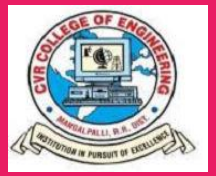


CVR COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Major Project

Green Leaf Detection using Raspberry pi

Batch No. 14

GUIDE :
Ms K.Manasa
Assistant Professor

Manasa Rajagopal Madabushi
18B81A05S2

Arakanti Maneesha
18B81A05S3

Liza Hans
18B81A05W2

Date Of Presentation : 30th April 2022

TABLE OF CONTENTS

ABSTRACT

Fig 1:leaf spots

- This project is about a framework utilizing raspberry PI to detect and prevent plant disease from spreading.
- In pharmaceutical research, the recognition of leaf ailment is essential and a critical theme for research, because it has the advantages of monitoring crops in the field in the form and thus automatically detects symptoms of disease by image processing .
- The term disease refers to the type of plant damage.
- This project aims to use the strategy to recognizing plant infections utilizing picture preparing and alarming the ailment brought about by email and SMS .
- Automatic detection of symptoms of disease is useful for upgrading agricultural products.
- Completely automatic design and implementation of these technologies will make a significant contribution to the chemical application. The cost of pesticides and other products will be reduced. This will lead to an increase in farm productivity.

MOTIVATION

Fig 2: Yellow Spot Disease

- India is cultivated country and about 70% of the population depends on agriculture. Farmers have large range of diversity for selecting various suitable crops and finding the suitable pesticides for plant.
- Disease on plant leads to the significant reduction in both the quality and quantity of agricultural product. The studies of plant disease refer to the studies of visually observable patterns on the plants. Monitoring of health and disease on plant plays an important role in successful cultivation of crops in the farm.
- In early days the monitoring and analysis of plant diseases were done manually by the expertise person in the field. This requires tremendous amount of work and also requires excessive processing time.
- The system uses raspberry pi to detect unhealthy leaves by training images.

PROBLEM STATEMENT

Develop an interface using Raspberry Pi using Green Leaf Detection the framework integrates the following components:

- Raspberry Pi 3
- Camera
- GSM Module
- Email Communication

PROPOSED METHODS

- Step 1: Connect Raspberry Pi and required hardware components later download VNC viewer for OS compatibility with Laptop and Raspberry Pi.
- Step 2: Later Installed Advanced IP Address Scan where we will be getting IP Address to access VNC Viewer.
- Step 3 : Here we scan IP Address in Advanced IP scanner and later enter IP Address in VNC Viewer then click on enter.
- Step 4 : We enter into Raspberry Pi where it runs Linux OS here we developed code in Thonny IDE by using python programming language.

PROPOSED METHODS

- Step 5: Afterwards run code for getting Plant Leaf Detection Application.
- Step 6 : Need to upload image of leaf with Disease
- Step 7 : Later it detects the Disease and gives this Disease information to Email id, SMS.

PROPOSED METHODS (CNN ALGORITHM)

Step 1: Input colour of the image IRGB of the leaf procure from the Plant detection dataset.

Step 2: Given input image IRGB, generate the mask Mveq (mask segmented) using CNN-based segmentation.

Step 3: Encrust IRGB with Mveq to get Mmask.

Step 4: Divide the image Mmask into smaller regions Ktiles (square tiles).

Step 5: Classify Ktiles from Mmask into Tomato and Corn Leaf images.

Step 6: Finally, Ktiles is the leaf part to detect disease.

Step 7: Stop.

FUNCTIONAL REQUIREMENTS

- Plant Leaf Disease Detection is an Application is used by user to detect the Diseased Leaf.
- It displays the name of the Disease.
- Plant Leaf Disease Detection Application is useful for Farmers in order to know which pesticide is useful for the crop.
- Based on this information farmers can decide which pesticide to be used for crop in order to make it disease free.
- Plant Leaf Detection Application has two ways to detect the Disease.
- First way to detect is by uploading the image from the Dataset.
- Second way to detect the Disease is by using Camera and detect the Disease.

NON FUNCTIONAL REQIREMENTS

- Plant Leaf Detection Application UI is developed by using Python Programming language in Thonny IDE.
- Dataset is taken from Kaggle.
- Here code is written for Camera to detect the Leaf with Disease.
- Another code is written for GSM Module to send Email and SMS to the user after detection.

LITERATURE REVIEW

Serial Number	Author	Technical Paper Title & Details	Description	Key learning
01	Yan Zhang , Shiyun Wa , Yutong Liu , Xiaoya Zhou, Pengshuo Sun and Qin Ma	High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module.	For the detection and prevention of disease of plants from getting spread, this paper discussed a system using raspberry PI.	➢ For the image analysis, the CNN algorithm was used. ➢ It has many advantages for the use in big farms of crops and thus it automatically detects signs of disease whenever they appear on leaves of the plant.
02	LAYA YESUDAS , SANTHIYA .S , PARIMALA .R , MOHAMM AD HARRIS	Leaf Disease Detection Using Raspberry Pi	Green plants are very much important to the human environment; they form the basis for the sustainability and long term health of environmental systems.	➢ The plant disease could be cured if it is known in the earlier stage. ➢ In this paper, they have proposed a system using raspberry pi to detect healthy and unhealthy plants & alerts the farmer by sending email.

LIMITATIONS OF EXISTING WORK

- The Capability of this framework is dependent on Configuration of Raspberry Pi.
- Internet connectivity is essential for the Email Communication is subject to quality of the camera.

SYSTEM SPECIFICATIONS/TECHNOLOGY

Hardware Requirements

Hardware	**Minimum Requirements**
Computer	4 GHz minimum, multicore-core processor
Raspberry Pi	Raspberry Pi 4 Model-B with 8GB RAM
GSM Module	SIM900A Modem is built with Dual Band GSM
USB Cable	Power/Charging, Data Transfer, Connector-A: 1 x USB (Type-A) Connector-B: 1 x USB (Type-C)
Web Camera	Quantum 720p Web Camera

