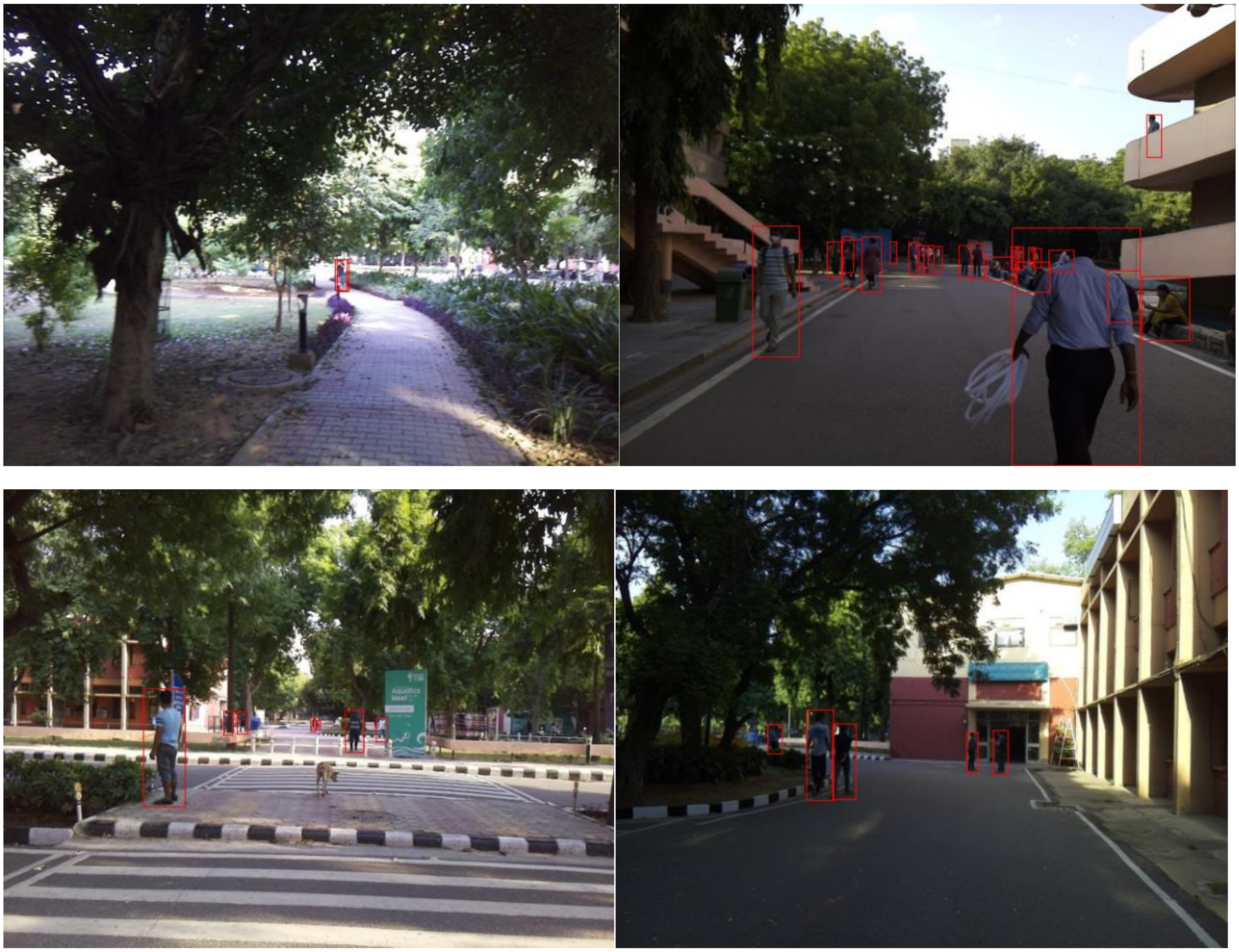
CV_Intern_Visionlab_IITD_Assignment_Sep_2024

This report outlines the experiments and results of the assignment given by IITD. It involved the evaluation, analysis and fine tuning of the DINO object detection model on a pedestrian dataset containing 200 images.

