```
                index = key_list[val_list.index(class_name)]
                # Structure data, that we could use it with our draw_bbox function
                tracked_bboxes.append(bbox.tolist() + [tracking_id, index])

        # draw detection on frame
        image = draw_bbox(original_frame, tracked_bboxes,
                          CLASSES=CLASSES, tracking=True)

        t3 = time.time()
        times.append(t2-t1)
        times_2.append(t3-t1)

        times = times[-20:]
        times_2 = times_2[-20:]

        ms = sum(times)/len(times)*1000
        fps = 1000 / ms
        fps2 = 1000 / (sum(times_2)/len(times_2)*1000)

        image = cv2.putText(image, "Time: {:.1f} FPS".format(
            fps), (0, 30), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0,
255), 2)

        # draw original yolo detection
        #image = draw_bbox(image, bboxes, CLASSES=CLASSES,
show_label=False, rectangle_colors=rectangle_colors, tracking=True)

        print("Time: {:.2f}ms, Detection FPS: {:.1f}, total FPS: {:.1f}".format(
            ms, fps, fps2))
        # if output_path != '':
        #       out.write(image)
        if show:
            cv2.imshow('output', image)

            if cv2.waitKey(25) & 0xFF == ord("q"):
                cv2.destroyAllWindows()
                break

    cv2.destroyAllWindows()

yolo = Load_Yolo_model()
# Object_tracking(yolo, video_path, "track.mp4",
input_size=YOLO_INPUT_SIZE, show=False, iou_threshold=0.1,
rectangle_colors=(255,0,0), Track_only = ["person"])
```