import cv2image = cv2.imread('https://github.com/Sruthi-Mohan/naanmudhalvan/blob/main/image.jpg')image = cv2.resize(image, (640, 480))h = image.shape[0]w = image.shape[1]weights = "https://github.com/Sruthi-Mohan/naanmudhalvan/blob/main/frozen\_inference\_graph%20(1).pb"model = " https://github.com/Sruthi-Mohan/naanmudhalvan/blob/main/ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt"net = cv2.dnn.readNetFromTensorflow(weights, model)class\_names = []

with open("https://github.com/Sruthi-Mohan/naanmudhalvan/blob/main/coco\_names.txt", "r") as f:

class\_names = f.read().strip().split("\n")

blob = cv2.dnn.blobFromImage( image, 1.0/127.5, (320, 320), [127.5, 127.5, 127.5])

net.setInput(blob)output = net.forward()

for detection in output[0, 0, :, :]:

probability = detection[2]

if probability < 0.5:

continue

box = [int(a \* b) for a, b in zip(detection[3:7], [w, h, w, h])]

box = tuple(box)

cv2.rectangle(image, box[:2], box[2:], (0, 255, 0), thickness=2)

class\_id = int(detection[1])

label = f"{class\_names[class\_id - 1].upper()} {probability \* 100:.2f}%"

cv2.putText(image, label, (box[0], box[1] + 15), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)cv2\_imshow('output',image)

cv2.waitKey()