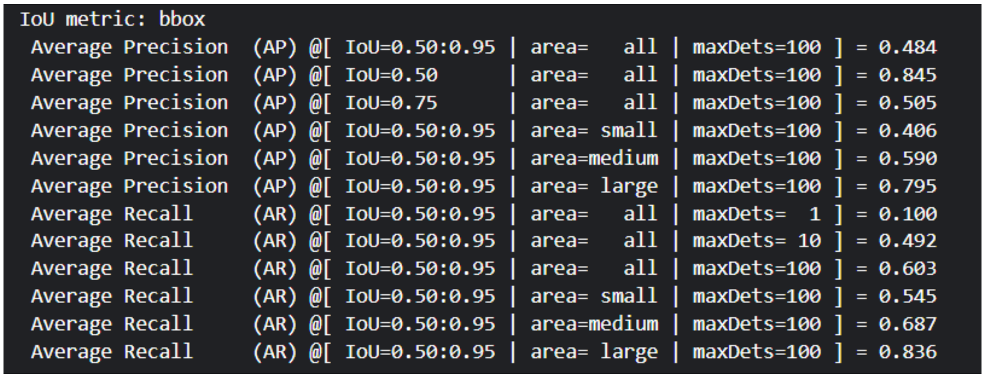
Ground Truth Visualization



LECTURE HALL COMPLEX
INTER IIT
Aquatics Meet
Breaststroke
50M, 100M, 200M
INTER-IIT AQUATIC

Inference and Evaluation on pre-trained weights

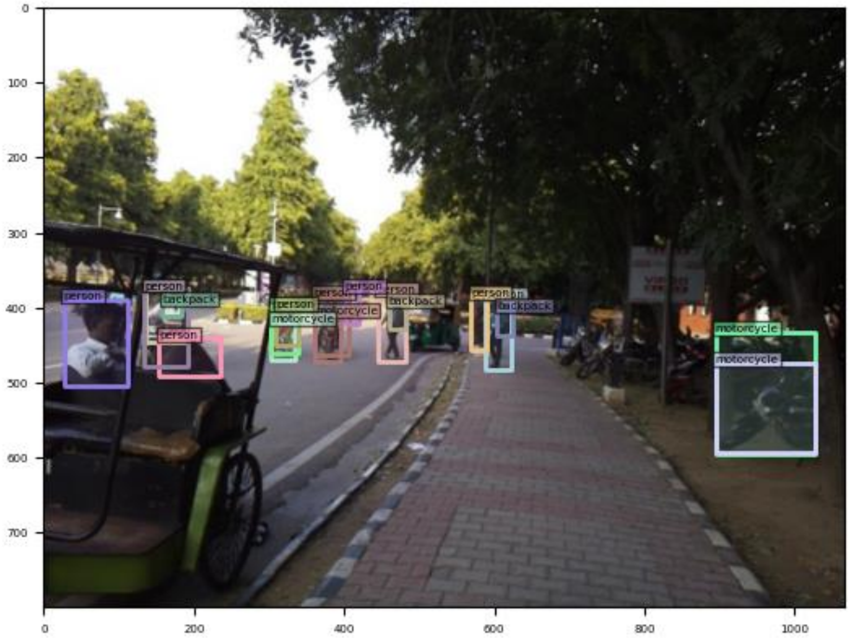
Bounding box AP values with pre-trained model:



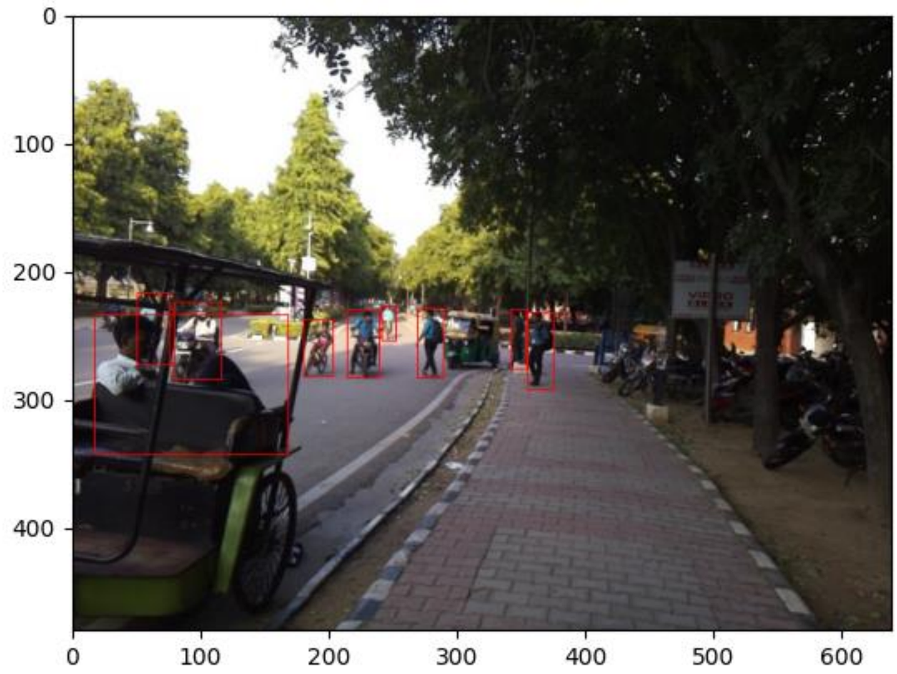
```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.484
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.845
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.505
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.406
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.590
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.795
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.100
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.492
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.603
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.545
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.687
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.836
```

Analysis

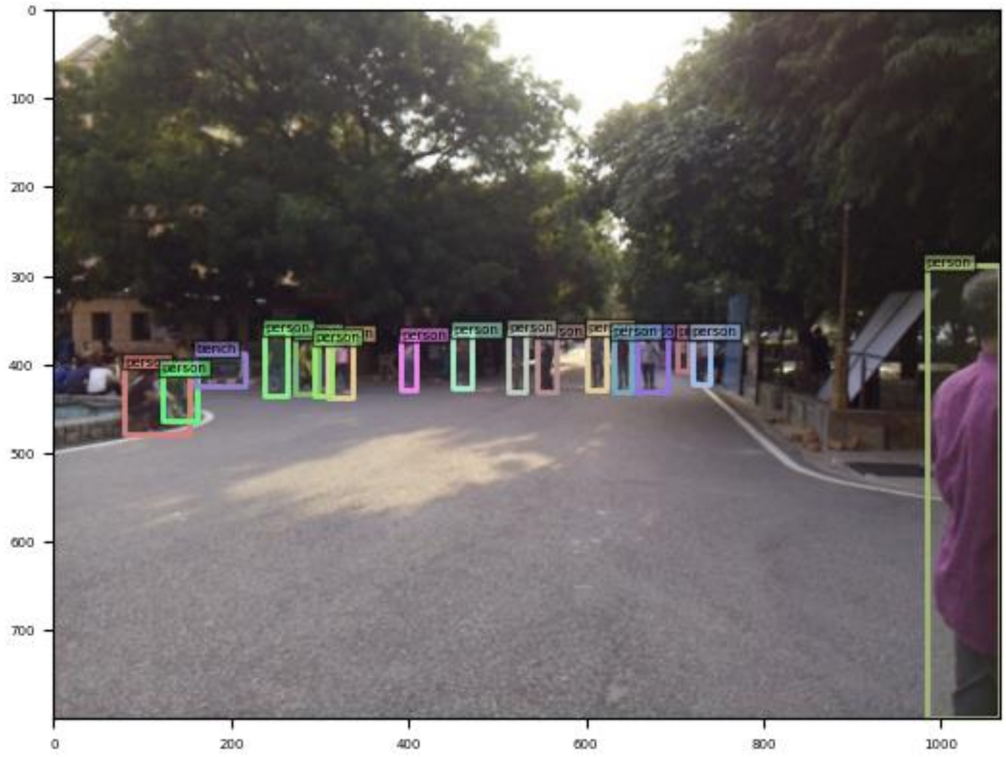
Inference with pre-trained model (checkpoint0033_4scale.pth):



Ground truth:



- Results are fairly consistent with the ground truth. Larger objects are detected with more average precision while smaller objects with lesser AP.
- False negatives for people who are sitting and are farther away (see middle left)



