1. **Project overview:**

The objective of this project is to train a CNN model to detect and classify vehicles, pedestrians and cyclists using the database provided by Waymo.

2. **Set up:**

The project was all done in the workspace provided by Udacity. The steps to build the project are the same as provided by udacity, just with a few changes to the folder name of the new template.

to run jupyter:

cd /home/workspace/jupyter notebook --port 3002 --ip=0.0.0.0 --allow-root

Download the pretrained model and move it to /home/workspace/experiments/pretrained_model/. Follow the steps below:

cd /home/workspace/experiments/pretrained_model/

wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz

tar -xvzf ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz

rm -rf ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.tar.gz

We need to edit the config files to change the location of the training and validation files, as well as the location of the label_map file, pretrained weights. We also need to adjust the batch size. To do so, run the following:
cd /home/workspace/

python edit_config.py --train_dir /home/workspace/data/train/ --eval_dir /home/workspace/data/val/ --batch_size 2 --checkpoint /home/workspace/experiments/pretrained_model/ssd_resnet50_v1_fpn_640x640_coco17_tpu-8/checkpoint/ckpt-0 --label_map /home/workspace/experiments/label_map.pbtxt

A new config file called pipeline_new.config will be created in the /home/workspace/ directory. Move this file to the /home/workspace/experiments/reference/ directory.

training:

python experiments/model_main_tf2.py --model_dir=experiments/experiment0/ --pipeline_config_path=experiments/experiment0/pipeline_new.config

evaluation:

python experiments/model_main_tf2.py --model_dir=experiments/reference/ --pipeline_config_path=experiments/experiment0/pipeline_new.config --checkpoint_dir=experiments/experiment0/

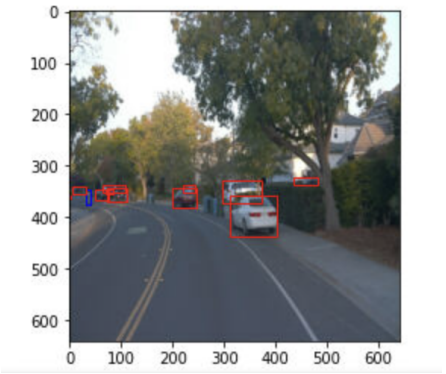
To monitor the training and evaluation: python -m tensorboard.main --logdir experiments/experiment0/

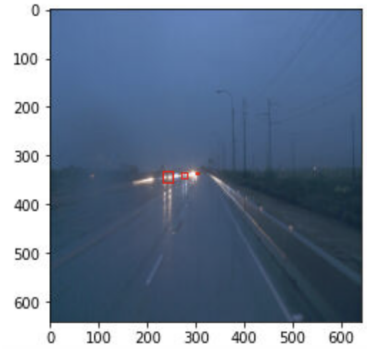
3. Dataset
    1. *Dataset Analysis*

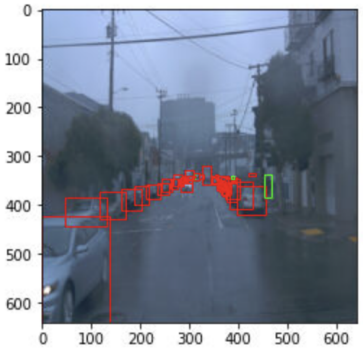
The database images present objects classified as cars, pedestrians and bicycles in different weather conditions . Most objects are cars, followed by pedestrians and bicycles. The

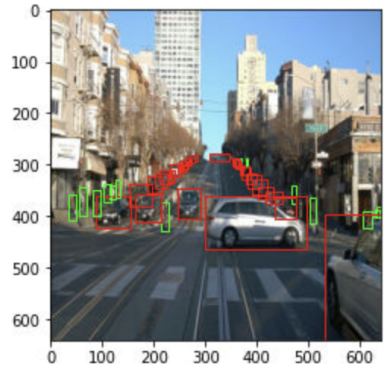
percentage of bicycles in the data is quite low, and may affect performance for bicycle training.

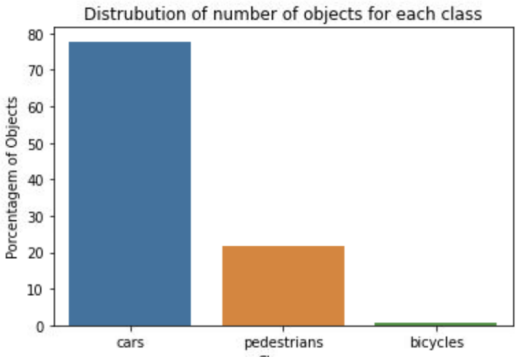
By analyzing the colors between the objects, we can see images containing pedestrians that have a greater red hue than cars and bicycles, which makes a lot of sense since green and blue are farther from skin color.

