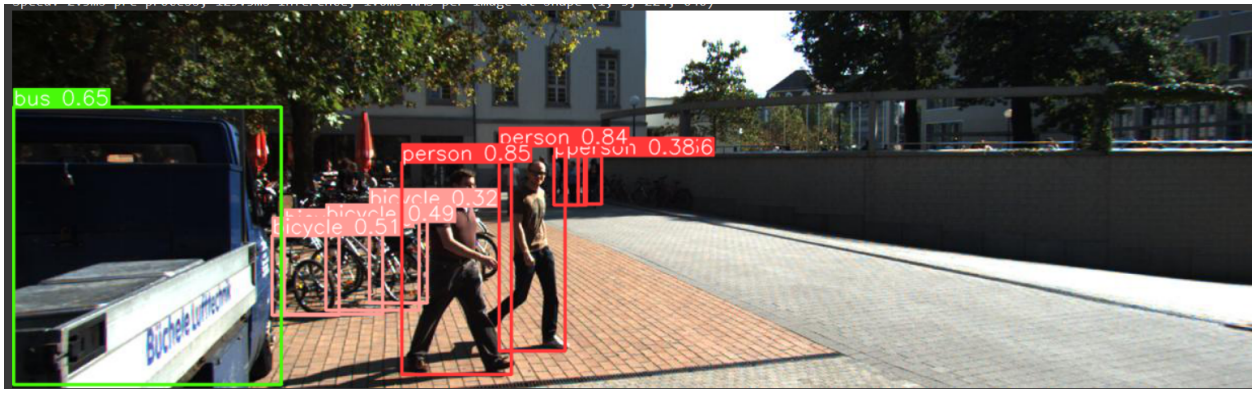
Question 1
In this assignment, we were asked to make a video of the object tracker.
For this, we are using a pre-trained YOLO model to detect, along with a SORT algorithm to track.
To implement this, we need to do two steps: first, detect the object.

Detection

In detection, we use the YOLO model (You only look once). We have imported the pre-trained model. Next, we will start send you images to YOLO and it will send us all the detections.
An example of detections is:

```
image 1/1: 370x1224 5 persons, 4 bicycles, 1 bus
Speed: 2.5ms pre-process, 129.3ms inference, 1.0ms NMS per image at shape (1, 3, 224, 640)
```



Now we only need to have the detections that are people. For this, I printed the xyxy coordinates of detections and took only the ones with class 0. Class 0 is the people class

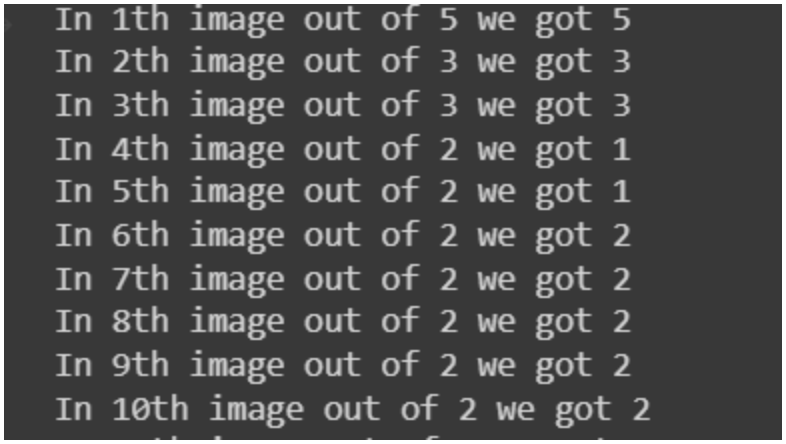
```
for guess in det.xyxy:
    for i in guess:
      if(i[5] == 0):
        img = cv2.rectangle(img, (int(i[0]),int(i[1])), (int(i[2]), int(i[3])), (255, 0, 0))
        new_det.append([i[0],i[1],i[2], i[3], i[4]])
```

After that, the code will detect these people:

Tracking

In this step, we will send the detections to the tracker, and it will try to track the objects. From the detection we had we will send them to SORT, and it will return the coordinates along with the unique ID for the objects it is tracking.

We can use this ID to track people. Let's see how many people our tracker was able to find:



It can track 5 out of 5 people in the first image, but like in image 4, it is not able to capture everyone. Out of 2 people, it only captured one person.
We will use these unique IDs to give colour to each ID. As there are utmost 10 people in a frame, I decided to make 10 unique colours and assign the value to people accordingly.