Errors encountered

- DINO_train.sh and DINO_eval.sh needed specific versions of numpy and yapf to work. Solved it by “pip install numpy==1.23.5 yapf==0.40.1”
- While setup, test.py gave CUDA out of memory error for dimensions higher than 71 in the following code:

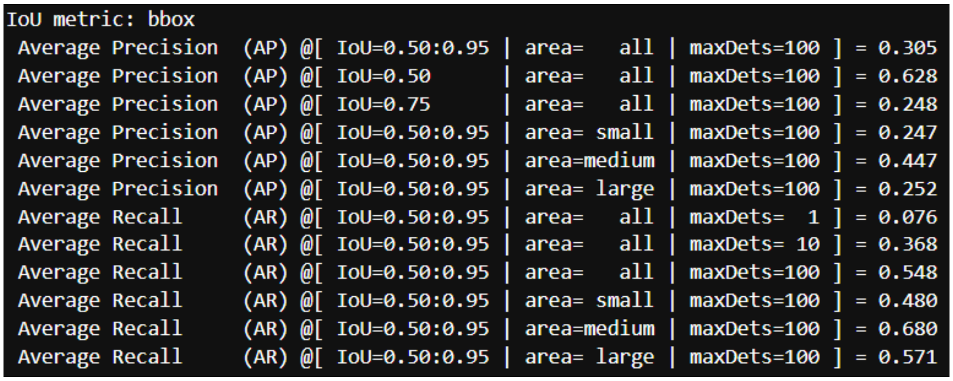
  ```
  for channels in [30, 32, 64, 71, 1025, 2048, 3096]:
          check_gradient_numerical(channels, True, True, True)
  ```

  Since it was only a testing code, I removed dimensions >= 1025

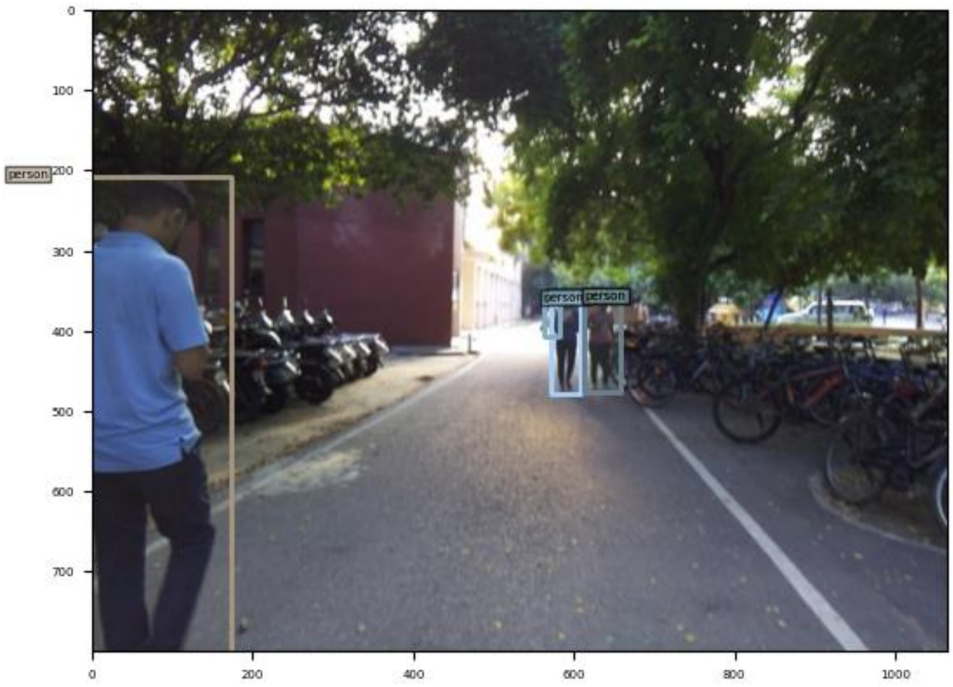
- Few config errors that were resolved by correcting the params (num_classes, dn_labelbook_size, etc)
- When trying to add the –save_results flag in DINO_train.sh and DINO_eval.sh, it threw a RuntimeError: Sizes of tensors must match except in dimension 1. This error was caused in engine.py within the “if arg.save_results:” block. Couldn’t resolve it in time.

