

```

import tkinter as tk

from PIL import Image, ImageTk

from tkinter.filedialog import askopenfilename

from keras.preprocessing import image

from keras.models import load_model

import requests

import numpy as np

import cv2

import shutil

import time

import imutils

import requests

```

```

dis_list = [0,1,2]

```

```

li = ['Apple___Apple_scab\n\n\nPesticides:\nBonide Sulfur Plant Fungicide\nrganocide\nBonide Orchard Spray', 'Apple___Black_rot\n\n\nPesticides:\nPhysan 20\nNEEM oil\nCopper sprays\nOrchard Spray', 'Apple___Cedar_apple_rust\n\n\nPesticides:\nSERENADE Garden\nOrchard Spray\nSulfur Plant Fungicide',

```

```

    'Apple___healthy', 'Blueberry___healthy',
    'Cherry___Powdery_mildew\n\n\nPesticides:\norganic compost\nNeem oil and PM Wash\nZero Tolerance Fungicide',

```

```

    'Cherry___healthy', 'Corn_(maize)___Cercospora_leaf_spot\n\n\nPesticides:\nsulfur sprays\ncopper-based fungicides\nGarden Dust',

```

```

    'Corn_(maize)___Common_rust\n\n\nPesticides:\nSERENADE Garden\nOrchard Spray\nSulfur Plant Fungicide', 'Corn_(maize)___Northern_Leaf_Blight\n\n\nNo Pesticides Found\n',
    'Corn_(maize)___healthy',

```

```

    'Grape___Black_rot\n\n\nPesticides:\nPhysan 20\nNEEM oil\nCopper sprays\nOrchard Spray',
    'Grape___Esca_(Black_Measles)\n\n\nNo Pesticides Found\n', 'Grape___Leaf_blight\n\n\nNo Pesticides Found\n',

```

```

    'Grape___healthy', 'Orange___Haunglongbing\n\n\nNo Pesticides Found\n',
    'Peach___Bacterial_spot\n\n\nNo Pesticides Found\n',

```

```

    'Peach___healthy', 'Pepper,_bell___Bacterial_spot\n\n\nNo Pesticides Found\n',
    'Pepper,_bell___healthy', 'Potato___Early_blight\n\n\nPesticides:\nFontelis\nEndura, Lance WDG\nCabrio\nReason',

```

```
'Potato___Late_blight\n\n\nPesticides:\ncopper based fungicide\nOrganocide\nLiquid Copper',  
'Potato___healthy', 'Raspberry___healthy', 'Soybean___healthy',
```

```
'Squash___Powdery_mildew\n\n\nPesticides:\norganic compost\nNeem oil and PM Wash\nZero  
Tolerance Fungicide', 'Strawberry___Leaf_scorch\n\n\n\nNo Pesticides Found\n',  
'Strawberry___healthy', 'Tomato___Bacterial_spot\n\n\n\nNo Pesticides Found',
```

```
'Tomato___Early_blight\n\n\nPesticides:\nFontelis\nEndura, Lance WDG\nCabrio\nReason',  
'Tomato___Late_blight\n\n\nPesticides:\ncopper based fungicide\nOrganocide\nLiquid Copper',  
'Tomato___Leaf_Mold\n\n\n\nNo Pesticides Found', 'Tomato___Septoria_leaf_spot\n\n\n\nNo  
Pesticides Found',
```

```
'Tomato___Spider_mites\n\n\n\nNo Pesticides Found\n', 'Tomato___Target_Spot\n\n\n\nNo  
Pesticides Found\n', 'Tomato___Tomato_Yellow_Leaf_Curl_Virus\n\n\n\nPesticides:\nsulfur or  
copper-based fungicides\nGarden Dust\norganic fertilizers high in nitrogen\nLiquid Copper',
```

```
'Tomato___Tomato_mosaic_virus\n\n\n\nPesticides:\nSafer Soap, Bon-Neem\nHarvest-Guard row  
cover\nleast-toxic herbicides\nAllDown', 'Tomato___healthy']
```

```
classifier = load_model('PLANT_MODEL.hdf5')
```

```
diseasename = None
```

```
root = tk.Tk()
```

```
root.title("Plant_Leaf")
```

```
root.geometry("900x550")
```

```
root.configure(background="white")
```

```
title = tk.Label(text="Select An Image To Process", background = "white", fg="Brown", font=("", 15))
```

```
title.grid(row=0, column=2, padx=10, pady = 10)
```

```
def update():
```

```
    put_header = {"Content-Type":"application/json"}
```

```
    val =
```

```
    requests.get("http://188.166.206.43/StslowrJ_pSOsRWsG03B02CWmex1ENGi/update/V8?value=1")
```

```
    print(val)
```

```

def exit():
    root.destroy()

def clear():
    disease = tk.Label(text='
background="white",
                        fg="white", font=("", 20))
    disease.grid(column=4, row=3, padx=10, pady=10)

def analysis():
    image_path = path
    new_img = image.load_img(image_path, target_size=(224, 224))
    img = image.img_to_array(new_img)
    img = np.expand_dims(img, axis=0)
    img = img/255

    print("Following is our prediction:")
    prediction = classifier.predict(img)
    # decode the results into a list of tuples (class, description, probability)
    # (one such list for each sample in the batch)
    d = prediction.flatten()
    j = d.max()
    for index,item in enumerate(d):
        if item == j:
            print(index)
            if(index in dis_list):
                print("present")
                update()

```

```

class_name = li[index]
print(class_name)
diseasename = class_name
disease = tk.Label(text='Status: ' + diseasename, background="white",
                    fg="Black", font=("", 15))
disease.grid(column=4, row=3, padx=10, pady=10)
button3 = tk.Button(text="Clear", command=clear)
button3.grid(row=6, column=2, padx=10, pady = 10)
button4 = tk.Button(text="Exit", command=exit)
button4.grid(row=7, column=2, padx=10, pady = 10)

def openphoto():
    global path
    path=askopenfilename(filetypes=[("Image File", '.jpg')])
    im = Image.open(path)
    tkimage = ImageTk.PhotoImage(im)
    myvar=tk.Label(root,image = tkimage, height="224", width="224")
    myvar.image = tkimage
    myvar.place(x=1, y=0)
    myvar.grid(row=3, column=2 , padx=10, pady = 10)
    button2 = tk.Button(text="Analyse Image", command=analysis)
    button2.grid(row=4, column=2, padx=10, pady = 10)

button1 = tk.Button(text="Select Photo", command = openphoto)
button1.grid(row=1, column=2, padx=10, pady = 10)

root.mainloop()

```