import numpy as np

import time

import cv2

import os

import imutils

import subprocess

from gtts import gTTS

from pygame import mixer

LABELS = open("yolo-coco/coco.names").read().strip().split("\n")

net = cv2.dnn.readNetFromDarknet("yolo-coco/yolov3.cfg", "yolo-coco/yolov3.weights")

ln = net.getLayerNames()

ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

mixer.init()

cap = cv2.VideoCapture(0)

frame\_count = 0

start = time.time()

first = True

frames = []

ii=0

while(True):

frame\_count += 1

ret, frame = cap.read()

frame = cv2.flip(frame,1)

frames.append(frame)

cv2.imshow('video',frame)

if frame\_count ==1000000000:

break

if ret:

key = cv2.waitKey(1)

if frame\_count % 60 == 0:

end = time.time()

(H, W) = frame.shape[:2]

blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),

swapRB=True, crop=False)

net.setInput(blob)

layerOutputs = net.forward(ln)

boxes = []

confidences = []

classIDs = []

centers = []

for output in layerOutputs:

for detection in output:

scores = detection[5:]

classID = np.argmax(scores)

confidence = scores[classID]

if confidence > 0.5:

box = detection[0:4] \* np.array([W, H, W, H])

(centerX, centerY, width, height) = box.astype("int")

x = int(centerX - (width / 2))

y = int(centerY - (height / 2))

boxes.append([x, y, int(width), int(height)])

confidences.append(float(confidence))

classIDs.append(classID)

centers.append((centerX, centerY))

idxs = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.3)

texts = []

if len(idxs) > 0:

for i in idxs.flatten():

centerX, centerY = centers[i][0], centers[i][1]

if centerX <= W/3:

W\_pos = "left "

elif centerX <= (W/3 \* 2):

W\_pos = "center "

else:

W\_pos = "right "

if centerY <= H/3:

H\_pos = "top "

elif centerY <= (H/3 \* 2):

H\_pos = "mid "

else:

H\_pos = "bottom "

texts.append(H\_pos + W\_pos + LABELS[classIDs[i]])

print(texts)

if texts:

ii=ii+1

description = ', '.join(texts)

tts = gTTS(description, lang='en')

tts.save('tts'+str(ii)+'.mp3')

mixer.music.load('tts'+str(ii)+'.mp3')

mixer.music.play()

cap.release()

cv2.destroyAllWindows()

os.remove("tts.mp3")