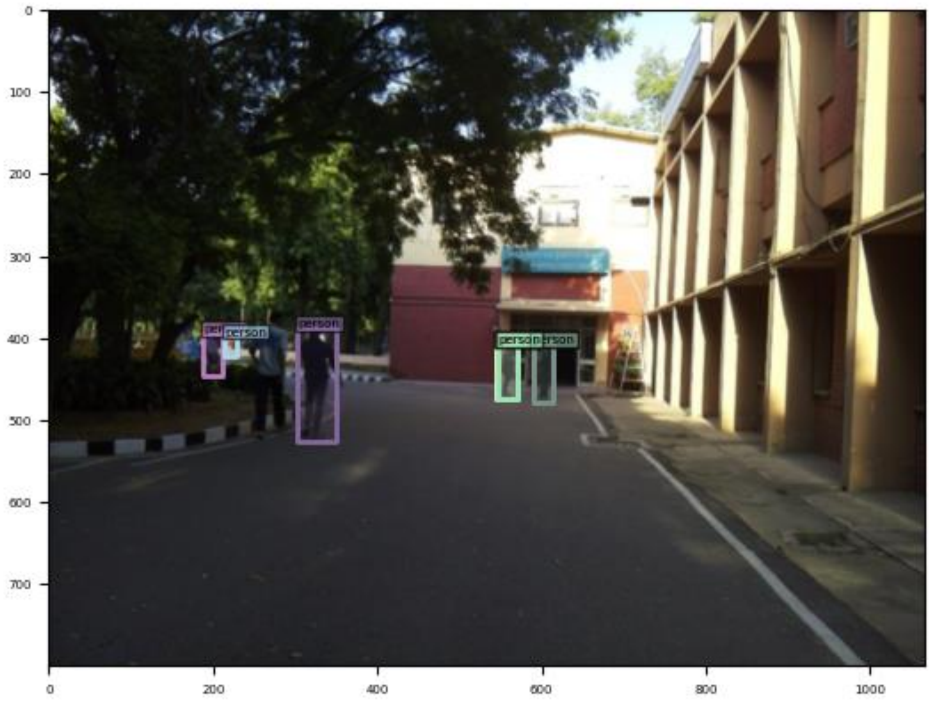
Inference and Evaluation on fine-tuned weights

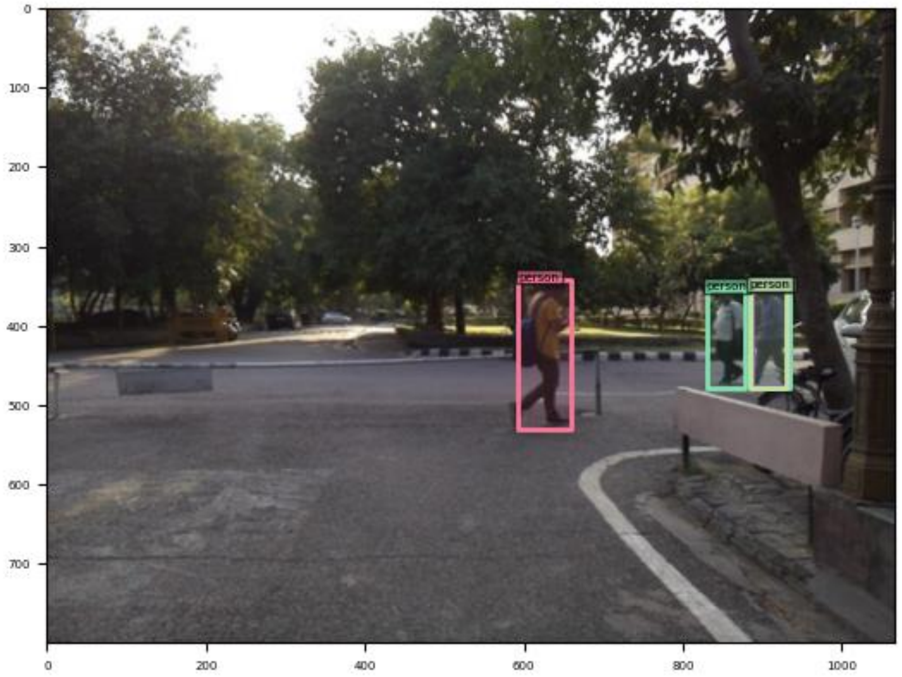
- lr=0.00025
- epochs=30



```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.305
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.628
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.248
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.247
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.447
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.252
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.368
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.548
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.480
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.680
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.571
```