Raspberry pi

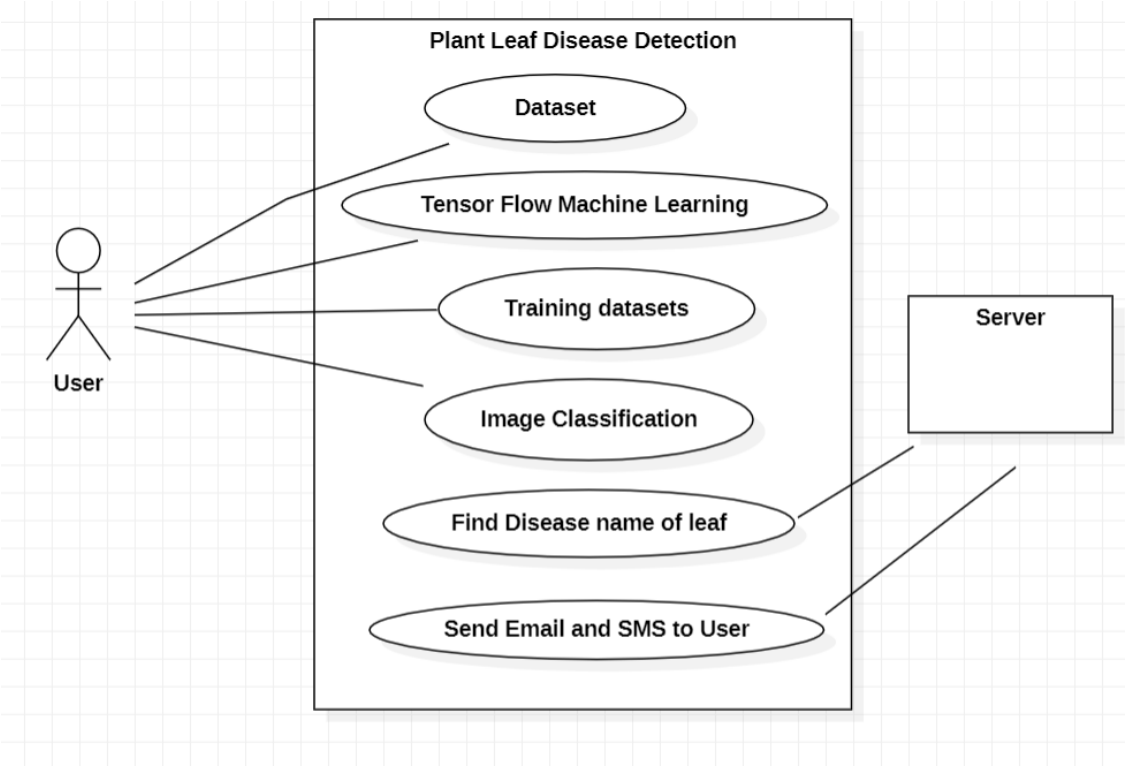
GSM Module

SYSTEM SPECIFICATIONS/TECHNOLOGY

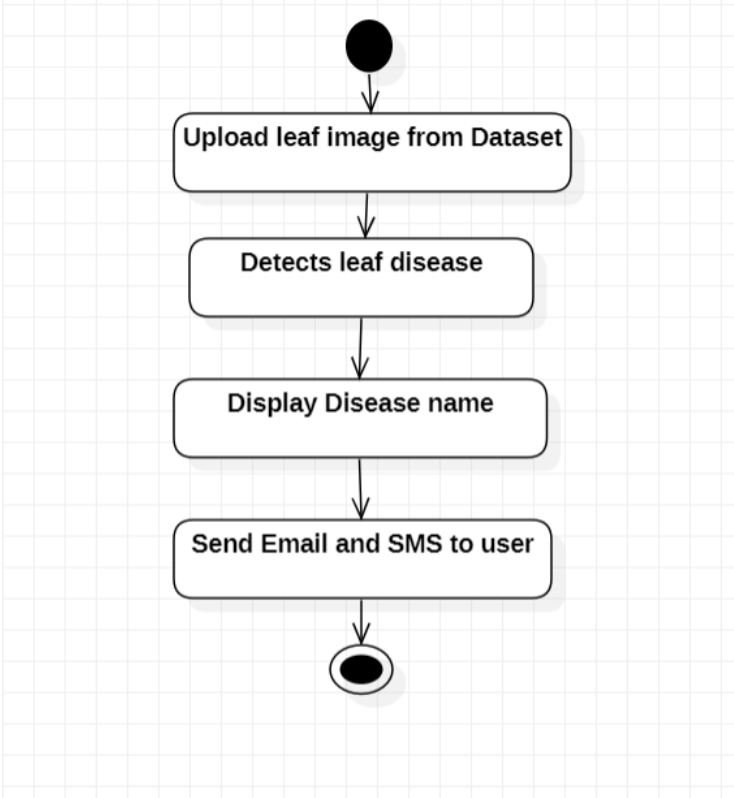
Software Requirements

Software	**Minimum Requirements**
VNC Viewer(OS Compatibility)	VNC Viewer is used for local computers and mobile devices you want to control from.
Python programming language	Python is modest, easy to learn. It is required for the programming of code related to raspberry Pi.
Email	Electronic mail (email) is a method of exchanging messages ("mail") between people using electronic devices.
SMS	SMS stands for Short Message Service and is commonly known as texting.
Browser Support	Application supports all popular browsers.

