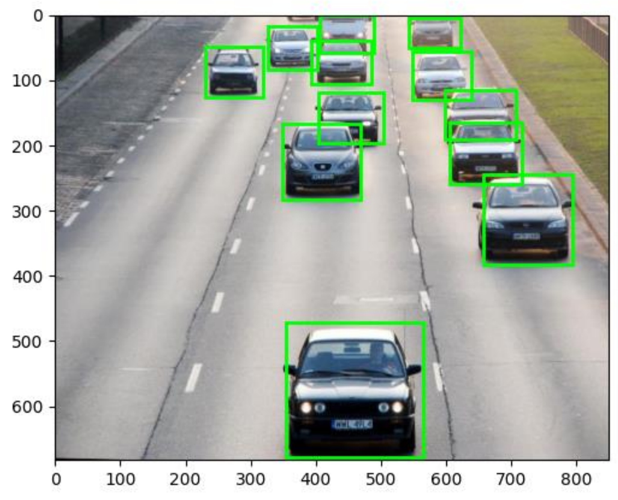
Homework #3

1 – Camera Calibration

Using your own (cell phone) camera of a chessboard (as one at A block) take pictures of 20+ non coplanar. A github repo, e.g. https://github.com/udacity/CarND-Camera-Calibration will be helpful for this question.

2 – Visualize the Bounding Boxes and Calculate IoUs

Fill the TODOs in each of the files attached starting with "visualization.py" to implement a function that plots the prediction and ground truth bounding boxes over an image. "iou.py" file calculates the ious between two all of the bounding boxes of the ./data/road1.png file. Add a new key:value to the dictionary with "scores":iou of the predictions.json file and save it as "predictions_sc.json"



Assumptions

- Bounding boxes: a list of [x_min, y_min, x_max, y_max] upper and lower corners of the bbox of image coordinates in pixels.
- Ground truth data: data/ground_truth.json contains the labels (bounding boxes and classes) for the ground truth data. Each observation is a dictionary with the following fields.

  ```
  {filename: str, boxes: List[List[int]], classes: List[int]}
  ```

- Predictions data data/predictions.json contains the labels for the predictions data. The format is the same as the ground truth data
- utils.py file will contain useful functions including
  - get_data(), reading the bboxes in json format in ./data/ground_truth.json and ./data/predictions.json.
  - calculate_iou(), takes two arrays containing the bounding boxes coordinates as inputs. iou = calculate_iou(np.array([0, 0, 100, 100]), np.array([10, 10, 120, 120]))

2.1 - Calculate Precision / Recall

Then, you are asked to calculate the precision and recall for a given set of predictions and ground truths in precision_recall.py. Use different threshold values of IoU (0.3-0.8) to determine if a prediction is a true positive or not.

Details

The precision_recall function in precision_recall.py takes as inputs a ious NxM array of IoU values as well as two list pred_classes and gt_classes containing the M predicted classes ids and the N ground truth classes ids.

The ious array contains the pairwise IoU values between the N ground truth bounding boxes and the M predicted bounding boxes such that:

```
ious[x, y] = calculate_iou(groundtruth[x], predictions[y])
```

You need to calculate the number of False Negatives to calculate the recall. You can use the IoU array to find the ground truth bounding boxes that are not predicted.

