilename(initialdir="testImages")
img = cv2.imread(filename)
img = cv2.resize(img, (64,64))
im2arr = np.array(img)
im2arr = im2arr.reshape(1,64,64,3)
test = np.asarray(im2arr)
test = test.astype('float32')
test = test/255
preds = model.predict(test)
predict = np.argmax(preds)
img = cv2.imread(filename)
img = cv2.resize(img, (800,400))
cv2.putText(img, 'Crop Disease Recognize as : '+plants[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 255, 0), 2)
cv2.imshow('Crop Disease Recognize as : '+plants[predict], img)
cv2.waitKey(0)
def graph():
acc = accuracy['accuracy']
loss = accuracy['loss']
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Iterations')
plt.ylabel('Accuracy/Loss')
plt.plot(acc, 'ro-', color = 'green')
plt.plot(loss, 'ro-', color = 'blue')
plt.legend(['Accuracy', 'Loss'], loc='upper left')
#plt.xticks(wordloss.index)
```

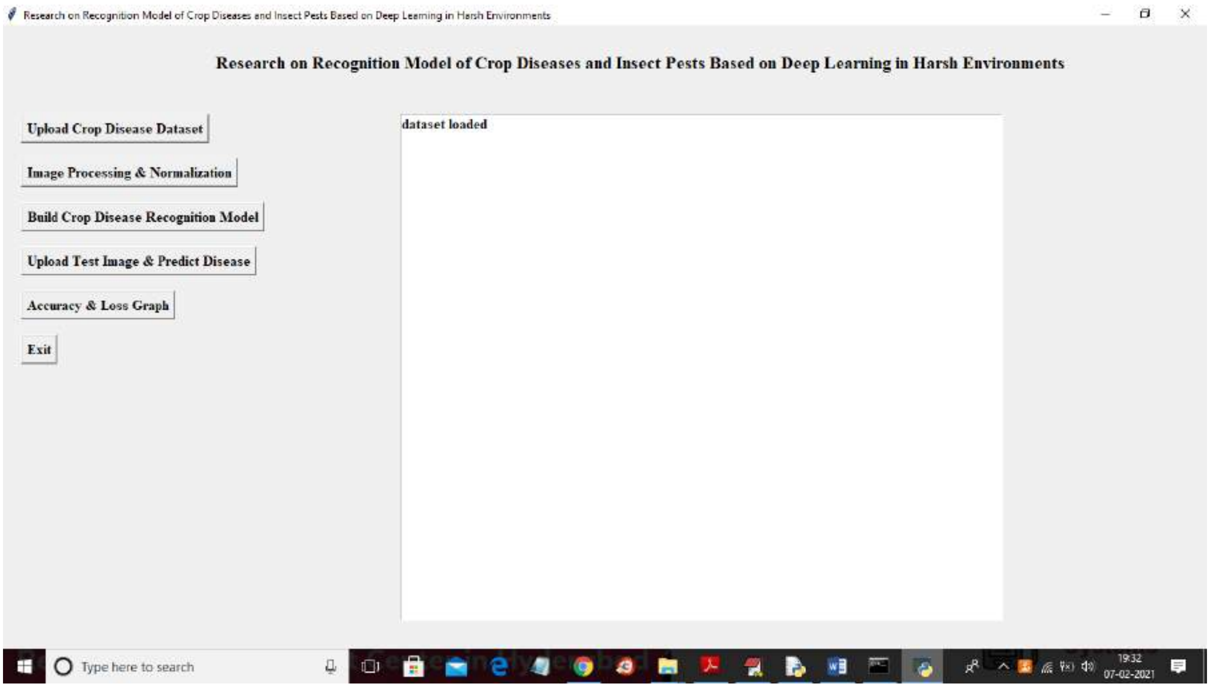
```
plt.title('Iteration Wise Accuracy & Loss Graph')
plt.show()
def close():
main.destroy()
text.delete('1.0', END)
font = ('times', 15, 'bold')
title = Label(main, text='Research on Recognition Model of Crop Diseases and I
nsect Pests Based on Deep Learning in Harsh Environments')
#title.config(bg='powder blue', fg='olive drab')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')
uploadButton = Button(main, text="Upload Crop Disease Dataset", command=upload
Dataset)
