import numpy as np

import cv2

import tensorflow as tf

# Load the pre-trained model

model\_path = 'ssd\_mobilenet\_v2.pb' # Replace with the path to your model

model = tf.saved\_model.load(model\_path)

# Load the labels

labels\_path = 'label\_map.pbtxt' # Replace with the path to your labels file

with open(labels\_path, 'r') as f:

labels = {int(line.split(': ')[1]): line.split(': ')[0] for line in f.readlines()}

# Function to perform object detection

def detect\_objects(image, model):

input\_tensor = tf.convert\_to\_tensor(image)

input\_tensor = input\_tensor[tf.newaxis,...]

detections = model(input\_tensor)

return detections

# Function to draw bounding boxes

def draw\_boxes(image, detections):

height, width, \_ = image.shape

for i in range(int(detections.pop('num\_detections'))):

score = detections['detection\_scores'][i].numpy()

if score < 0.5:

continue

box = detections['detection\_boxes'][i].numpy()

ymin, xmin, ymax, xmax = box

left, right, top, bottom = (xmin \* width, xmax \* width,

ymin \* height, ymax \* height)

label = labels[int(detections['detection\_classes'][i].numpy())]

cv2.rectangle(image, (int(left), int(top)), (int(right), int(bottom)), (255, 0, 0), 2)

cv2.putText(image, label, (int(left), int(top) - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255, 0, 0), 2)

return image

# Load an image

image\_path = 'path\_to\_your\_image.jpg' # Replace with the path to your image

image = cv2.imread(image\_path)

# Perform object detection

detections = detect\_objects(image, model)

# Draw bounding boxes

output\_image = draw\_boxes(image, detections)

# Display the image

cv2.imshow('Object Detection', output\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()