import keras

from keras.utils import np\_utils

import numpy as np

import os

import cv2

import random

from glob import glob

import keras

from tensorflow.keras.layers import Input, Convolution2D, Conv2DTranspose, Conv2D

from tensorflow.keras.models import Model

from tensorflow.keras.callbacks import LearningRateScheduler

from tensorflow.keras import backend as K

from tensorflow.keras.optimizers import Adam

import matplotlib.pyplot as plt

import cv2

import numpy as np

import matplotlib.pyplot as plt

import numpy as np

import cv2

import matplotlib.pyplot as plt

from scipy import ndimage

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler

from matplotlib.colors import hsv\_to\_rgb

import mahotas

from tkinter import filedialog

clas1 = [item[10:-1] for item in sorted(glob("./dataset/\*/"))]

from keras.preprocessing import image

from tqdm import tqdm

# Note: modified these two functions, so that we can later also read the inception tensors which

# have a different format

def path\_to\_tensor(img\_path, width=224, height=224):

# loads RGB image as PIL.Image.Image type

print(img\_path)

img = image.load\_img(img\_path, target\_size=(width, height))

# convert PIL.Image.Image type to 3D tensor with shape (width, heigth, 3)

x = image.img\_to\_array(img)

# convert 3D tensor to 4D tensor with shape (1, width, height, 3) and return 4D tensor

return np.expand\_dims(x, axis=0)

def paths\_to\_tensor(img\_paths, width=224, height=224):

list\_of\_tensors = [path\_to\_tensor(img\_paths, width, height)]

return np.vstack(list\_of\_tensors)

#vilization\_and\_show()

from PIL import ImageFile

ImageFile.LOAD\_TRUNCATED\_IMAGES = True

#from tkinter import filedialog

#filename = filedialog.askopenfilename(title='open')

#main\_img = cv2.imread(filename)

filename = filedialog.askopenfilename(title='open')

img = cv2.imread(filename )

plt.imshow(img)

plt.show()

bins = 8

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.imshow(img)

plt.show()

hsv\_img = cv2.cvtColor(img, cv2.COLOR\_RGB2HSV)

plt.imshow(hsv\_img)

lower\_green = np.array([25,0,20])

upper\_green = np.array([100,255,255])

mask = cv2.inRange(hsv\_img, lower\_green, upper\_green)

result = cv2.bitwise\_and(img, img, mask=mask)

plt.subplot(1, 2, 1)

plt.imshow(mask, cmap="gray")

plt.subplot(1, 2, 2)

plt.imshow(result)

plt.show()

lower\_brown = np.array([10,0,10])

upper\_brown = np.array([30,255,255])

disease\_mask = cv2.inRange(hsv\_img, lower\_brown, upper\_brown)

disease\_result = cv2.bitwise\_and(img, img, mask=disease\_mask)

plt.subplot(1, 2, 1)

plt.imshow(disease\_mask, cmap="gray")

plt.subplot(1, 2, 2)

plt.imshow(disease\_result)

plt.show()

'''

final\_mask = mask + disease\_mask

final\_result = cv2.bitwise\_and(img, img, mask=final\_mask)

plt.figure(figsize=(15,15))

plt.subplot(1, 2, 1)

plt.imshow(final\_mask, cmap="gray")

plt.subplot(1, 2, 2)

plt.imshow(final\_result)

plt.show()

'''

from twilio.rest import Client

account\_sid = 'AC9cb5136eb1823314138ae6f2d8bb53c0'

auth\_token = 'dbd755d297547698548ab3d5b401b772'

client = Client(account\_sid, auth\_token)

from tensorflow.keras.models import load\_model

model = load\_model('trained\_model\_CNN.h5')

test\_tensors = paths\_to\_tensor(filename)/255

pred=model.predict(test\_tensors)

print('Given Image Predicted is : '+str(clas1[np.argmax(pred)]))

if "downy mildew"==clas1[np.argmax(pred)]:

message = client.messages.create(from\_='+18288274967',body ='Affected grape plantation disease:downy mildew pesticide:chlorothalonil and mancozeb',to ='+918217822388')

if "powdery mildew"==clas1[np.argmax(pred)]:

message = client.messages.create(from\_='+18288274967',body ='Affected grape plantation disease:powdery mildew pesticide:potassium bicarbonate ',to ='+918217822388')

if "healthy"==clas1[np.argmax(pred)]:

message = client.messages.create(from\_='+18288274967',body ='healthy plant',to ='+918217822388')