uploadButton.place(x=20,y=100)
uploadButton.config(font=ff)

processButton = Button(main, text="Image Processing & Normalization", command=
imageProcessing)
processButton.place(x=20,y=150)
processButton.config(font=ff)
modelButton = Button(main, text="Build Crop Disease Recognition Model", comman
d=cnnModel)
modelButton.place(x=20,y=200)
modelButton.config(font=ff)
predictButton = Button(main, text="Upload Test Image & Predict Disease", comma
nd=predict)
predictButton.place(x=20,y=250)
predictButton.config(font=ff)
graphButton = Button(main, text="Accuracy & Loss Graph", command=graph)
graphButton.place(x=20,y=300)
graphButton.config(font=ff)
exitButton = Button(main, text="Exit", command=close)
exitButton.place(x=20,y=350)
exitButton.config(font=ff)
font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=85)

scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=450,y=100)
text.config(font=font1)
main.config()
main.mainloop()
```

5. SCREENSHOTS

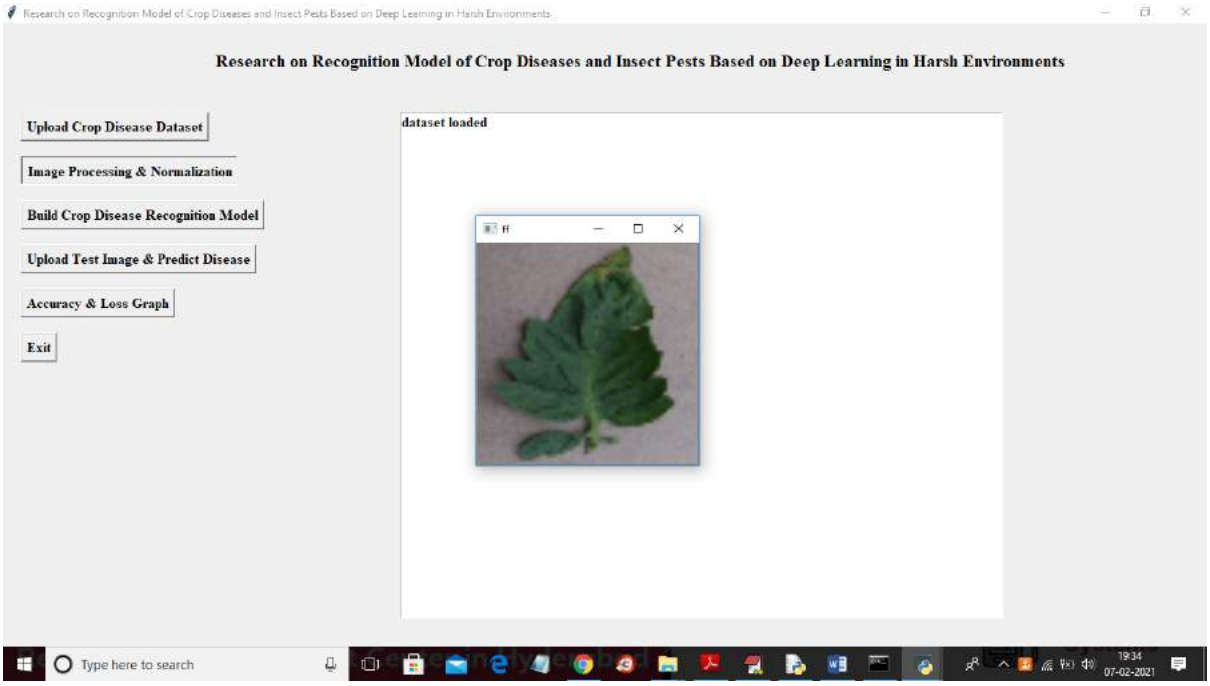
5.1 UPLOAD DATASET PAGE



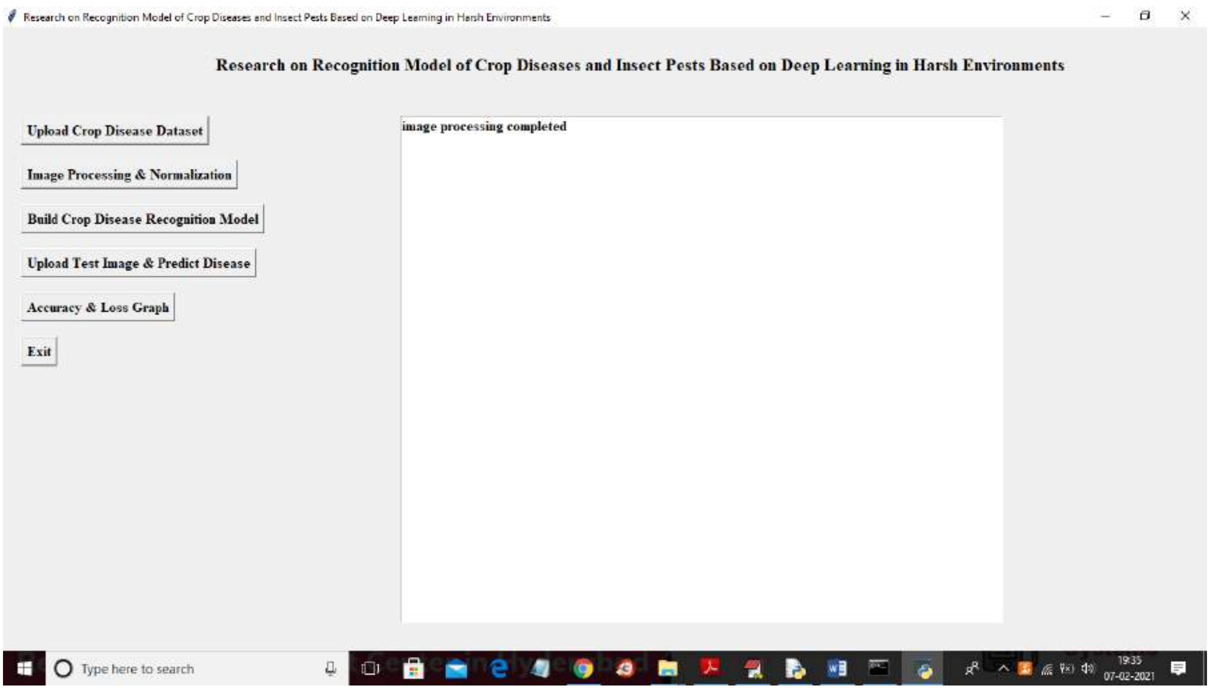
5.2 DATASET LOADED PAGE



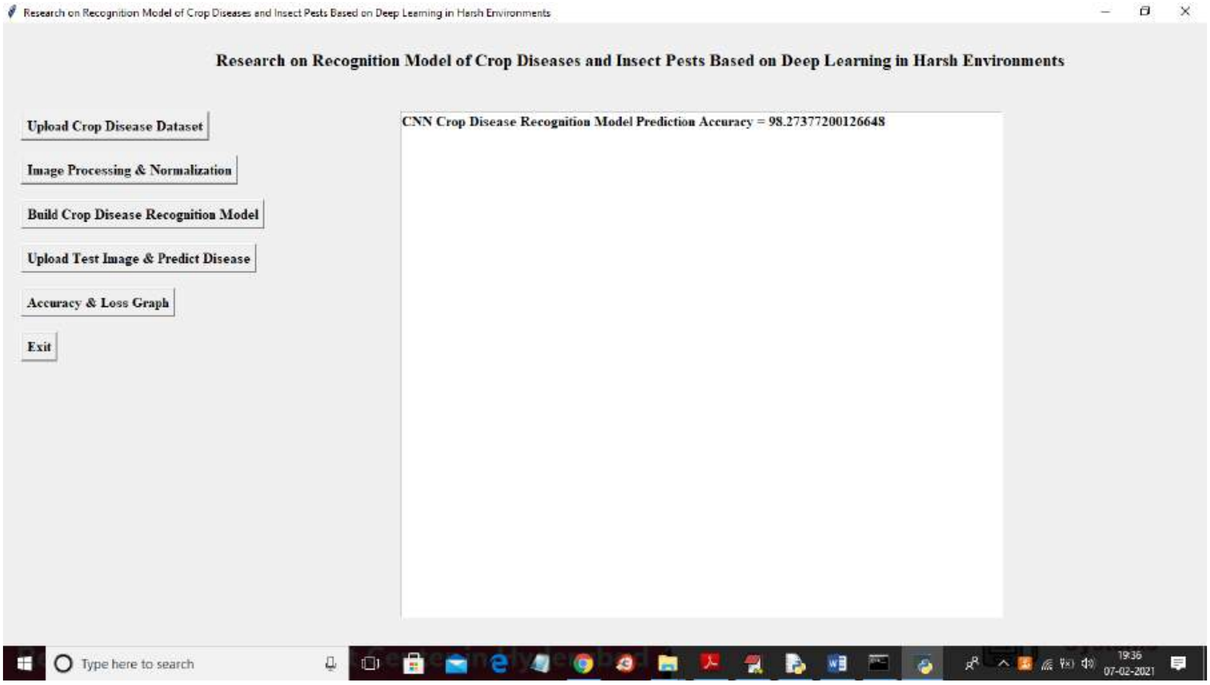
5.3 IMAGE CLASSIFICATION PAGE



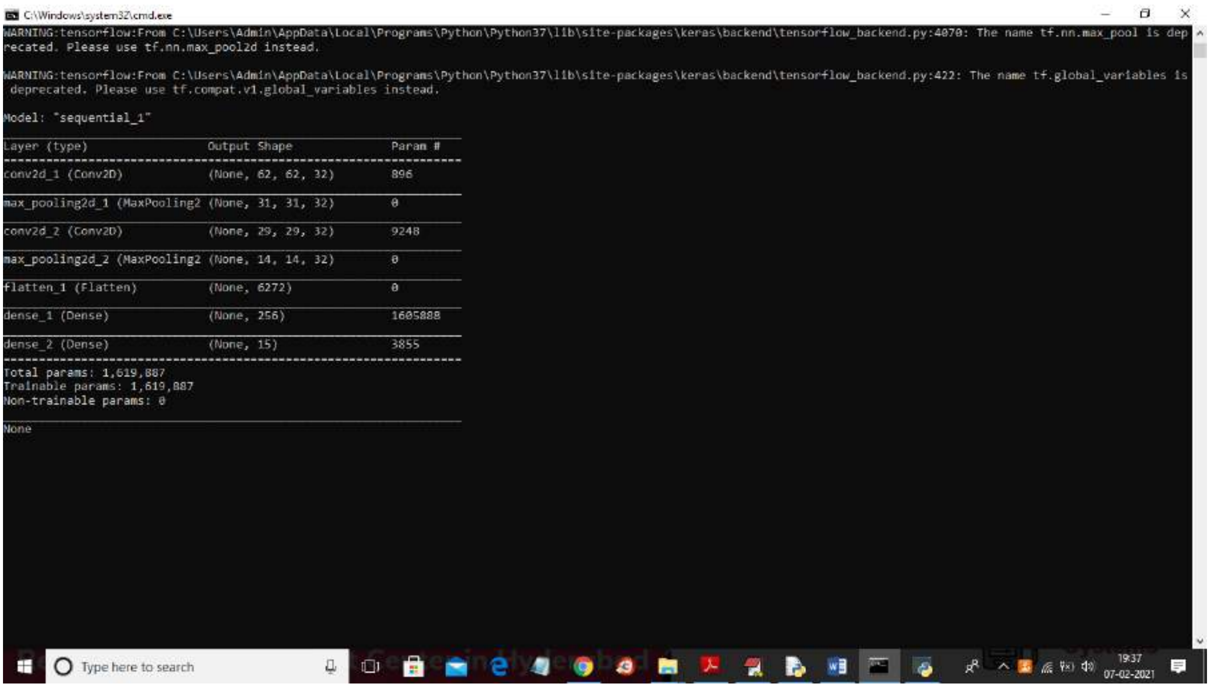
5.4 IMAGES PROCESSED PAGE



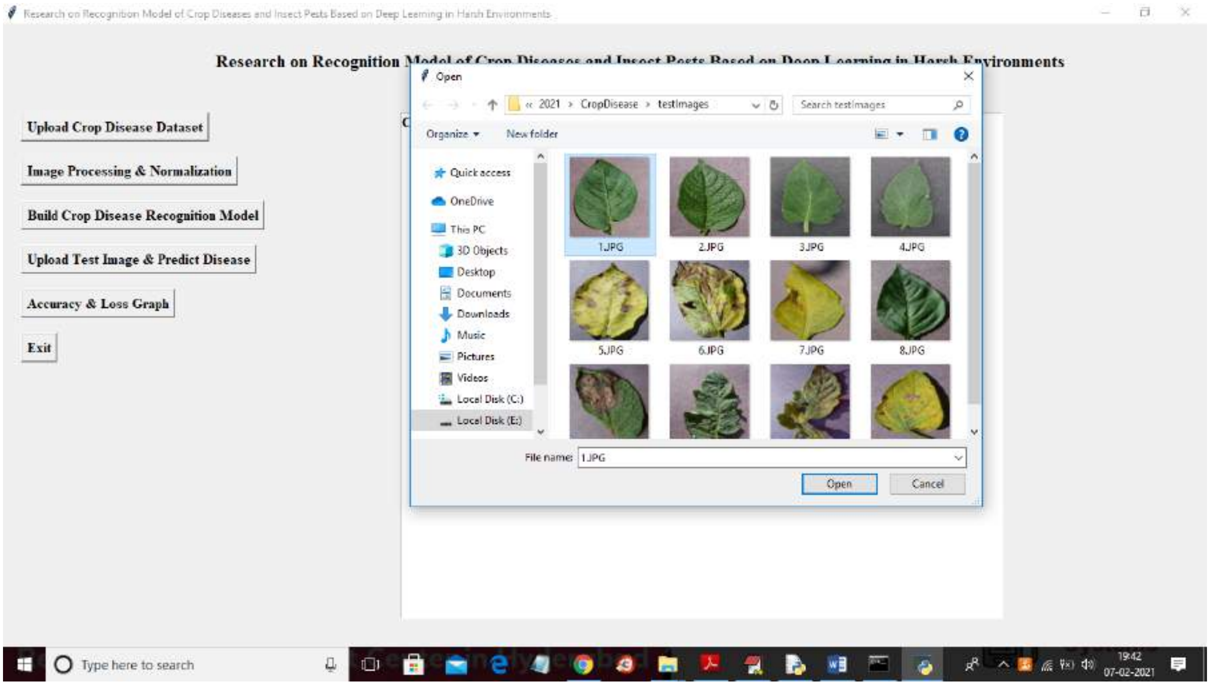
5.5 CNN MODEL GENERATION PAGE



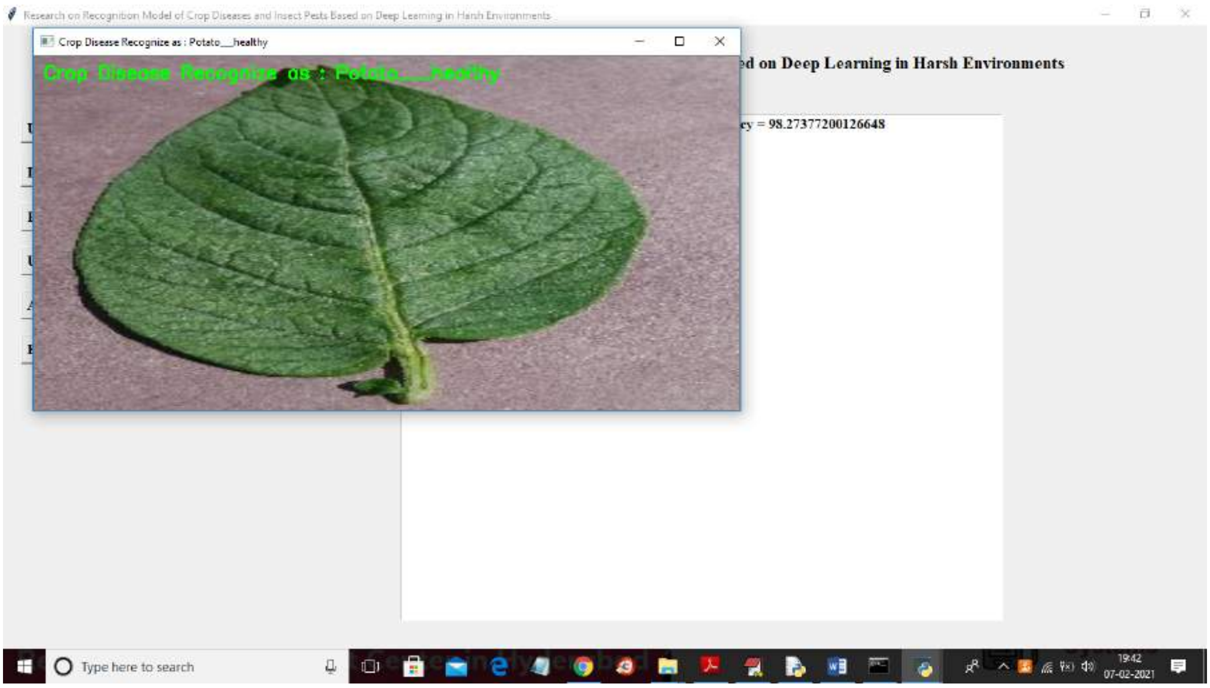
