CMPE-249: HOMEWORK ASSIGNMENT

2D or 3D Object Detection

Repository: https://github.com/lphucthinh40/cmpe249-object-detection-hw

1. Chosen Dataset: Argoverse

   For this assignment, I picked Argoverse 1.1 Dataset for 2D object detection. This dataset contains 8 classes: person, bicycle, car, motorcycle, bus, truck, traffic_light, stop_sign. Train set contains 39372 images with a total of 771774 annotations. Argoverse dataset annotations and metadata are all stored in a json file (one for each train, validation, and test set). This json file contains an "images" list for metadata of all images and an "annotations" list for annotations of all images.

   Class distributions:

Class Name	Label Count
person	89795
bicycle	8372
car	486022
motorcycle	4090
bus	6532
truck	76494
traffic_light	95018
stop_sign	5451

2. Training Ultralytics' YOLOv5 model
   In order to train YOLOv5, I converted Argoverse annotation json files (3 json files for train,eval, and test set) into annotation text files for each image. I picked yolov5s model which provides good balance in term of accuracy and training speed. I trained YOLOv5 on Argoverse dataset using the provided train.py script from the original repo. After training for 10 epochs on Google Colab and I obtained mean Average Precision for this model as follow:
   mAP 0.5: 0.377
   mAP 0.95: 0.22
   mAP was still increasing steadily up to 10 epochs which means we can continue to train this model with more epochs to improve its mean average precision. It is noteworthy that 'car' labels take up 58% of all labels in this dataset. As a result, 'car' class has mAP-0.5 of 0.652, much higher than other classes. The training results & evaluation can be found in Train-Yolov5.ipynb notebook.

3. Training Detectron 2 model
   In order to train Detectron 2, I convert Argoverse annotation json files into a dataset dictionary that can be registered to DatasetCatalog. I chose the faster_rcnn_R_50_FPN_3x model to train for this object detection task. Similar to YOLOv5, I trained Detectron2 on Argoverse dataset for 10 epochs. Since we cannot explicitly set the number of epochs to train detectron model, I tried to set Max-Iterations and Batch-per-Image to produce approximately the same epoch using this formula: epochs = max-iters x batch-size / images-size. I obtained mean average precision for this model as follow:
   mAP: 0.2383
   mAP 0.5: 0.385439
   Training results & evaluation can be found in Train-Detectron2.ipynb notebook

4. Comparison:
   Detectron 2 has better mean average precision in general after 10 epochs, however, YOLOv5 has a higher average precision for vehicle class. It also offers much better inference speed and its mAP can still improve with extra epochs. I will continue to train both models with extra epochs while mAP still appears to increase steadily.

5. Inference:
   I added inference script detect.py to perform inferences on images and videos. I also created Yolov5Detector & Detectron2Detector that can load the pretrained weights for Argo dataset and perform object detection on images.