```
            net.setInput(blob)
            detections = net.forward()

            for i in np.arange(0, detections.shape[2]):
                confidence = detections[0, 0, i, 2]

                confidence_level = 0.7

                if confidence > confidence_level:
                    # extract the index of the class label from the `detections`, then
compute the (x, y)-coordinates of
                    # the bounding box for the object
                    idx = int(detections[0, 0, i, 1])
                    box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                    (startX, startY, endX, endY) = box.astype("int")
                    # draw the prediction on the frame
                    label = "{}: {:.2f}%".format(CLASSES[idx],
                                                 confidence * 100)
                    cv2.rectangle(frame, (startX, startY), (endX, endY),
                                  COLORS[idx], 2)
                    y = startY - 15 if startY - 15 > 15 else startY + 15
                    cv2.putText(frame, label, (startX, y),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5,
COLORS[idx], 2)

            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 0), 2)
            # Start tracker
            now = datetime.now()
            if differ == None or differ > 9:
                tracker.init(frame, initBB2)
                fps = FPS().start()

    # check to see if we are currently tracking an object, if so, ignore other boxes
    # this code is relevant if we want to identify particular persons (section 2 of
this tutorial)
    if initBB2 is not None:

        # grab the new bounding box coordinates of the object
        (success, box) = tracker.update(frame)

        # check to see if the tracking was a success
        differ = 10
        if success:
            (x, y, w, h) = [int(v) for v in box]
```