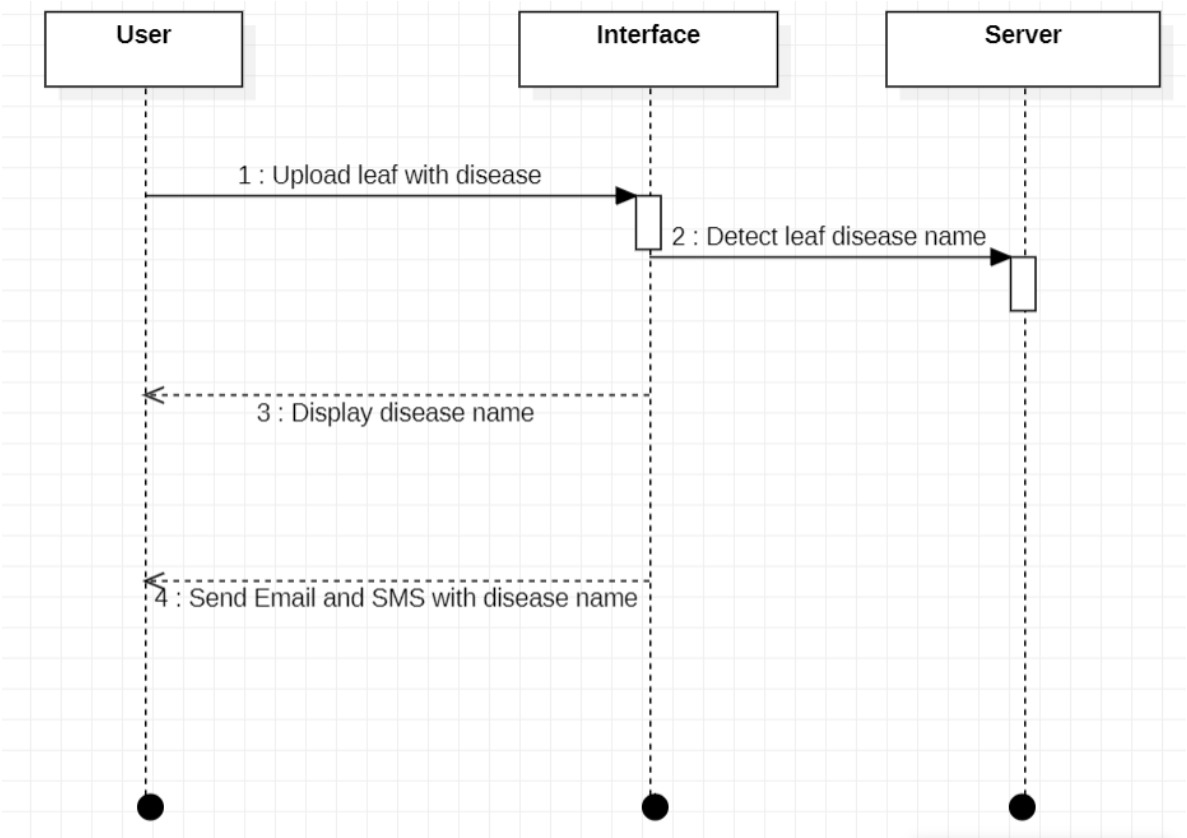
USE CASE DIAGRAM



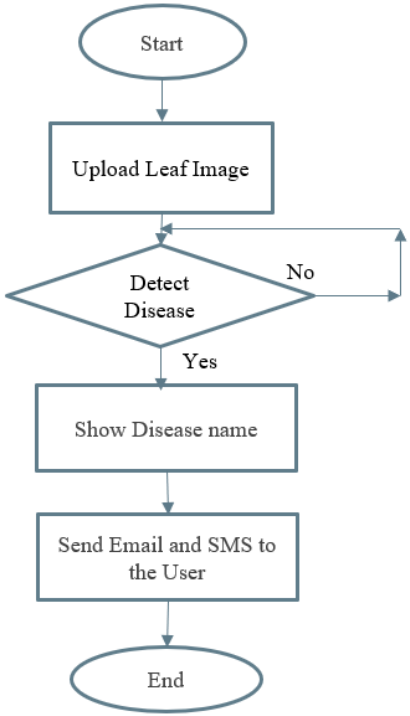
ACTIVITY DIAGRAM



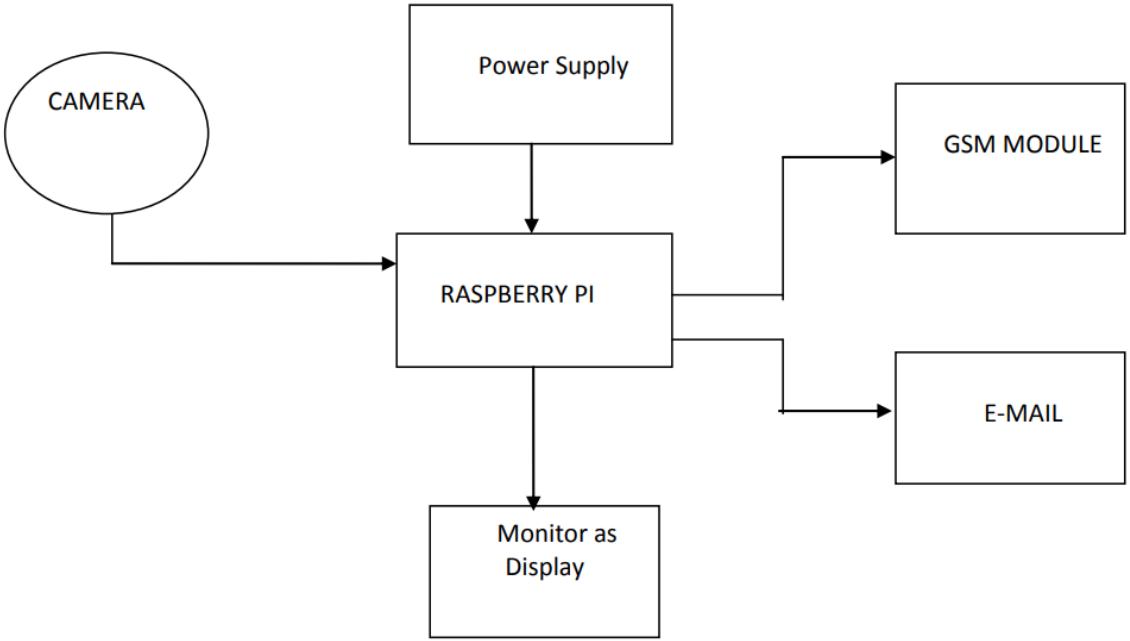
SEQUENCE DIAGRAM



FLOW DIAGRAM



ARCHITECTURE DIAGRAM



DATASET DESCRIPTION

- Dataset is taken from Kaggle.
- There were many unique classes of disease in the sample. Tomato leaves and corn leaves of four types and nine types of leaf infected images are considered.
- We used leaf infected disease photo to identify the disease name.
- Preparing the dataset was the first stage in processing the existing dataset.
- The Convolutional Neural Network process was used during this step as image data input, which eventually formed a model that assessed performance.
- CNNs are used for a variety of tasks in computer vision, primarily image classification and object detection. The open source TensorFlow framework allows you to create highly flexible CNN architectures for computer vision tasks.

CODE

camera.py

```
import os

cam = cv2.VideoCapture(0)

cv2.namedWindow("test")

img_counter = 0

save path="/home/pi/Images"

while True:
    ret, frame = cam. read ()
    cv2.imshow("test", frame)
    if not ret:
        break
    k = cv2.waitKey(1)

    if k%256 == 27:
        # ESC pressed
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        # SPACE pressed
        img_name = "leaf_cameraimage_ {}.jpg". format(img_counter)
        cv2.imwrite(os. path. join (save path, img_name), frame)
        print ("{} written!". Format(img_name))
        img_counter += 1

cam. release ()

cv2.destroyAllWindows()
```