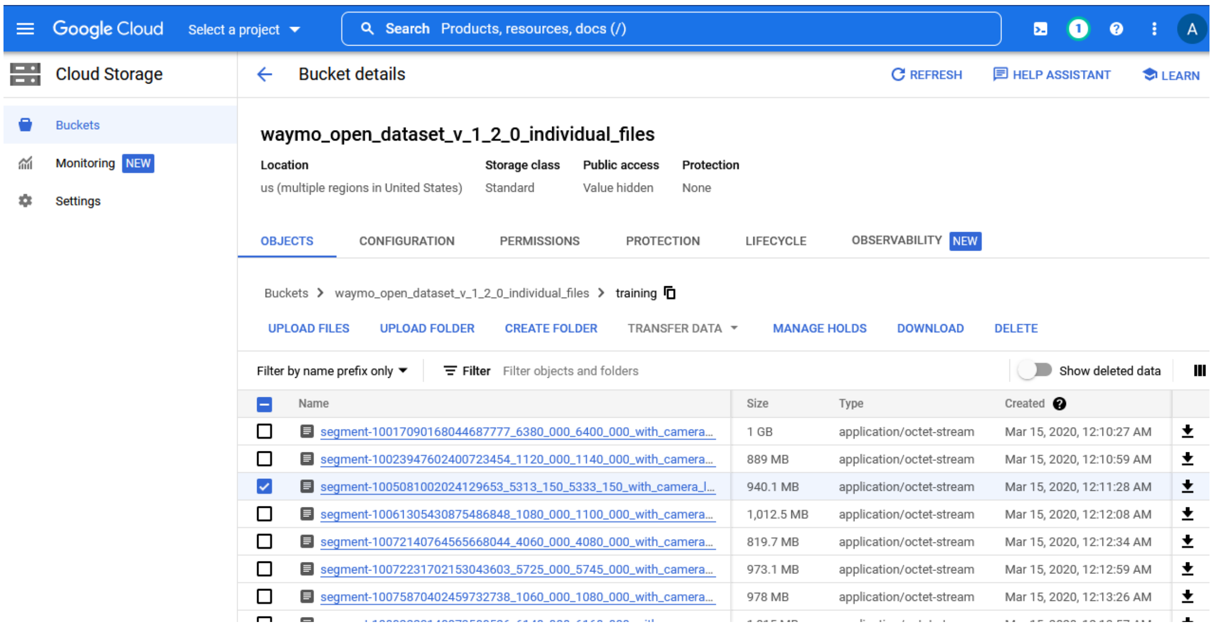
2.2 – Compute Mean Average Precision and NMS

You'll implement the mAP and NMS in "map.py" and "nms.py" files

The "utils.py" file contains various helper functions.

3 – Waymo Open Dataset

Register and download at least one tfrecord file (~1GB) from v_1_2_0 at https://waymo.com/open/download/. We have one of them install in PC labs ubuntu machines at "/datasets/waymo/1005081002024129653_5313_150_5333_150_with_camera_labels.tfrecord"



waymo_open_dataset_v_1_2_0_individual_files

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Value hidden	None

Buckets > waymo_open_dataset_v_1_2_0_individual_files > training

	Name	Size	Type	Created
☐	segment-10017090168044687777_6380_000_6400_000_with_camera...	1 GB	application/octet-stream	Mar 15, 2020, 12:10:27 AM
☐	segment-10023947602400723454_1120_000_1140_000_with_camera...	889 MB	application/octet-stream	Mar 15, 2020, 12:10:59 AM
✓	segment-1005081002024129653_5313_150_5333_150_with_camera_l...	940.1 MB	application/octet-stream	Mar 15, 2020, 12:11:28 AM
☐	segment-10061305430875486848_1080_000_1100_000_with_camera...	1,012.5 MB	application/octet-stream	Mar 15, 2020, 12:12:08 AM
☐	segment-10072140764565668044_4060_000_4080_000_with_camera...	819.7 MB	application/octet-stream	Mar 15, 2020, 12:12:34 AM
☐	segment-10072231702153043603_5725_000_5745_000_with_camera...	973.1 MB	application/octet-stream	Mar 15, 2020, 12:12:59 AM
☐	segment-10075870402459732738_1060_000_1080_000_with_camera...	978 MB	application/octet-stream	Mar 15, 2020, 12:13:26 AM

Accessing the data from tfrecord file use simple reader by cloning https://github.com/gdlg/simple-waymo-open-dataset-reader that does not require tensorflow. Check the examples for accessing labeled images to display a single image.

4 – Object Detection

YOLO is one of the state-of-the-art, real-time object detection algorithms. In this question, you will apply the YOLO algorithm to detect objects in images. We have provided a series of images that you can test the YOLO algorithm on. We have provided some images located in the`./images/`folder to test yolov3 model or you can work with the latest version YOLOv9 - Ultralytics YOLOv8 Dokümanlar or any other method of your choice. To verify the detection use the images under "./data/images" optionally you can take your own pictures and feed them to the model. Since the weights of yolo v3 file is too big, you are to locate and download on your own.