Object detection using SSD:

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread('image.jpg')

image = cv2.resize(image, (640, 480))

h = image.shape[0]

w = image.shape[1]

# path to the weights and model files

weights = "frozen\_inference\_graph.pb"

model = "ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt"

# load the MobileNet SSD model trained  on the COCO dataset

net = cv2.dnn.readNetFromTensorflow(weights, model)

# load the class labels the model was trained on

class\_names = []

with open("coco\_names.txt", "r") as f:

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

# create a blob from the image

blob = cv2.dnn.blobFromImage(

    image, 1.0/127.5, (320, 320), [127.5, 127.5, 127.5])

# pass the blog through our network and get the output predictions

net.setInput(blob)

output = net.forward()  # shape: (1, 1, 100, 7)

# loop over the number of detected objects

for detection in output[0, 0, :, :]:  # output[0, 0, :, :] has a shape of: (100, 7)

    # the confidence of the model regarding the detected object

    probability = detection[2]

    # if the confidence of the model is lower than 50%,

    # we do nothing (continue looping)

    if probability < 0.5:

        continue

    # perform element-wise multiplication to get

    # the (x, y) coordinates of the bounding box

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

    box = tuple(box)

    # draw the bounding box of the object

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

    # extract the ID of the detected object to get its name

    class\_id = int(detection[1])

    # draw the name of the predicted object along with the probability

    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(image)

cv2.waitKey()