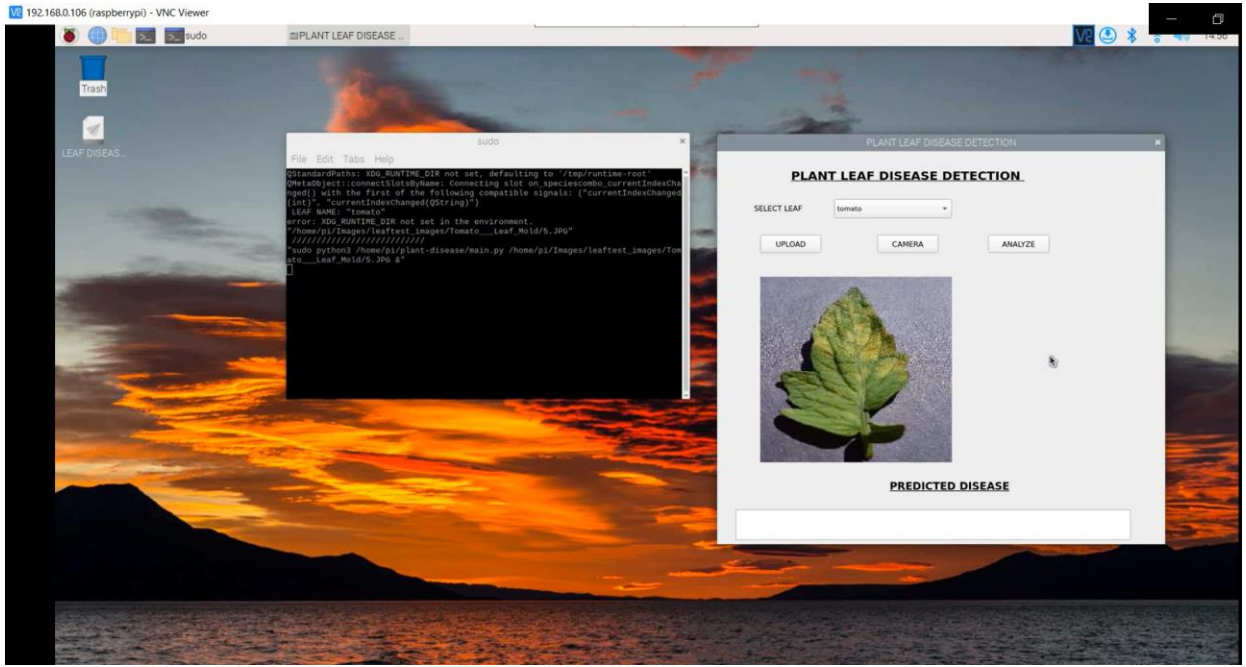
```
Main.py
import os
import argparse
import subprocess
import numpy as np
import cv2
import numpy as np

from PIL import Image

from tensorflow. python. keras. models inport load model
from tensorflow. python. keras. preprocessing import image
from tensorflow. python. keras. applications. inception_v3 import
preprocess_input

import smtplib, ssl
from time import sleep
from email.mine.multipart import MIMEMultipart
from email.mine.base import MIMEBase
from email.mine.text import MIMEText
from email.utils import formatdate
from email import encoders
import serial
import string
import time

SERIAL_PORT = "/dev/ttyUSB0"  # Raspberry Pi 2
ser2 = serial. Serial(SERIAL_PORT, baudrate = 9600, tineout = 5)

print (" ")
cmd1="AT+CMGF=1\r"
cmd21 ="AT+CMGS="
quote="\""
smsnumber="9030995234"
cmd22= "\r"
#cmd2 = cmd21 + quote+ number + quote

cmd3="chr (26))"
sentemail=0

def sendsms(msg):
    cmd2 = cmd21 + quote+ smsnumber + quote
    print(cmd2)
    ser2.write(cmd1.encode())
    tine. sleep (3)
    ser2.write(cmd2.encode())
    ser2.write(cmd22.encode())
    tine. sleep (3)
    print ("Sending SMS with status info:" + msg)
    ser2.write(msg. encode ())
    time. sleep (2)
    ser2.write(chr (26). encode())
    time. sleep (3)



def send_an_email (leafinagepath, identifeddisease):
    toaddr = 'cvrcse2018batch14@gmail.com'      # To id
    me = 'cvrcse2018batch14@gmail.com'          # your id
    subject = "LEAF ANALYZE RESULT"             # Subject
    print ("leafinagepath:" leafinagepath)
    print ("natter:", identifeddisease)
    msg = MIMEMultipart ()
    msg['Subject'] = subject
    msg['From'] = me
    msg['To'] = toaddr
    msg. preamble = identifeddisease
    msg. attach (MIMEText(identifeddisease))

    part = MIMEBase ('application', "octet-stream")
    part.set_payload (open (leafinagepath, "rb"). read ())
    encoders. encode_base64(part)
    part.add_header ('Content-Disposition', 'attachment;
filename="leaf_image.jpg"') # File name and format name
    msg. attach(part)

    try:
        s = smtplib. SMTP ('smtp.gmail.com', 587)  # Protocol
        s.ehlo()
        s. starttls()
        s.ehlo()
        s.login (user = 'cvrcse2018batch14@gmail.com', password =
'cvrcse2018')  # User id & password
        #s.send_message(msg)
        s. sendmail (me, toaddr, msg.as_string ())
        s. quit ()
        sentemail=0
    #except:
    #   print ("Error: unable to send email")
    except SMTPException as error:
          print ("Error")             # Exception
          sentemail=0



CORN = 'corn'
TOMATO = 'tomato'

SPECIES = [CORN, TOMATO]
DISEASE_SUPPORTED_SPECIES = {CORN, TOMATO}

CORN_CLASSES = ['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
'Corn_(maize)___Common_rust_',
                'Corn_(maize)___Northern_Leaf_Blight',
'Corn_(maize)___healthy']

TOMATO_CLASSES = ['Tomato___Bacterial_spot', 'Tomato___Early_blight',
'Tomato___Late_blight', 'Tomato___Leaf_Mold',
                'Tomato___Septoria_leaf_spot', 'Tomato___Spider_mites Two-
spotted_spider_mite',
                'Tomato___Target_Spot',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus', 'Tomato___Tomato_mosaic_virus',
                'Tomato___healthy']


PLANT_CLASSES = {
    CORN: CORN_CLASSES,
    TOMATO: TOMATO_CLASSES
}


VGG_ARCHITECTURE = 'vgg'
INCEPTIONV3_ARCHITECTURE = 'inceptionv3'
SUPPORTED_MODEL_TYPES = {VGG_ARCHITECTURE, INCEPTIONV3_ARCHITECTURE}


DISEASE_DETECTION = 'disease_detection'



TARGET_IMAGE_SIZES = {
    VGG_ARCHITECTURE: {
        DISEASE_DETECTION: (64, 64),

    },
    INCEPTIONV3_ARCHITECTURE: {
        DISEASE_DETECTION: (100, 100),

    }
}


VGG_MODELS = {
    CORN: '/home/pi/plant-disease/Models/Corn_0.8926_VGG.h5',
    TOMATO: '/home/pi/plant-disease/Models/Tomato_0.8675_VGG.h5'
}
```

```
INCEPTIONV3_MODELS = {
    CORN: '/home/pi/plant-disease/Models/InceptionV3-scratch_segCorn.h5',
    TOMATO: '/home/pi/plant-disease/Models/InceptionV3-scratch_segTomato.h5'
}


MODEL_STORAGE_BASE = '/home/pi/plant-disease/Models'


def get_classes(species_name):

    return PLANT_CLASSES[species_name]


def get_disease_model(species, model type):

    if species not in DISEASE_SUPPORTED_SPECIES:
        raise ValueError("`{}` species has no disease model yet.\n"
                         "Species tha have disease models are
{}".format(species, DISEASE_SUPPORTED_SPECIES))

    if model_type == VGG_ARCHITECTURE:
        return VGG_MODELS[species]
    elif model_type == INCEPTIONV3_ARCHITECTURE:
        return INCEPTIONV3_MODELS[species]
    else:
        raise ValueError("No such `{}` model type is supported.\n"
                         "Supported model types are {}".format(model_type,
SUPPORTED_MODEL_TYPES))



def get_predictions(model_path, img_path, img_target_size):


    if not os.path.exists(model_path):
        raise ValueError('No such `{}` file found\n'
                         'Please, checkout the readme of the project '
                         'on github and download required models'.
format(model_path))
    model = load_model(model_path)

    pil_img = Image.open(img_path)
    if pil_img. size!= img_target_size:
        pil_img = pil_img.resize(img_target_size)

    img = image.img_to_array(pil_img)
```