PART 2

In this part we were told to run the evaluation on the code. For this we use the trackerEval and MOT15 benchmark.
In this, you can see that they want a very specific file structure to run the code. For this we will follow the structure that they asked for.
It started my making a seqmaps
Seq maps are in the form of:
File name: MOT15-train (it could be test or all as well. The names can be chosen depending on purpose)
Content:

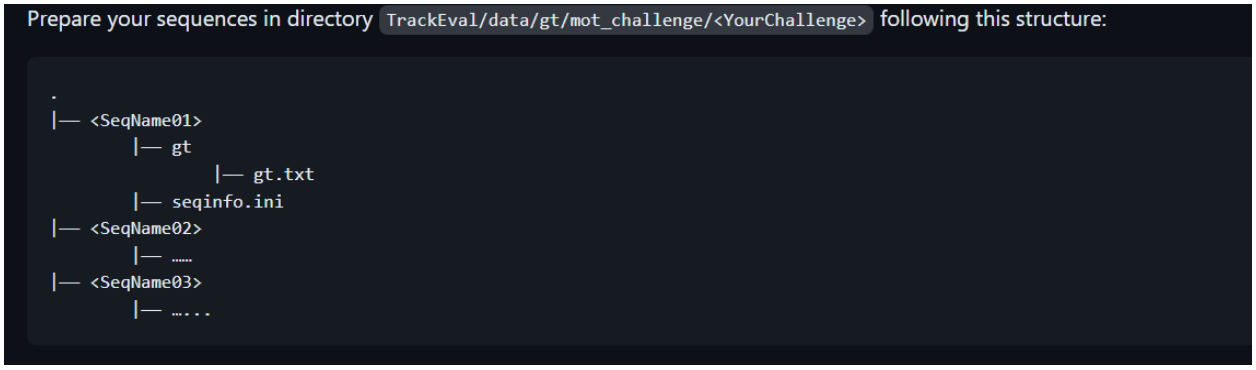
name
MOT15

Next we need a seqinfo.in

Name: seqinfo.in
Content:
[Sequence]
name=MOT15
frameRate=30
seqLength=145
imWidth=370
imHeight=1244
imExt=.jpg

Imwidth and imheight can be found from the given dataset using all_images[0].shape.

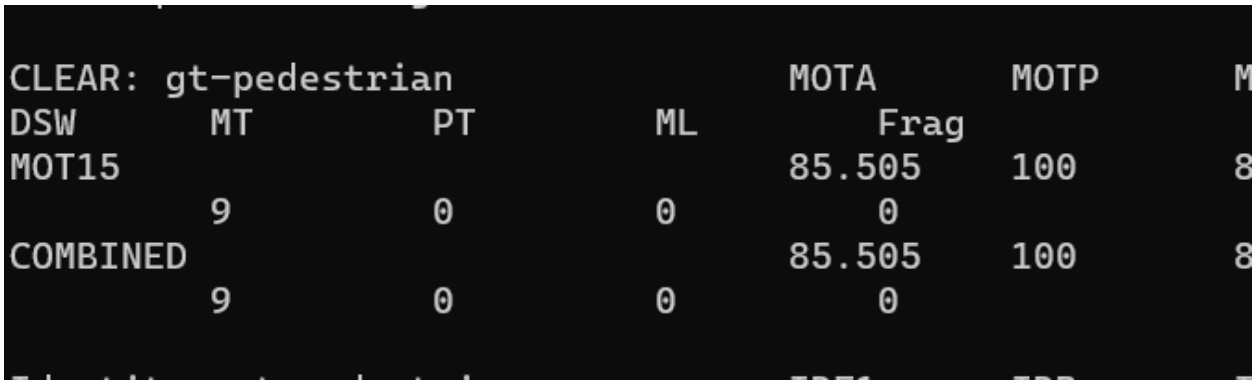
The file structure is hard to explain but it follows this structure



Where YourChallenge is MOT15. Moreover there is a dataset available on the net that shows the file structure (linked in the github repo given to us)

Using that we run the command
python .\TrackEval-master\scripts\run_mot_challenge.py --BENCHMARK MOT15 --SPLIT_TO_EVAL train --TRACKERS_TO_EVAL gt --METRICS CLEAR Identity --GT_FOLDER "C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\gt\mot_challenge" --TRACKERS_FOLDER "C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\tracker\mot_challenge"

