import tflite\_runtime.interpreter as tflite

import os

import argparse

import cv2

import numpy as np

import sys

import time

from threading import Thread

import importlib.util

parser = argparse.ArgumentParser()

parser.add\_argument('--model', help='model dosya yolu',

default='model.tflite')

parser.add\_argument('--labels', help='etiketlerin dosya yolu',

default='label\_map.txt')

parser.add\_argument('--video', help='video dosya yolu',

default='test.mp4')

parser.add\_argument('--threshold', help='minimum doğruluk oranı',

default=0.5)

args = parser.parse\_args()

# model yolu değişkene atanıyor

PATH\_TO\_MODEL\_DIR = args.model

# etiket yolu değişkene atanıyor

PATH\_TO\_LABELS = args.labels

# video yolu değişkene atanıyor

VIDEO\_PATH = args.video

# minimum doğruluk oranı float tipinde değişkene atanıyor

MIN\_CONF\_THRESH = float(args.threshold)

import time

print('Model yükleniyor...', end='')

start\_time = time.time()

# TFLite model yükleniyor

interpreter = tflite.Interpreter(model\_path=PATH\_TO\_MODEL\_DIR)

# Etiketler yükleniyor

with open(PATH\_TO\_LABELS, 'r') as f:

labels = [line.strip() for line in f.readlines()]

end\_time = time.time()

elapsed\_time = end\_time - start\_time

print('Tamamlandı! İşlemler {} saniye sürdü'.format(elapsed\_time))

interpreter.allocate\_tensors()

input\_details = interpreter.get\_input\_details()

output\_details = interpreter.get\_output\_details()

height = input\_details[0]['shape'][1]

width = input\_details[0]['shape'][2]

floating\_model = (input\_details[0]['dtype'] == np.float32)

input\_mean = 127.5

input\_std = 127.5

# Saniyede görütülenen kare hız işlemi başlatılıyor

frame\_rate\_calc = 1

freq = cv2.getTickFrequency()

print('Model çalışması {} saniye sürdü... '.format(VIDEO\_PATH), end='')

# Video başlatılıyor

video = cv2.VideoCapture(VIDEO\_PATH)

imW = video.get(cv2.CAP\_PROP\_FRAME\_WIDTH)

imH = video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)

counter=1

fileobj = open('test1.txt', 'a')

while(video.isOpened()):

# FPS(Saniyede işlenen görüntü sayısı için zaman başlatıldı.

t1 = cv2.getTickCount()

# Videodan bir kare alınıyor

ret, frame1 = video.read()

# Resmi veri girişi için gerekli formata çeviriyoruz.

frame = frame1.copy()

frame\_rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

frame\_resized = cv2.resize(frame\_rgb, (width, height))

input\_data = np.expand\_dims(frame\_resized, axis=0)

if floating\_model:

input\_data = (np.float32(input\_data) - input\_mean) / input\_std

# Tespit çalışması için gelen resmi işleme alıyoruz.

interpreter.set\_tensor(input\_details[0]['index'],input\_data)

interpreter.invoke()

# Tespit sonuçlarını alıyoruz.

boxes = interpreter.get\_tensor(output\_details[0]['index'])[0] # Tespit kutusu

classes = interpreter.get\_tensor(output\_details[1]['index'])[0] # Tespit sınıfı

scores = interpreter.get\_tensor(output\_details[2]['index'])[0] # Doğrulu oranı

# Eğer doğruluk oranı istenilenin üstünde ise tüm algılanan nesneler için tespit kutusunu çiziyoruz.

for i in range(len(scores)):

if ((scores[i] > MIN\_CONF\_THRESH) and (scores[i] <= 1.0)):

# Tespit kutusunun koordinatları alınıp çizim yapılıyor

ymin = int(max(1,(boxes[i][0] \* imH)))

xmin = int(max(1,(boxes[i][1] \* imW)))

ymax = int(min(imH,(boxes[i][2] \* imH)))

xmax = int(min(imW,(boxes[i][3] \* imW)))

cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

# Etiketi yazdırıyoruz.

object\_name = labels[int(classes[i])] # etiketi etiket dosyamızdan çekiyoruz.

label = '%s: %d%%' % (object\_name, int(scores[i]\*100)) # Örnek: 'eagle: 72%'

labelSize, baseLine = cv2.getTextSize(label, cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, 2) # Font boyutu ayarlanıyor

label\_ymin = max(ymin, labelSize[1] + 10) # label konumunu düzenliyoruz

cv2.rectangle(frame, (xmin, label\_ymin-labelSize[1]-10), (xmin+labelSize[0], label\_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # etiket için beyaz bir kutu çiziyoruz

cv2.putText(frame, label, (xmin, label\_ymin-7), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 0), 2) # etiketi yazdırıyoruz

fileobj.write("Id:{} Kuşun İsmi:{} Doğruluk Oranı:{} \n".format(counter,object\_name,int(scores[i]\*100))

uzanti='jpg'

isim='{0}{1}.{2}'.format(object\_name,counter,uzanti)

print(isim)

cv2.imwrite(isim, frame)

counter=counter+1

# FPS i yazdırıyoruz

cv2.putText(frame,'FPS: {0:.2f}'.format(frame\_rate\_calc),(15,25),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,255,55),2,cv2.LINE\_AA)

# FPS, etiket ve tespit kutusu ile birlikte resmi bastırıyoruz.

cv2.imshow('Object Detector', frame)

# Calculate framerate

t2 = cv2.getTickCount()

time1 = (t2-t1)/freq

frame\_rate\_calc= 1/time1

# Press 'q' to quit

if cv2.waitKey(1) == ord('q'):

break

# Clean up

cv2.destroyAllWindows()

print("Done")