# AUTOMATIC TRAFFIC CONTROL SYSTEM

import tkinter as tk  
from tkinter import filedialog  
from PIL import ImageTk,Image  
import os  
import numpy as np  
import PIL.Image  
from tkinter import \*  
import cv2  
import cvlib as cv  
from cvlib.object\_detection import draw\_bbox  
from werkzeug.utils import secure\_filename  
import matplotlib.pyplot as plt  
import tensorflow as tf  
  
from tensorflow.keras.utils import load\_img  
from tensorflow.keras.utils import img\_to\_array  
from keras.applications.imagenet\_utils import preprocess\_input  
  
#from keras.applications.vgg16 import preprocess\_input, decode\_predictions  
#from keras.models import load\_model  
#from keras.preprocessing.image import img\_to\_array, load\_img  
  
  
root = tk.Tk(className='Traffic light direction')  
root.geometry("550x300+300+150")  
root.configure(bg='lavender')  
l=[]  
  
def openfn():  
    filename = filedialog.askopenfilename()  
    l.append(filename)  
    return filename  
def open\_img():  
    x = openfn()  
    img = PIL.Image.open(x)  
    img = img.resize((400,200))  
    img.thumbnail(((root.winfo\_width()/5),root.winfo\_height()/5))  
    img = ImageTk.PhotoImage(img)  
    panel = Label(root, image=img)  
    panel.image = img  
    panel.pack()  
  
  
upload2=Button(root, text="Upload an image",command=open\_img,  
            padx=0.2,pady=2)  
  
upload2.configure(background='#364156', foreground='white',  
        font=('arial',10,'bold'))  
  
upload2.place(x=300, y=350)  
  
  
  
def classify(l):  
    min\_veh=0  
    min\_time=0  
    max\_veh=0  
    max\_time=0  
    #min\_img=None  
    #max\_img=None  
    vehicle\_list = []  
    for img\_path in l:  
        im = cv2.imread(img\_path)  
        #print(im.shape)  
  
        bbox, label, conf = cv.detect\_common\_objects(im)  
  
        output\_image = draw\_bbox(im, bbox, label, conf)  
        plt.imshow(output\_image)  
        plt.show()  
        image = load\_img(img\_path, target\_size=(100, 100))  
        image = img\_to\_array(image)  
        image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))  
        image = preprocess\_input(image)  
        im =image  
        N\_car=label.count('car')  
        N\_bus=label.count('bus')  
        N\_motorcycle=label.count('motorcycle')  
        N\_truck=label.count('truck')  
        vehicle = N\_bus + N\_car + N\_motorcycle+N\_truck  
        vehicle\_list.append(vehicle)  
    print(vehicle\_list)  
     
    for count in vehicle\_list:  
        time = 5  
        if vehicle<=55:  
            time+=count  
            print("Time displayed in signal Indicator for",count,"image",time)  
        else:  
            time=60  
            print("Time displayed in signal Indicator for",count,"image",time)  
             
    mini = min(vehicle\_list)  
    ind\_mini = vehicle\_list.index(mini)  
    print(mini)  
    if ind\_mini == 0:  
        btn = Button(root, text="NORTH",padx=10,pady=0.2)  
        btn.configure(background = '#364156', foreground='white',font=('arial',10,'bold'))  
        btn.place(x=1150,y=450)  
    elif ind\_mini == 1:  
        btn = Button(root, text="EAST",padx=0.2,pady=2)  
        btn.configure(background = '#364156', foreground='white',font=('arial',10,'bold'))  
        btn.place(x=1150,y=450)  
    elif ind\_mini == 2:  
        btn = Button(root, text="SOUTH",padx=15,pady=2)  
        btn.configure(background = 'dark orchid', foreground='white',font=('arial',10,'bold'))  
        btn.place(x=1150,y=450)  
    elif ind\_mini == 3:  
        btn = Button(root, text="WEST",padx=0.2,pady=2)  
        btn.configure(background = '#364156', foreground='white',font=('arial',10,'bold'))  
        btn.place(x=1100,y=450)  
  
    for i in vehicle\_list:  
            max\_veh=max(vehicle\_list)  
            if max\_veh <= 55:  
                max\_time = max\_veh+5  
            else:  
                max\_time = 60  
        #max\_img=draw\_bbox(im, bbox, label, conf)  
    print("HIGH DENSITY")  
    #plt.imshow(max\_img)  
    plt.show()  
    print('Number of vehicle in the image is ',max\_veh)  
    print("Time displayed in signal Indicator is ",max\_time)  
  
  
  
def show\_classify\_button():  
    classify\_b=Button(root,text="predict Vehicles",  
    command=lambda: classify(l),padx=10,pady=5)  
    classify\_b.configure(background='#364156', foreground='white',  
    font=('arial',10,'bold'))  
    classify\_b.place(relx=0.79,rely=0.46)  
  
  
show\_classify\_button()  
heading = Label(root, text="Traffic Signal Optimisation",pady=20, font=('arial',20,'bold'))  
  
heading.configure(background='#CDCDCD',foreground='#364156')  
heading.pack()  
  
  
root.mainloop()