To get the matrix. Not setting the evaluation matrix to the ground truth we get



And running on our dataset we get

```
CLEAR: gt-pedestrian                    MOTA      MOTP
DSW       MT        PT        ML          Frag
MOT15                                   40.264    80.409
0         1         8         0           49
COMBINED                                40.264    80.409
0         1         8         0           49
```

The whole screenshot is

```
PS C:\Users\spars\Desktop\sem6\cv\Homework3> python .\TrackEval-master\scripts\run_mot_challenge.py --BENCHMARK MOT15 --SPLIT_TO_EVAL train --TRACKERS_TO_EVAL gt  --METRICS CLEAR Identity --GT_FOLDER "C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\gt\mot_challenge" --TRACKERS_FOLDER "C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\tracker\mot_challenge"
Error importing BURST due to missing underlying dependency: No module named 'tabulate'

Eval Config:
USE_PARALLEL         : False
NUM_PARALLEL_CORES   : 8
BREAK_ON_ERROR       : True
RETURN_ON_ERROR      : False
LOG_ON_ERROR         : C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\error_log.txt
PRINT_RESULTS        : True
PRINT_ONLY_COMBINED  : False
PRINT_CONFIG         : True
TIME_PROGRESS        : True
DISPLAY_LESS_PROGRESS : False
OUTPUT_SUMMARY       : True
OUTPUT_EMPTY_CLASSES : True
OUTPUT_DETAILED      : True
PLOT_CURVES          : True

MotChallenge2DBox Config:
PRINT_CONFIG         : True
GT_FOLDER            : C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\gt\mot_challenge
TRACKERS_FOLDER      : C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\tracker\mot_challenge
OUTPUT_FOLDER        : None
TRACKERS_TO_EVAL     : ['gt']
CLASSES_TO_EVAL      : ['pedestrian']
BENCHMARK            : MOT15
SPLIT_TO_EVAL        : train
INPUT_AS_ZIP         : False
DO_PREPROC           : True
TRACKER_SUB_FOLDER   : data
OUTPUT_SUB_FOLDER    :
TRACKER_DISPLAY_NAMES : None
SEQMAP_FOLDER        : None
SEQMAP_FILE          : None
SEQ_INFO             : None
```

```
MotChallenge2DBox Config:
PRINT_CONFIG         : True
GT_FOLDER            : C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\gt\mot_challenge
TRACKERS_FOLDER      : C:\Users\spars\Desktop\sem6\cv\Homework3\TrackEval-master\data\tracker\mot_challenge
OUTPUT_FOLDER        : None
TRACKERS_TO_EVAL     : ['gt']
CLASSES_TO_EVAL      : ['pedestrian']
BENCHMARK            : MOT15
SPLIT_TO_EVAL        : train
INPUT_AS_ZIP         : False
DO_PREPROC           : True
TRACKER_SUB_FOLDER   : data
OUTPUT_SUB_FOLDER    :
TRACKER_DISPLAY_NAMES : None
SEQMAP_FOLDER        : None
SEQMAP_FILE          : None
SEQ_INFO             : None
GT_LOC_FORMAT        : {gt_folder}/{seq}/gt/gt.txt
SKIP_SPLIT_FOL       : False

CLEAR Config:
METRICS              : ['CLEAR', 'Identity']
THRESHOLD            : 0.5
PRINT_CONFIG         : True

Identity Config:
METRICS              : ['CLEAR', 'Identity']
THRESHOLD            : 0.5
PRINT_CONFIG         : True

Evaluating 1 tracker(s) on 1 sequence(s) for 1 class(es) on MotChallenge2DBox dataset using the following metrics: CLEAR, Identity, Count


Evaluating gt

    MotChallenge2DBox.get_raw_seq_data(gt, MOT15)                       0.1848 sec
    MotChallenge2DBox.get_preprocessed_seq_data(pedestrian)             0.1451 sec
    CLEAR.eval_sequence()                                               0.0511 sec
```

