D20IT159 - Anadani Riya

D20IT156 - Doshi Heet

CODE:

import numpy as np

import cv2

import imutils

import os

import time

def Check(a, b):

dist = ((a[0] - b[0]) \*\* 2 + 550 / ((a[1] + b[1]) / 2) \* (a[1] - b[1]) \*\* 2) \*\* 0.5

calibration = (a[1] + b[1]) / 2

if 0 < dist < 0.25 \* calibration:

return True

else:

return False

def Setup(yolo):

global net, ln, LABELS

weights = os.path.sep.join([yolo, "yolov3.weights"])

config = os.path.sep.join([yolo, "yolov3.cfg"])

labelsPath = os.path.sep.join([yolo, "coco.names"])

LABELS = open(labelsPath).read().strip().split("\n")

net = cv2.dnn.readNetFromDarknet(config, weights)

ln = net.getLayerNames()

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

def ImageProcess(image):

global processedImg

(H, W) = (None, None)

frame = image.copy()

if W is None or H is None:

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

blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416), swapRB=True, crop=False)

net.setInput(blob)

starttime = time.time()

layerOutputs = net.forward(ln)

stoptime = time.time()

print("Video is Getting Processed at {:.4f} seconds per frame".format((stoptime-starttime)))

confidences = []

outline = []

for output in layerOutputs:

for detection in output:

scores = detection[5:]

maxi\_class = np.argmax(scores)

confidence = scores[maxi\_class]

if LABELS[maxi\_class] == "person":

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))

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

confidences.append(float(confidence))

box\_line = cv2.dnn.NMSBoxes(outline, confidences, 0.5, 0.3)

if len(box\_line) > 0:

flat\_box = box\_line.flatten()

pairs = []

center = []

status = []

for i in flat\_box:

(x, y) = (outline[i][0], outline[i][1])

(w, h) = (outline[i][2], outline[i][3])

center.append([int(x + w / 2), int(y + h / 2)])

status.append(False)

for i in range(len(center)):

for j in range(len(center)):

close = Check(center[i], center[j])

if close:

pairs.append([center[i], center[j]])

status[i] = True

status[j] = True

index = 0

for i in flat\_box:

(x, y) = (outline[i][0], outline[i][1])

(w, h) = (outline[i][2], outline[i][3])

if status[index] == True:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 150), 2)

elif status[index] == False:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

index += 1

for h in pairs:

cv2.line(frame, tuple(h[0]), tuple(h[1]), (0, 0, 255), 2)

processedImg = frame.copy()

create = None

frameno = 0

filename = "newVideo.mp4"

yolo = "yolov3/"

opname = "output2.avi"

cap = cv2.VideoCapture(filename)

time1 = time.time()

while(True):

ret, frame = cap.read()

if not ret:

break

current\_img = frame.copy()

current\_img = imutils.resize(current\_img, width=480)

video = current\_img.shape

frameno += 1

if(frameno%2 == 0 or frameno == 1):

Setup(yolo)

ImageProcess(current\_img)

Frame = processedImg

cv2.imshow("Image", Frame)

if create is None:

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

create = cv2.VideoWriter(opname, fourcc, 30, (Frame.shape[1], Frame.shape[0]), True)

create.write(Frame)

if cv2.waitKey(1) & 0xFF == ord('s'):

break

time2 = time.time()

print("Completed. Total Time Taken: {} minutes".format((time2-time1)/60))

cap.release()

cv2.destroyAllWindows()