5.6 CONV2D, MAXPOOLING, FLATTEN and DENSE layer PAGE



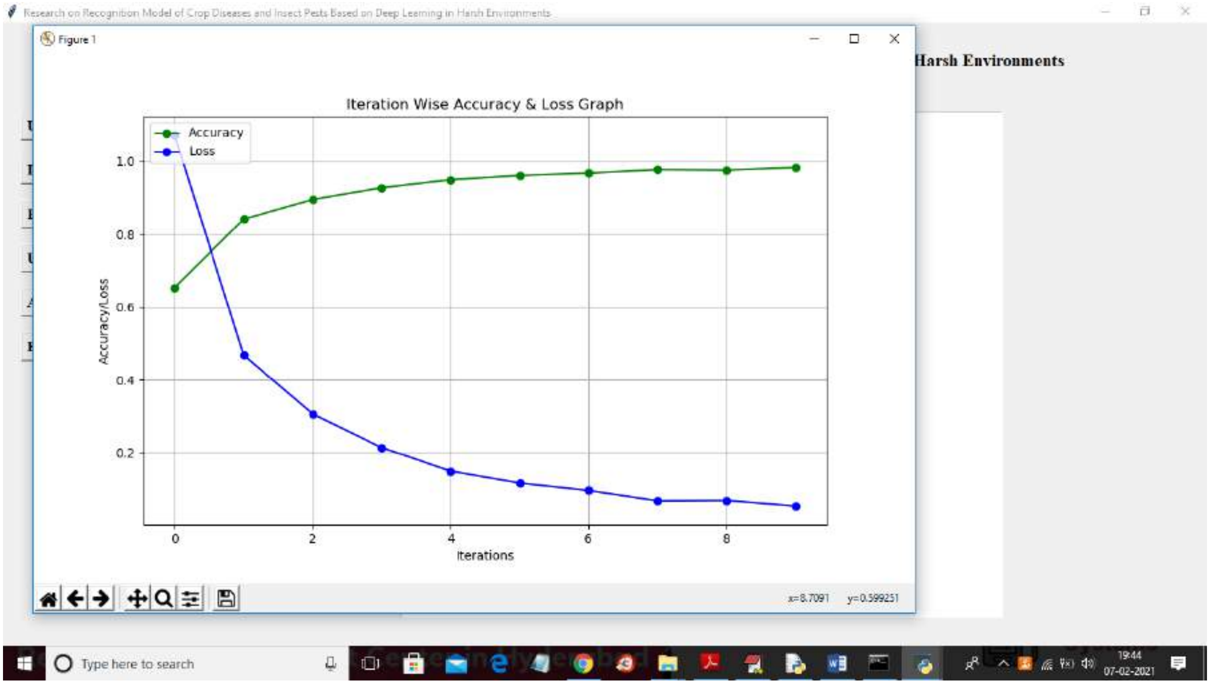
5.7 IMAGE UPLOAD PAGE



5.8 PREDICTION RESULT PAGE



5.9 ACCURACY & LOSS PAGE



6. TESTING

6.1 INTRODUCTION TO TESTING

Software testing is the process used to assess the quality of computer software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respective content of in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs. Testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and the behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (S Q A), which encompasses all business process areas, not just testing. Software testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding. Testing is the exposure of the system to trial input to see whether it produces correct output or not. Once the source code is generated, software must be tested to uncover as many errors as possible before delivery to the customer. So the main goal of testing is to design a series of test cases that have a high likelihood of finding errors.Except for small computer programs, systems should not be tested as a single monolithic unit. Large systems are built out of subsystems which are built out of modules, which are composed of procedures and functions. The testing process should therefore proceed in stages where testing is carried out incrementally in conjunction with system implementation.