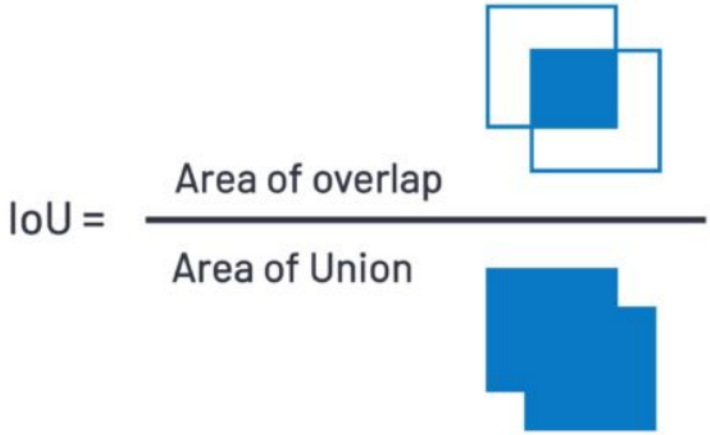
```
Evaluating gt

    MotChallenge2DBox.get_raw_seq_data(gt, MOT15)                       0.1848 sec
    MotChallenge2DBox.get_preprocessed_seq_data(pedestrian)             0.1451 sec
    CLEAR.eval_sequence()                                               0.0511 sec
    Identity.eval_sequence()                                            0.0142 sec
    Count.eval_sequence()                                               0.0000 sec
1 eval_sequence(MOT15, gt)                                             0.3991 sec

All sequences for gt finished in 0.40 seconds

CLEAR: gt-pedestrian              MOTA      MOTP      MODA      CLR_Re    CLR_Pr    MTR       PTR       MLR       sMOTA     CLR_TP    CLR_FN    CLR_FP    I
DSW      MT        PT        ML       Frag
MOT15                             40.264    80.409    41.728    68.082    72.093    11.111    88.889    0         26.926    465       218       180       1
0        1         8         0        49
COMBINED                          40.264    80.409    41.728    68.082    72.093    11.111    88.889    0         26.926    465       218       180       1
0        1         8         0        49

Identity: gt-pedestrian           IDF1      IDR       IDP       IDTP      IDFN      IDFP
MOT15                             63.705    61.933    65.581    423       260       222
COMBINED                          63.705    61.933    65.581    423       260       222

Count: gt-pedestrian              Dets      GT_Dets   IDs       GT_IDs
MOT15                             645       683       35        9
COMBINED                          645       683       35        9

Timing analysis:
MotChallenge2DBox.get_raw_seq_data                                      0.1848 sec
MotChallenge2DBox.get_preprocessed_seq_data                             0.1451 sec
CLEAR.eval_sequence                                                     0.0511 sec
Identity.eval_sequence                                                  0.0142 sec
Count.eval_sequence                                                     0.0000 sec
eval_sequence                                                           0.3991 sec
Evaluator.evaluate                                                      0.4123 sec
PS C:\Users\spars\Desktop\sem6\cv\Homework3>
```

Looking at the score we can see that the MOTP and MOTA scores.
MOTA (Multiple Object Tracking Accuracy) measures the overall accuracy of the tracker in terms of detection, tracking, and false positives.
It is computed as the percentage of ground truth trajectories that have been correctly tracked by the algorithm minus the percentage of false positives (i.e., detections that are not part of any ground truth trajectory) and the percentage of missed targets (i.e., ground truth targets that were not detected or tracked).
The formula for it is:

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t GT_t}$$

Where FNt is false negative, FPt is false positive, and IDSt is the mismatch error and GT is ground truth.
A higher MOTA represents a better tracking performance. In our case, when we run it for Ground truth vs ground truth, we get **an 85 per cent match**. And running on our case, we get a **41 per cent match**.
That means our algorithm is not able to collect some images that is why it's giving alow match.

MOTP (Multiple Object Tracking Precision) measures the average distance between the predicted and ground truth bounding boxes. This uses the idea of Intersection over Union (IOU) to compute the average overlap between the predicted bounding boxes and their corresponding ground truth bounding boxes.
The formula for IOU is:



Ans the formula to calculate MOTP is:

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}.$$

Where,
dt – Distance between the localisation of objects in the ground truth and the detection output
ct – total matches made between ground truth and the detection output

Now we get a 100 per cent overlap on the ground truth, which makes sense as we are technically comparing the same boxes. When running on our case, we are able to score an 80 per cent accuracy, which is good. That means 80 per cent of the area of the rectangle we were able to track.

A better MOTP means better-localized predictions