**OBJECT DETECTION CODE**

#pip install pyttsx3

#pip install pywin32

#pip install opencv-python

#pip install opencv-contrib-python

#pip install imutils

import cv2

import imutils

import numpy as np

import pyttsx3

def tts():

engine = pyttsx3.init()

rate = engine.getProperty('rate')

engine.setProperty('rate', rate+0)

engine.say(label)

engine.runAndWait()

net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

classes = []

with open("coco.names", "r") as f:

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

layer\_names = net.getLayerNames()

output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()]

colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image

cap = cv2.VideoCapture(0)

while True:

ret,img=cap.read()

img = imutils.resize(img, width=600)

height, width, channels = img.shape

# Detecting objects

blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

net.setInput(blob)

outs = net.forward(output\_layers)

# Showing informations on the screen

class\_ids = []

confidences = []

boxes = []

for out in outs:

for detection in out:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5:

# Object detected

center\_x = int(detection[0] \* width)

center\_y = int(detection[1] \* height)

w = int(detection[2] \* width)

h = int(detection[3] \* height)

# Rectangle coordinates

x = int(center\_x - w / 2)

y = int(center\_y - h / 2)

boxes.append([x, y, w, h])

confidences.append(float(confidence))

class\_ids.append(class\_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

font = cv2.FONT\_HERSHEY\_PLAIN

for i in range(len(boxes)):

if i in indexes:

x, y, w, h = boxes[i]

label = str(classes[class\_ids[i]])

if(classes[class\_ids[i]] == classes[class\_ids[i]] ):

color = colors[i]

cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

cv2.putText(img, label, (x, y + 30), font, 3, color, 3)

print(label)

tts()

# show the output frame

cv2.imshow("Frame", img)

key = cv2.waitKey(33) & 0xFF

# if the `q` key was pressed, break from the loop

if key == ord("s"):

break

cv2.destroyAllWindows()