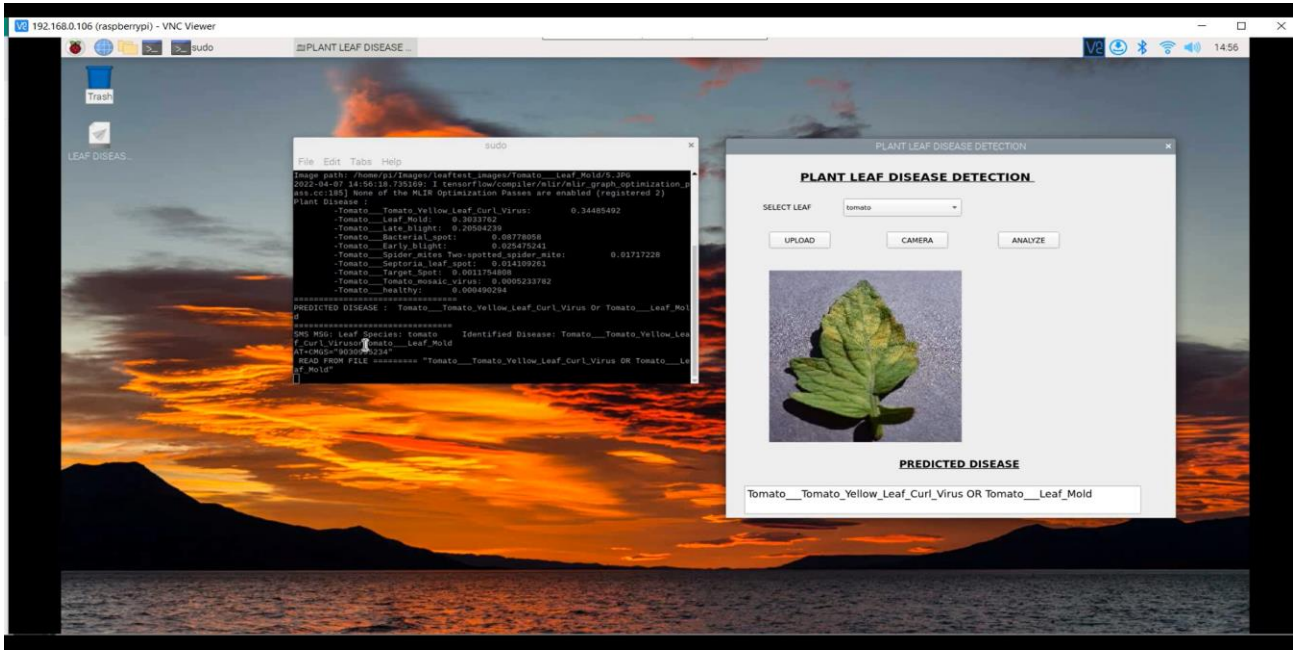
```
    if img.shape[2] == 4:
        img = img[:, :, :3]

    img = np.expand_dims (img, axis=0)
    img = preprocess_input(img)

    preds = model. Predict(img).flatten()

    value_ = preds.argsort()
    sorted_preds_index = value_[::-1]

    return preds, sorted_preds_index


def segment_image(img_path):

    image_name, extension = os.path.splitext(img_path)
    segmented_image_name = image_name + "_marked" + extension  # the future
segmented image name to be
    result = subprocess.check_output(['python', "leaf-image-
segmentation/segment.py", "-s", img_path])
    print('Info: Input image segmented.')

    return segmented_image_name



def predict_disease(img_path, species, model_type=VGG_ARCHITECTURE,
do_print=True):

    if species not in SPECIES:
        raise ValueError("No such `{}` species is supported.\n"
                         "Supported species are {}".format(species, SPECIES))

    if species not in DISEASE_SUPPORTED_SPECIES:
        print("Info: For `{}` species, a disease can not be predicted "
              "since its disease model is not implemented
yet.".format(species))
        return None
    else:
        SPECIES_CLASSES = get_classes(species)
        model_path = os.path.join(MODEL_STORAGE_BASE,
get_disease_model(species, model_type))

        target_image_size = TARGET_IMAGE_SIZES[model_type][DISEASE_DETECTION]
        preds, sorted_preds_index = get_predictions(model_path, img_path,
target_image_size)

        if do_print:
```

```
            print("Plant Disease : ")
            for i in sorted_preds_index:
                print("\t-" + str(SPECIES_CLASSES[i]) + ": \t" +
str(preds[i]))

        return str
(SPECIES_CLASSES[sorted_preds_index[0]]),str(SPECIES_CLASSES[sorted_preds_inde
x[1]])


def get_cmd_args():

    parser = argparse.ArgumentParser()
    parser.add_argument("image", type=str, help='Image file path')
    args = parser.parse_args()

    return args



if __name__ == "__main__":

        args = get_cmd_args ()
        f = open ("/home/pi/build-leafdiseasegui/species.txt", "r")
        species=f. readline ()
        print ("SPECIES: ", species)
        f. close()
        leafimagepath=args. image
        print("Image path:",leafimagepath)
        diseasereturn1, diseasereturn2=predict_disease(args.image, species)
        print ("==============================")
        print("PREDICTED DISEASE : ", diseasereturn1 , "Or", diseasereturn2)
        print("==============================")
        f = open("/home/pi/build-leafdiseasegui/diseasename.txt", "w")
        f.write(diseasereturn1)
        f.write(" OR ")
        f.write(diseasereturn2)
        f.close()
        f = open("/home/pi/build-leafdiseasegui/readfilebit.txt", "w")
        f.write("1")
        f.close()
        identifeddisease="Leaf Species: "+species+"     "+ "Identified
Disease: "+ diseasereturn1+ "or" + diseasereturn2
        print("SMS MSG:", identifeddisease)
        sendsms(identifeddisease)
        time.sleep(2)
        send_an_email(leafimagepath,identifeddisease)
```