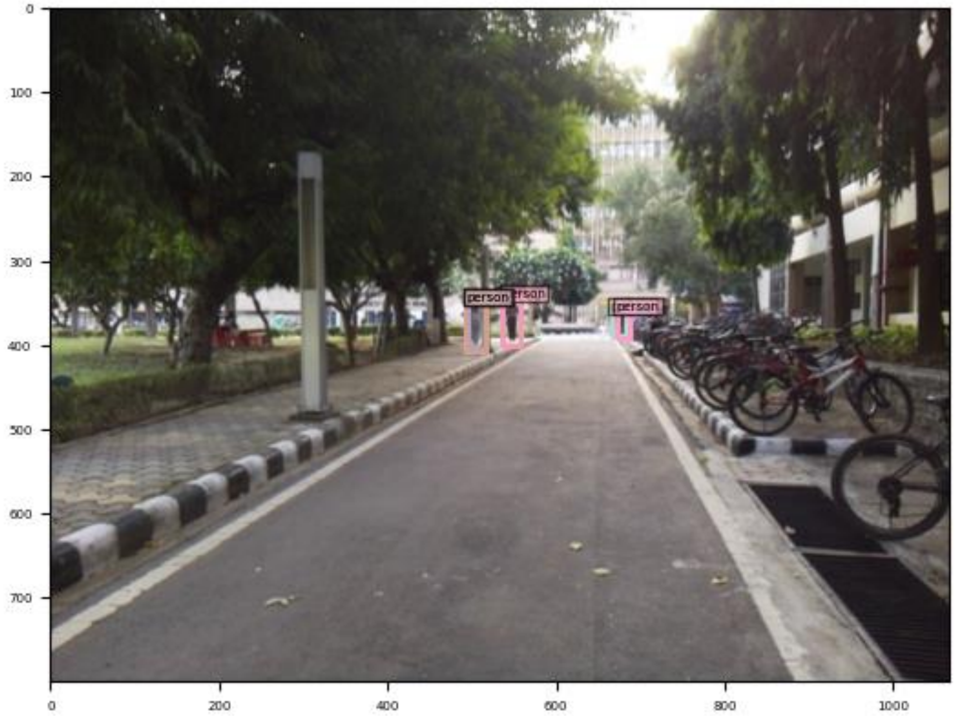


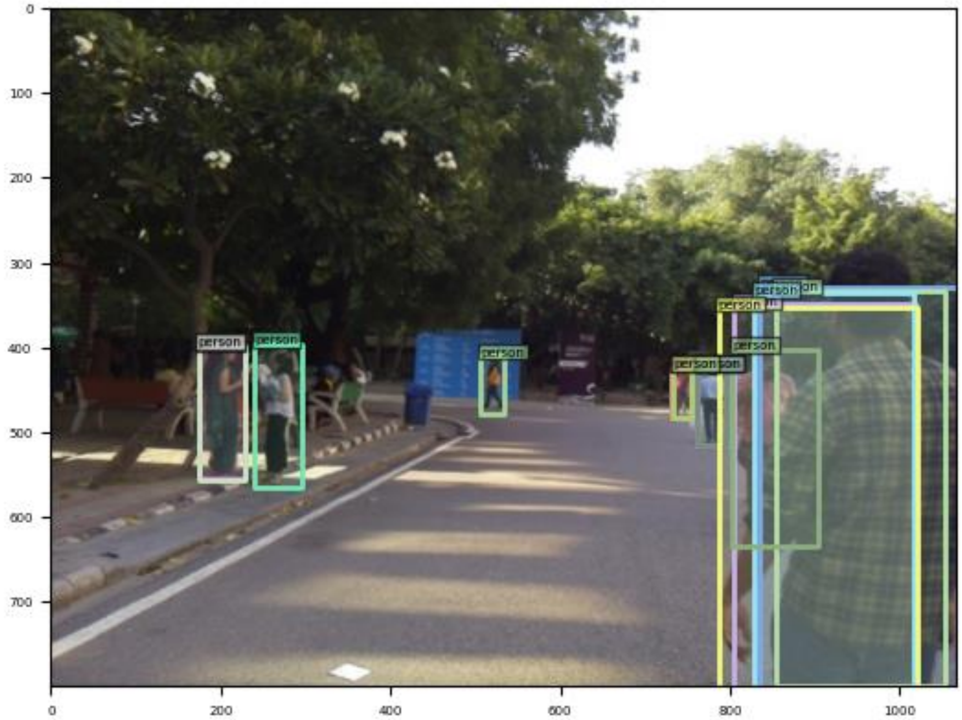

0
100
200
300
400
500
600
700
person
person
person
person
person
0
200
400
600
800
1000


0
100
200
300
400
500
600
700
person
