**CMPE 249  
Report for HW1 – 2D Object Detection**

**Uttej Kumar Reddy Gade  
016065543**

**Option**: Training & Inference

**Models for Training**:  
YOLOv5 (<https://github.com/ultralytics/yolov5>)  
MMDetection (<https://github.com/open-mmlab/mmdetection>) (In Progress)

**Models for Inference:**Fast RCNN

**Dataset:** [KITTI 2D Object Detection Dataset](https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark)

**Description**:  
**Step 1: Reducing KITTI Dataset for quick training**

Extracted the first 1000 images and labels for training and the next 100 for testing from below links.  
[Download left color images of object data set (12 GB)](https://s3.eu-central-1.amazonaws.com/avg-kitti/data_object_image_2.zip)  
[Download training labels of object data set (5 MB)](https://s3.eu-central-1.amazonaws.com/avg-kitti/data_object_label_2.zip)

Organised the files in the below manner which is what YOLOv5 looks for.

[*Directory structure*](https://github.com/uttejkumarreddy/cmpe249-hw1/tree/master/dataset) *dataset  
 - images  
 -train  
 -test   
 - labels  
 -train  
 -test*

***TRAINING – YOLOv5***

**Step 2: Formatting dataset into suitable format for YOLOv5 training**

YOLOv5 looks for labels in the format [CATEGORY] [BBOX\_X] [BBOX\_Y] [HEIGHT] [WIDTH] where BBOX\_X and BBOX\_Y are the center coordinates of the bounding box. However, the labels downloaded from the source has [CATEGORY] with 14 values encoding more attributes.

So, as the next step, converted the labels to required format using the below link as reference:  
<https://github.com/packyan/Kitti2Coco/blob/master/kitti2coco-label-trans.py>

Code:[https://github.com/uttejkumarreddy/cmpe249-hw1/blob/master/kitti-labels-to-coco-format.ipynb](https://github.com/uttejkumarreddy/cmpe249-hw1/blob/master/kitti-labels-to-coco-format.ipynb%20)

Summary:

Encode all categories in the label to a number and store it.

1. For each line in label file, extract the 5th-8th value and scale them to get required values.

bbox\_center\_x = float( (x1 + (x2 - x1) / 2.0) / img\_width)

bbox\_center\_y = float( (y1 + (y2 - y1) / 2.0) / img\_height)

bbox\_width = float((x2 - x1) / img\_width)

bbox\_height = float((y2 - y1) / img\_height)

1. Write them in new label files in the above mentioned format.

[Raw Labels:](https://github.com/uttejkumarreddy/cmpe249-hw1/tree/master/dataset/labels-raw)  
Pedestrian 0.00 0 -0.20 712.40 143.00 810.73 307.92 1.89 0.48 1.20 1.84 1.47 8.41 0.01

[Processed Labels:](https://github.com/uttejkumarreddy/cmpe249-hw1/tree/master/dataset/labels-raw)  
0 0.6221936274509804 0.6093513513513513 0.08033496732026148 0.4457297297297298

Step 3: Download, configure and run YOLOv5

Download YOLOv5 from Github. Configure the coco.yaml file In data folder as follows:

path: D:\present\cmpe249-hw1\dataset  # dataset root dir

train: D:\present\cmpe249-hw1\dataset\images\train

val: D:\present\cmpe249-hw1\dataset\images\test

test:

# Classes

names:

  0: Pedestrian

  1: Truck

  2: Car

  3: Cyclist

  4: DontCare

  5: Misc

  6: Van

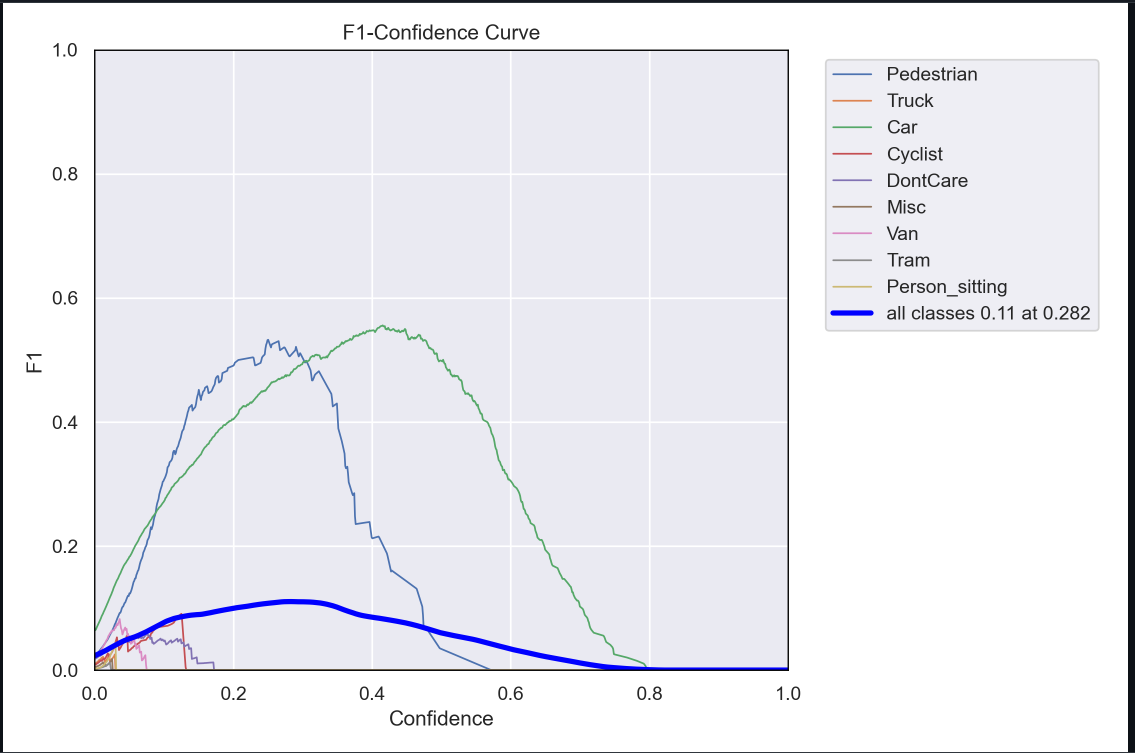
  7: Tram

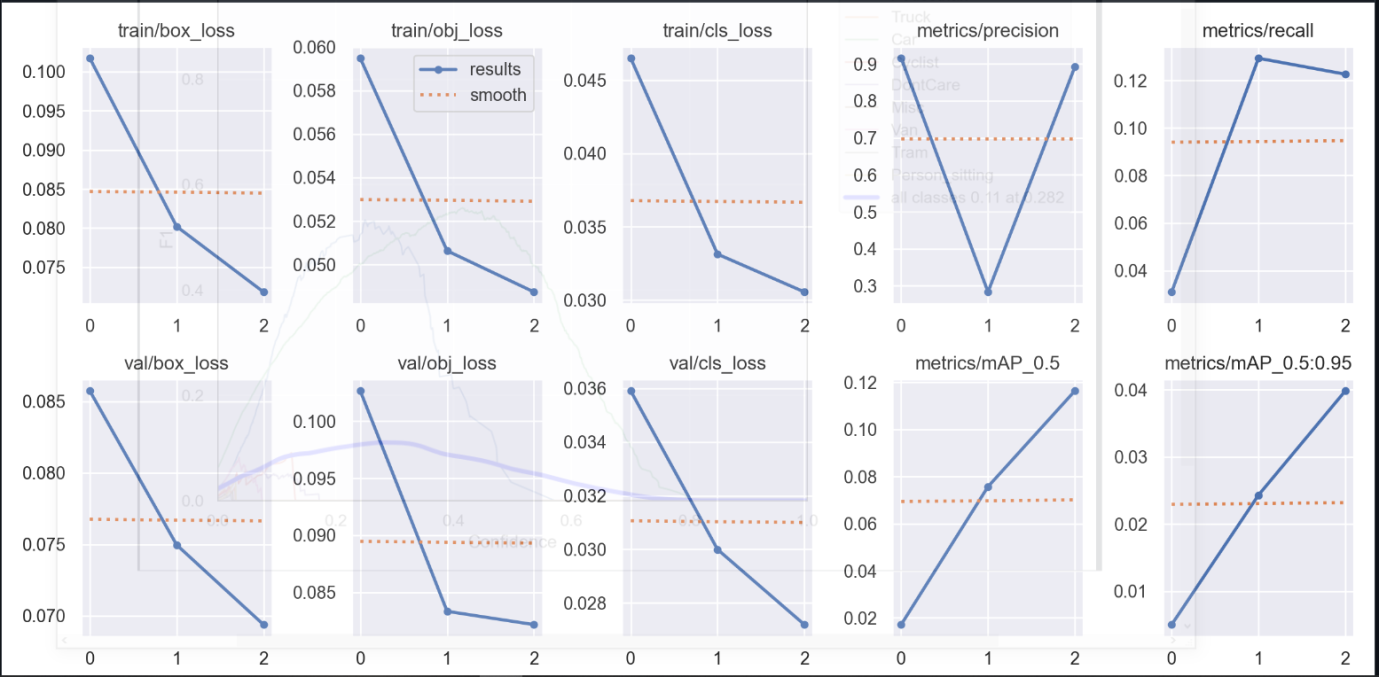
  8: Person\_sitting

As the training takes a long time for each epoch, ran the model for only 3 epochs with the following command:  
 python train.py --epochs 3 --data coco.yaml --weights yolov5s.pt

The results can be seen [here](https://github.com/uttejkumarreddy/cmpe249-hw1/tree/master/yolov5-results/exp)

For 3 epochs:





***INFERENCE – FAST RCNN (In Progress)***

**Step 1: Convert the KITTI dataset to COCO format**

Fast RCNN requires

Following the below references, converted the KITTI dataset to COCO format. The following is the summary of the code.

<https://medium.com/codable/convert-any-dataset-to-coco-object-detection-format-with-sahi-95349e1fe2b7>  
<https://pypi.org/project/sahi/>

1. Encode the categories from the labels files.
2. For every training label, create a COCOImage and for every annotation in that label, create a COCOAnnotation object with the bounding boxes information (mid point and height and width of the box) and the category.
3. Similarly, perform the same for test labels.
4. Save both in .json format.

References:   
<https://medium.com/codable/convert-any-dataset-to-coco-object-detection-format-with-sahi-95349e1fe2b7>  
<https://pypi.org/project/sahi/>  
<https://github.com/packyan/Kitti2Coco/blob/master/kitti2coco-label-trans.py>