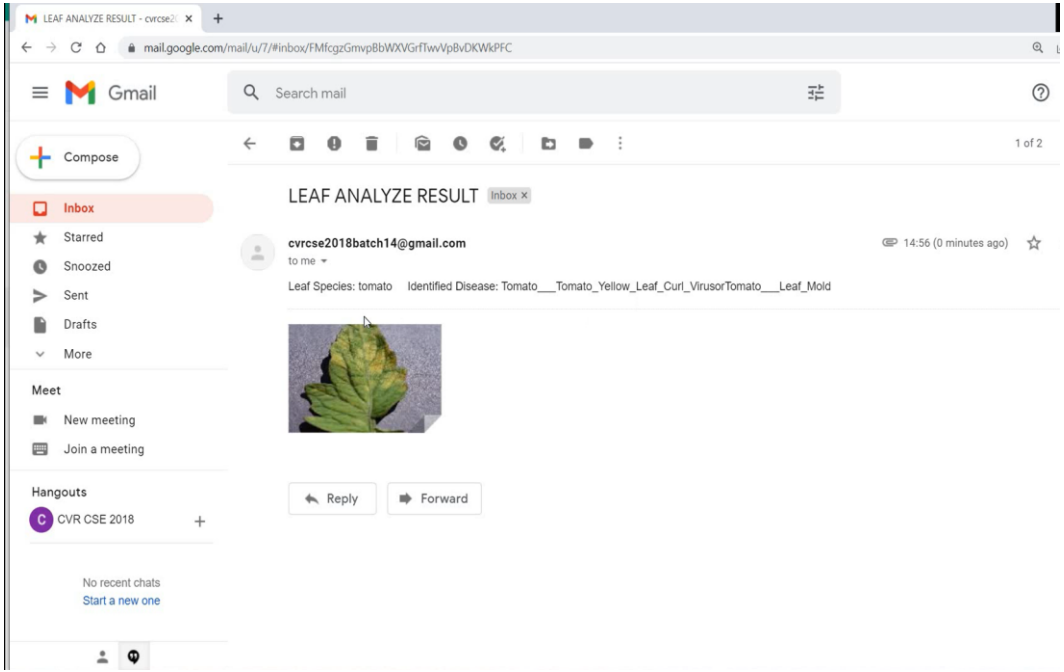
OUTPUT (PLANT LEAF DISEASE DETECTION)



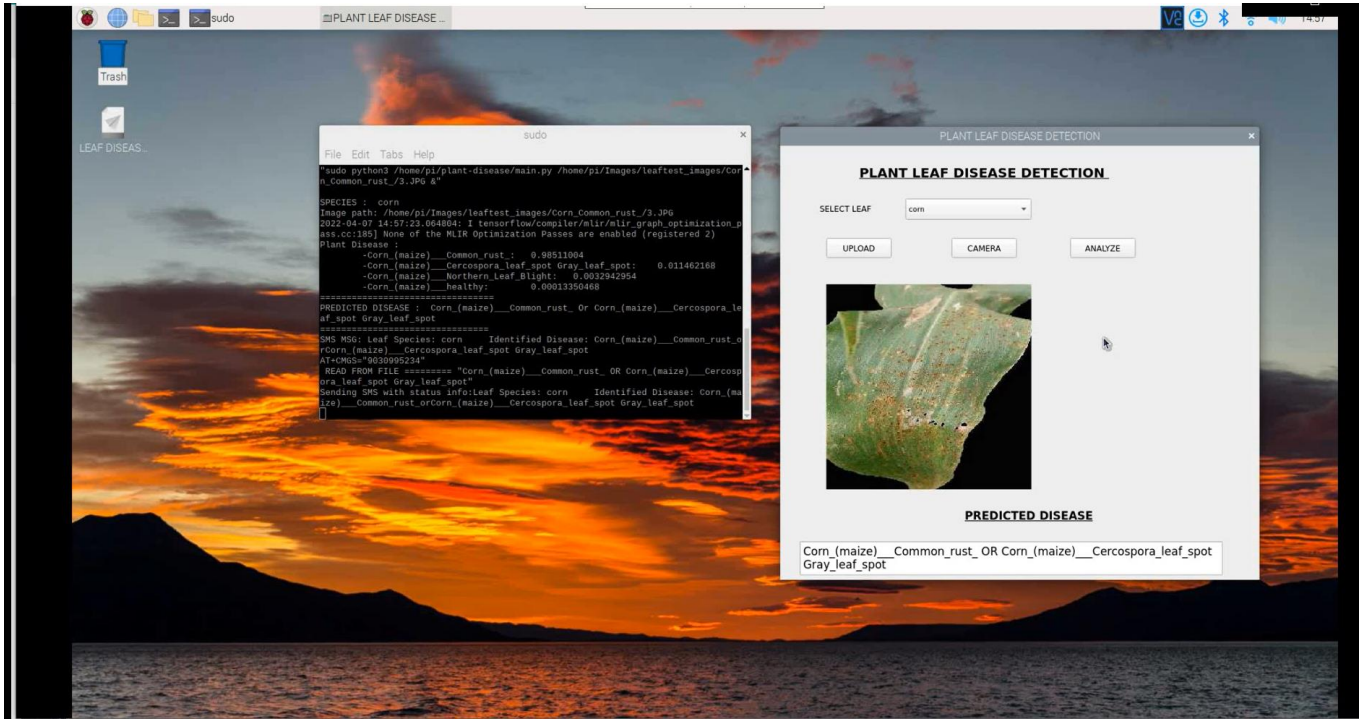
OUTPUT


192.168.0.106 (raspberrypi) - VNC Viewer
sudo
PLANT LEAF DISEASE ...
14:56
Trash
LEAF DISEAS...
sudo
File Edit Tabs Help
Image path: /home/pi/Images/leaftest_images/Tomato___Leaf_Mold/5.JPG
Plant Disease :
-Tomato___Tomato_Yellow_Leaf_Curl_Virus: 0.34485492
-Tomato___Leaf_Mold: 0.3033762
-Tomato___Late_blight: 0.20504239
-Tomato___Bacterial_spot: 0.08778058
-Tomato___Early_blight: 0.025475241
-Tomato___Spider_mites Two-spotted_spider_mite: 0.01717228
-Tomato___Septoria_leaf_spot: 0.014109261
-Tomato___Target_Spot: 0.0011754808
-Tomato___Tomato_mosaic_virus: 0.0005233782
-Tomato___healthy: 0.000490294
PREDICTED DISEASE : Tomato___Tomato_Yellow_Leaf_Curl_Virus Or Tomato___Leaf_Mold
SMS MSG: Leaf Species: tomato Identified Disease: Tomato___Tomato_Yellow_Leaf_Curl_Virus Or Tomato___Leaf_Mold
READ FROM FILE ========= "Tomato___Tomato_Yellow_Leaf_Curl_Virus OR Tomato___Leaf_Mold"
PLANT LEAF DISEASE DETECTION
PLANT LEAF DISEASE DETECTION
SELECT LEAF
tomato
UPLOAD
CAMERA
ANALYZE
PREDICTED DISEASE
Tomato___Tomato_Yellow_Leaf_Curl_Virus OR Tomato___Leaf_Mold