SOFTWARE TESTING is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to Business process flows; data fields, predefined processes.

6.2.4 Module Testing

A module is a collection of dependent components such as an object or some looser collection of procedures and functions. A module encapsulates related components so that they can be tested without other system modules.

6.2.5 Sub-system Testing

This phase involves testing collections of modules which have been integrated into sub-systems. may be independently designed and implemented and most common problems which arise in large software systems or sub-system interface mismatches.

6.2.6 System Testing

Subsystems are integrated to make up the entire system. The testing process is concerned with finding errors which normally result from an unanticipated interface between sub-systems and components. It is also concerned with validating that the system meets its functional and non-functional requirements .

6.2.7 Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. It involves testing the system with the data applied by the system prowler rather than simulated data developed as part of the testing process. Acceptance testing of an reveals errors and emissions in the system requirements definition. The requirement may not reflect the actual facilities and performance required by the user and testing may demonstrate that the system doesn’t exhibit the anticipated performance and functionality **.**

6.3 TEST CASES

6.3.1 TABLE: DATASET UPLOAD

S. No	Input	Expected output	Results
1	upload dataset	dataset is uploaded	dataset is uploaded

6.3.2 TABLE: MODEL GENERATION

S. No	Input	Expected Output	Results
1	Build Crop Disease Recognition Model	Accuracy is shown	CNN model is generated

6.3.3 TABLE: PREDICTING THE DISEASE

S. No	Input	Expected Output	Results
1	Upload any dataset	Their disease should be predicted	Plant disease is predicted

7. CONCLUSION

In this project, 27 kinds of disease recognition of 10 kinds of crops were studied.The Inception-ResNet-v2 model is constructed by using deep learning theory and convolution neural network technology. Experiments show that the model can effectively identify the data set, and the overall recognition accuracy is as high as 86.1%. The results show that the recognition accuracy of this hybrid network model is relatively higher than the traditional model, and it can be effectively applied to the identification and detection of plant diseases and insect pests.

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] J. Schmidhuber, ‘‘Deep learning in neural networks: An overview,’’ Neural Netw., vol. 61, pp. 85–117, Jan. 2015.

[2] A.-R. Mohamed, G. E. Dahl, and G. Hinton, ‘‘Acoustic modeling using deep belief networks,’’ IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 1, pp. 14–22, Jan. 2012.

[3] A.-R. Mohamed, G. E. Dahl, and G. Hinton, ‘‘Acoustic modeling using deep belief networks,’’ IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 1, pp. 14–22, Jan. 2012.

[4] RA.-R. Mohamed, G. E. Dahl, and G. Hinton, ‘‘Acoustic modeling using deep belief networks,’’ IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 1, pp. 14–22, Jan. 2012.

.

[5] S. Datir and S. Wagh, ‘‘Monitoring and detection of agricultural disease using wireless sensor network,’’ Int. J. Comput. Appl., vol. 87, no. 4, pp. 1–5, Feb. 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012,pp. 1097–1105.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012,pp. 1097–1105.

[11] .A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012,pp. 1097–1105.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012,pp. 1097–1105.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classifica-tion with deep convolutional neural networks,’’ in Proc. Int. Conf. Neu-ral Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

.