person
person
0
200
400
600
800
1000

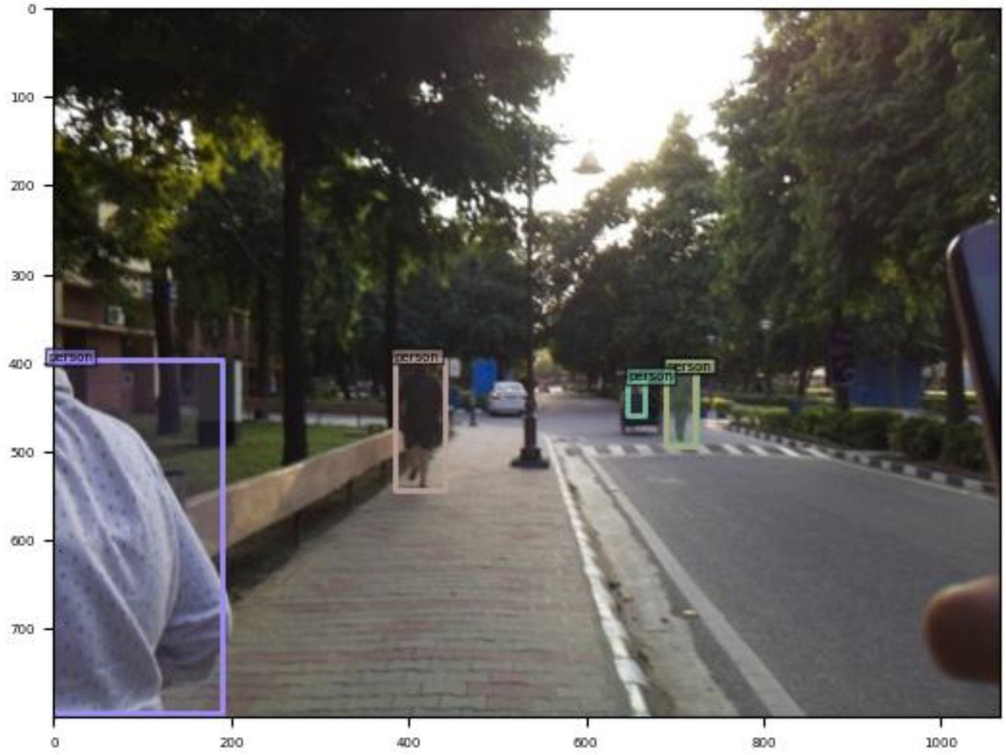
Observations:

- Lots of false positives (overlapping bounding boxes) for small and large objects. Possibly, due to IOU threshold value.