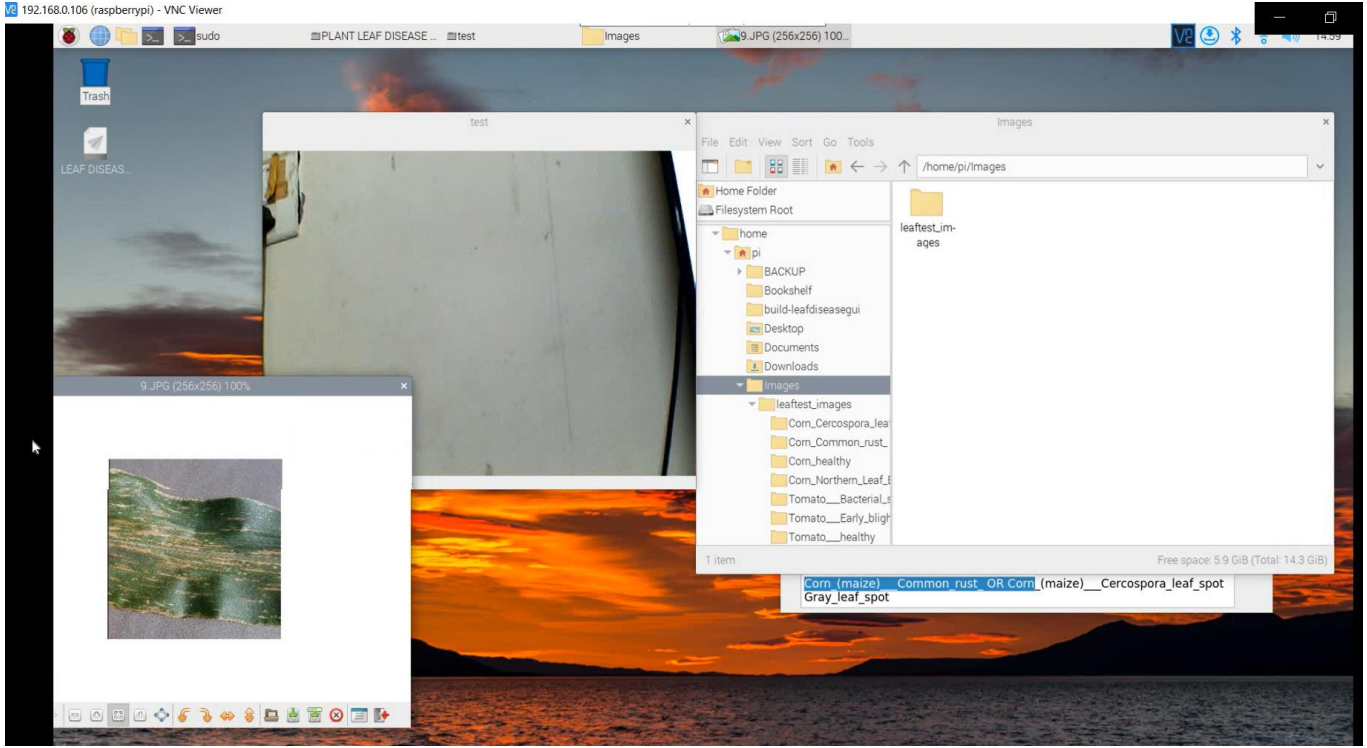
OUTPUT(EMAIL)



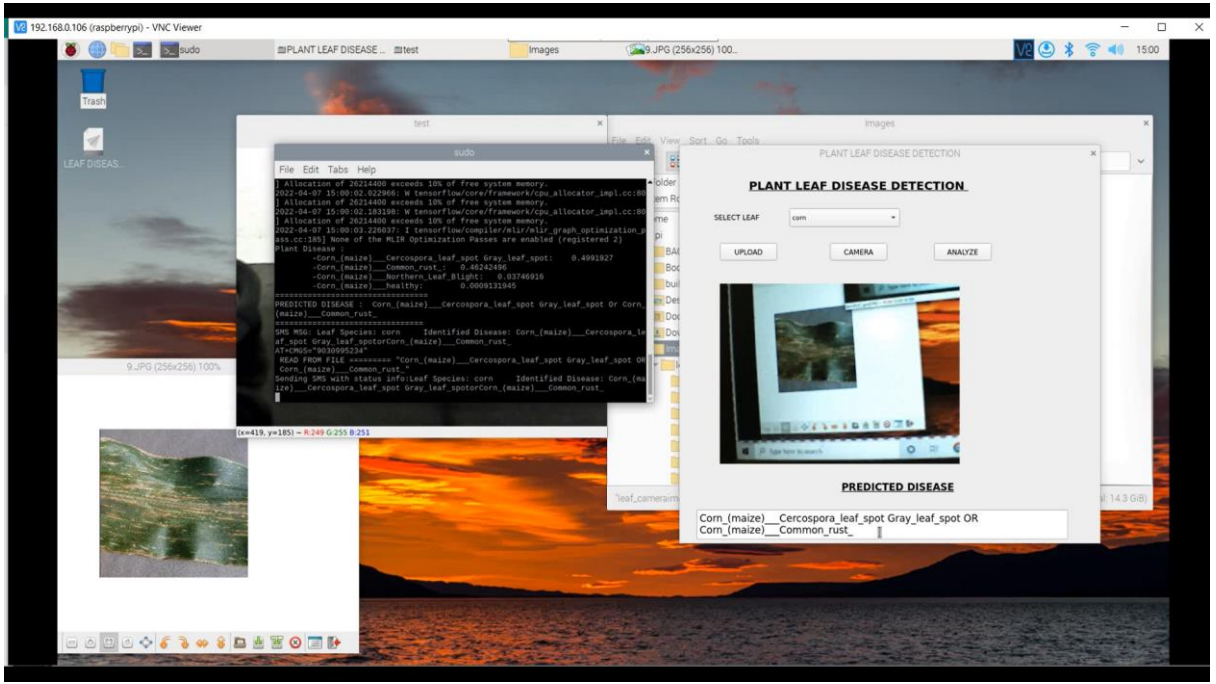
OUTPUT


sudo
PLANT LEAF DISEASE ...
Trash
LEAF DISEAS...
sudo
File Edit Tabs Help
"sudo python3 /home/pi/plant-disease/main.py /home/pi/Images/leaftest_images/Corn_Common_rust_/3.JPG &"
SPECIES : corn
Image path: /home/pi/Images/leaftest_images/Corn_Common_rust_/3.JPG
2022-04-07 14:57:23.064864: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
Plant Disease :
-Corn_(maize)___Common_rust_: 0.98511004
-Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot: 0.011462168
-Corn_(maize)___Northern_Leaf_Blight: 0.0032942954
-Corn_(maize)___healthy: 0.00013350468
PREDICTED DISEASE : Corn_(maize)___Common_rust_ Or Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
SMS MSG: Leaf Species: corn Identified Disease: Corn_(maize)___Common_rust_orCorn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
AT+CMGS="9030995234"
READ FROM FILE ========= "Corn_(maize)___Common_rust_ OR Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot"
Sending SMS with status info:Leaf Species: corn Identified Disease: Corn_(maize)___Common_rust_orCorn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
PLANT LEAF DISEASE DETECTION
PLANT LEAF DISEASE DETECTION
SELECT LEAF
corn
UPLOAD
CAMERA
ANALYZE
PREDICTED DISEASE
Corn_(maize)___Common_rust_ OR Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot

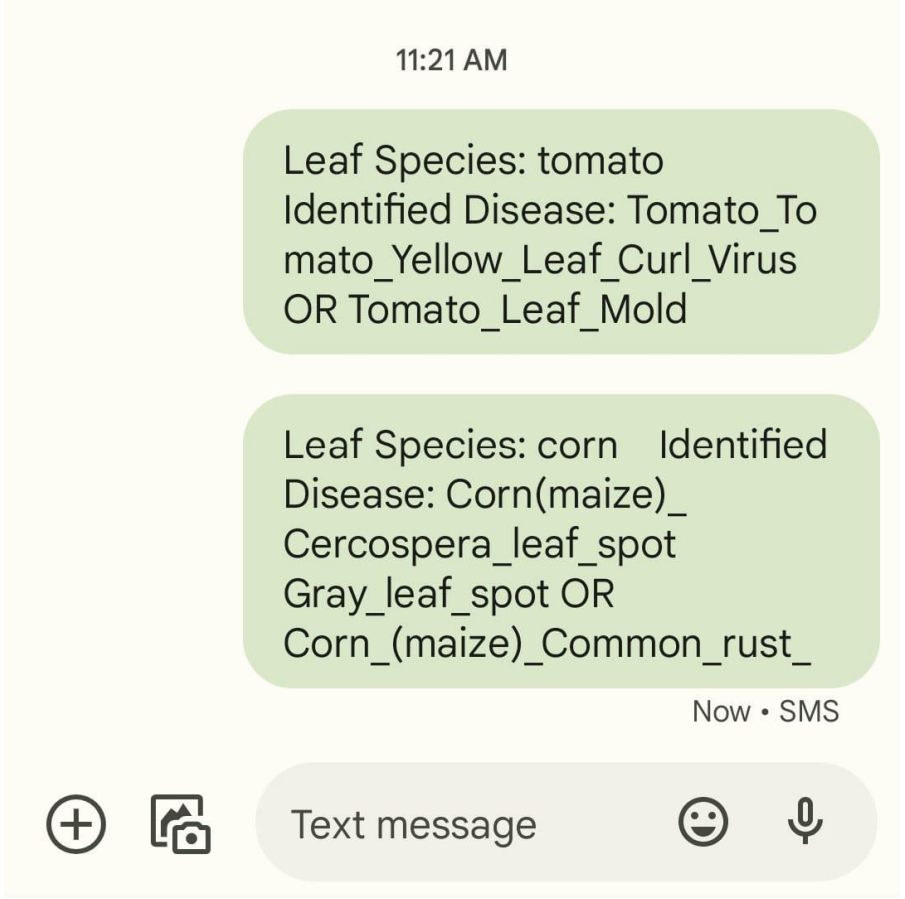
OUTPUT


192.168.0.106 (raspberrypi) - VNC Viewer
sudo
PLANT LEAF DISEASE ...
test
Images
9.JPG (256x256) 100...
Trash
LEAF DISEAS...
test
Images
File Edit View Sort Go Tools
/home/pi/Images
Home Folder
Filesystem Root
home
pi
BACKUP
Bookshelf
build-leafdiseasegui
Desktop
Documents
Downloads
Images
leaftest_images
Corn_Cercospora_lea
Corn_Common_rust_
Corn_healthy
Corn_Northern_Leaf_B
Tomato___Bacterial_s
Tomato___Early_bligh
Tomato___healthy
leaftest_im-
ages
1 item
Free space: 5.9 GiB (Total: 14.3 GiB)
9.JPG (256x256) 100%
Corn_(maize)___Common_rust_ OR Corn_(maize)___Cercospora_leaf_spot
Gray_leaf_spot

OUTPUT


192.168.0.106 (raspberrypi) - VNC Viewer
PLANT LEAF DISEASE DETECTION
SELECT LEAF
corn
UPLOAD
CAMERA
ANALYZE
PREDICTED DISEASE
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot OR
Corn_(maize)___Common_rust_

OUTPUT(SMS)


11:21 AM
Leaf Species: tomato
Identified Disease: Tomato_To
mato_Yellow_Leaf_Curl_Virus
OR Tomato_Leaf_Mold
Leaf Species: corn Identified
Disease: Corn(maize)_
Cercospera_leaf_spot
Gray_leaf_spot OR
Corn_(maize)_Common_rust_
Now • SMS
Text message

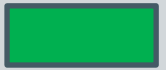
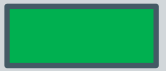
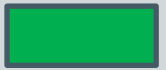
CONCLUSION

- It is important in plant disease detection to have the accuracy in the plant disease detection but at the same time the process should be of high speed. This system can be connected to the server for further processing.
- The objective of this work is the detection, classification of leaf diseases using image processing tools and all information about the disease is sent to the farmer’s mobile phone through the GSM module.
- To increase the speed and accuracy of detection as well as classification of leaf diseases we are using Raspberry pi 3 model B module.
- This system will largely contribute in growth in the yield of the farms.

FUTURE ENHANCEMENTS

- In this project we proposed system using raspberry pi which can detect disease infected leaf. The project has many verticle in it leaf detection.
- In this project we have achieved in detecting the disease affected leaf.
- In the future, we will expand the model to include certain abiotic diseases due to the deficiency of nutrient values in the crop leaf.
- The main objective is to increase unique data collection and accumulate a vast amount of data on several diseases of plants.
- To improve accuracy, we will apply subsequent technology in the future.

Action Plan

Serial Number	Activity	Planned Start Date	Planned End Date	Current Status	RAG Status
1	Problem Statement	10-02-2022	12-02-2022	Completed	
2	Brain Storming for a Solution	05-02-2022	12-02-2022	Completed	
3	High Level Design	11-03-2022	23-03-2022	Completed	
4	Coding	25-02-2022	26-03-2022	Completed	
5	Testing and Deployment	25-03-2022	27-04-2022	Completed	

REFERENCES

- M. Reyalat, Al Hiary, Bani Ahmad, Braik and Z. Rahamneh, “Fast and Accurate Detection and Classification of Plant Diseases”.
- Smith J S., Camargo A., “An image processing based algorithm to automatically identify plant disease visual symptoms”.
- Stilwell, M. The Global Tomato Online News Processing in 2018. Available online: https://www.tomatonews.com/ (accessed on 15 November 2021).
- Yu, J.; Wang, J.; Leblon, B. Evaluation of Soil Properties, Topographic Metrics, Plant Height, and Unmanned Aerial Vehicle Multispectral Imagery Using Machine Learning Methods to Estimate Canopy Nitrogen Weight in Corn. Remote Sens. 2021, 13, 3105.
- Xie, Q.; Wang, J.; Lopez-Sanchez, J.M.; Peng, X.; Liao, C.; Shang, J.; Zhu, J.; Fu, H.; Ballester-Berman, J.D. Crop height estimation of corn from multi-year RADARSAT-2 polarimetric observables using machine learning. Remote Sens. 2021, 13, 392.

THANK YOU