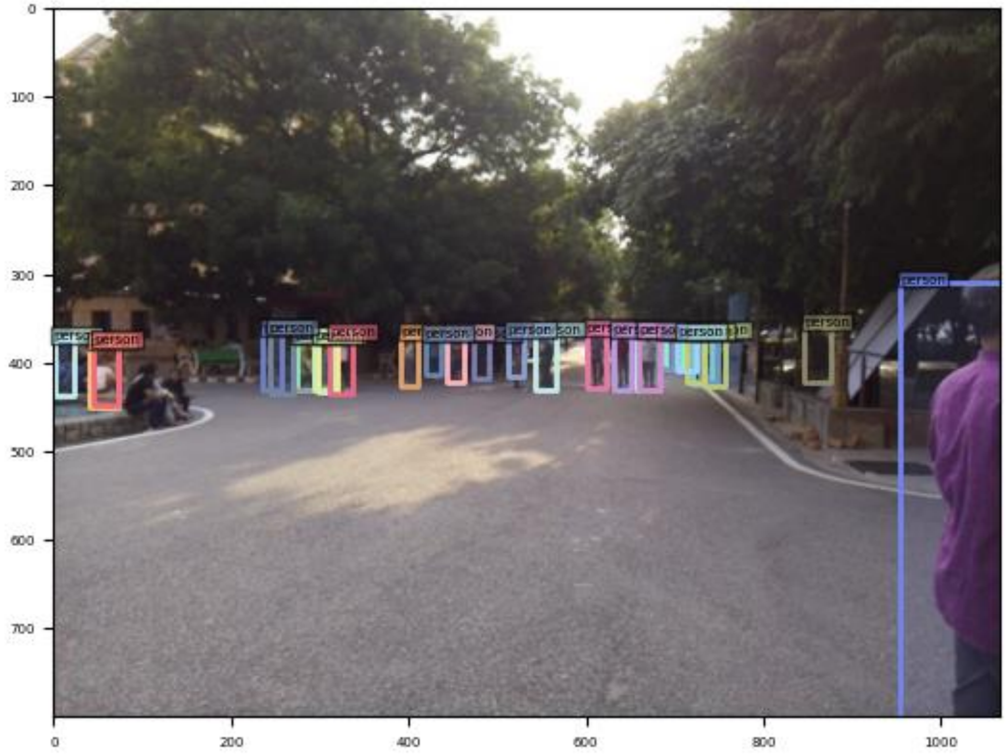




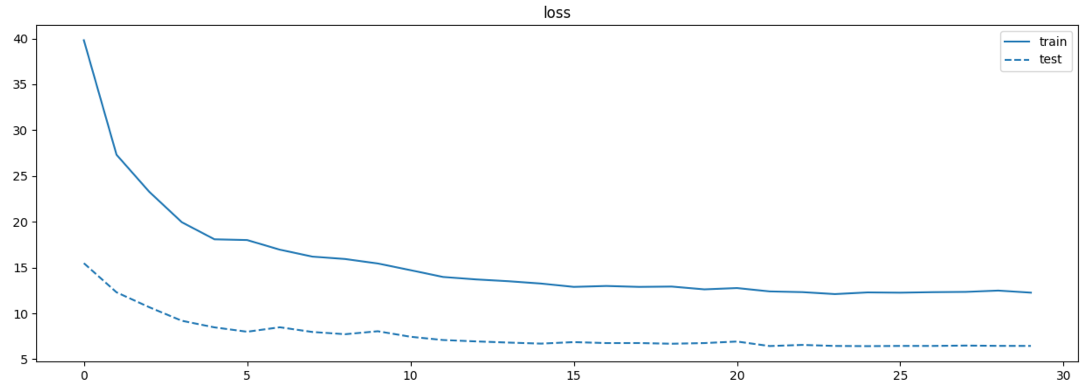
- Low sample size but occlusion handled well in this case:



- False negative for sitting people



Loss plot:



Due to time constraints (practical exam on 23rd Sept), I wasn't able to experiment more. If I could, I would've experimented by doing the following:

- Lower learning rate, around 0.0001 (I used 0.00025)
- If I had access to more GPU memory, I would have increased batch size (I used batch size of 2)
- Since the dataset is smaller, I would have run it for more epochs (I ran it for 30 epochs)
- In DETR based models, num_queries generally refers to the number of object queries used in the model. It correlates to the number of objects the model can detect in a single image. Default value is 900. I would've tried with a smaller value (200-500)
- Since it's a small dataset which I'm running for more epochs, I would slightly increase weight decay to prevent overfitting. Weight decay adds a penalty term to the loss function that is proportional to the sum of the squared weights in the model.