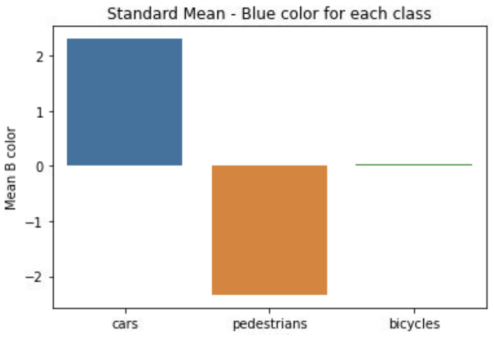


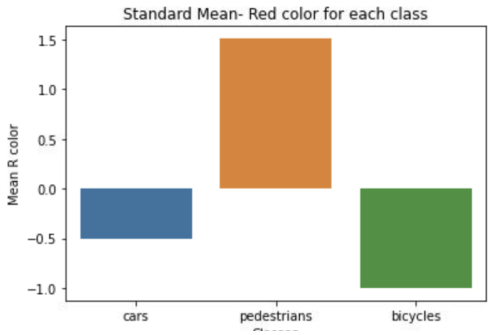


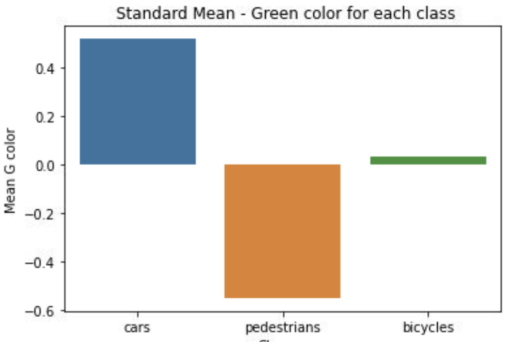












2.Cross-validation: The data already came divided.

4. Training

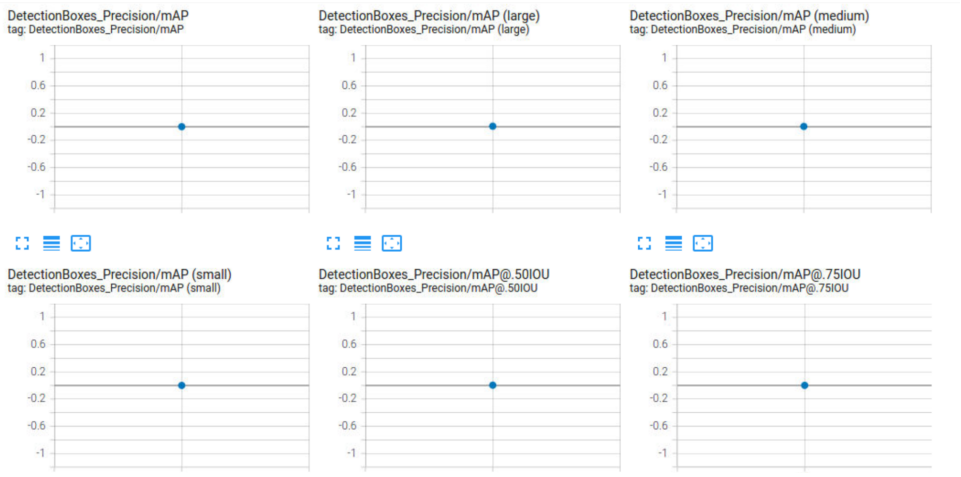
1.*Reference experiment*: During the training of the reference, the training was interrupted several times due to lack of memory in the Udacity workspace, so the results have jumps in time and performance in the plots. From the results of the reference, we see that the algorithm starts to converge better when it has a lower learning rate. The results of both Loss and mPA are quite bad, the algorithm would probably need more than 20 k interactions to converge on a good model.

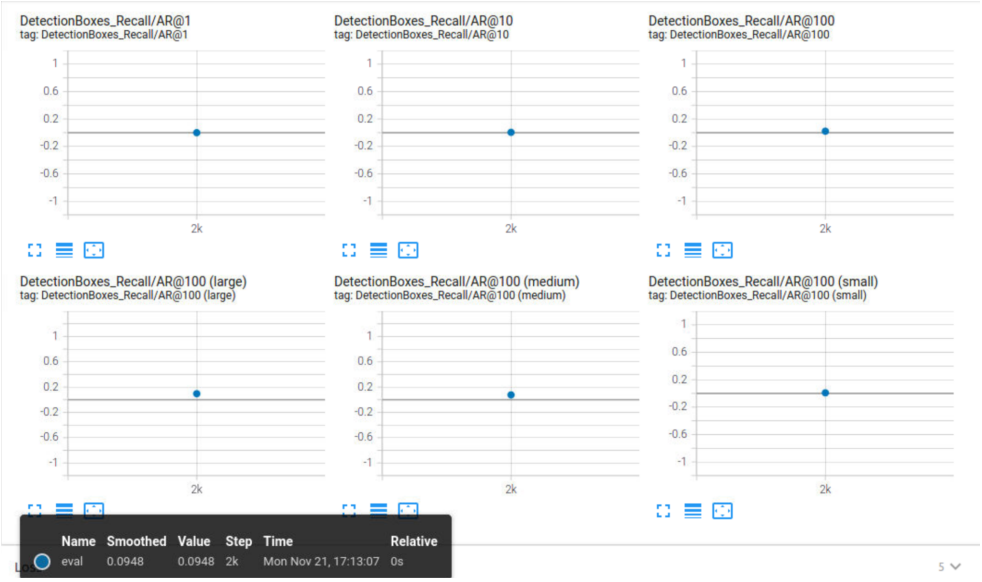
Loss: We can see that it is converging because all losses are decreasing. however, if we analyze the training period with the warm-up learning rate, we see that the losses were much lower than when the learning rate increased. One of the reasons for this is the batch size used. With such a small batch size (2) and a high learning rate, the fit of the model ends up being very random and depends on the images in the batch. In addition, this generates an overfitting of the data used in training.

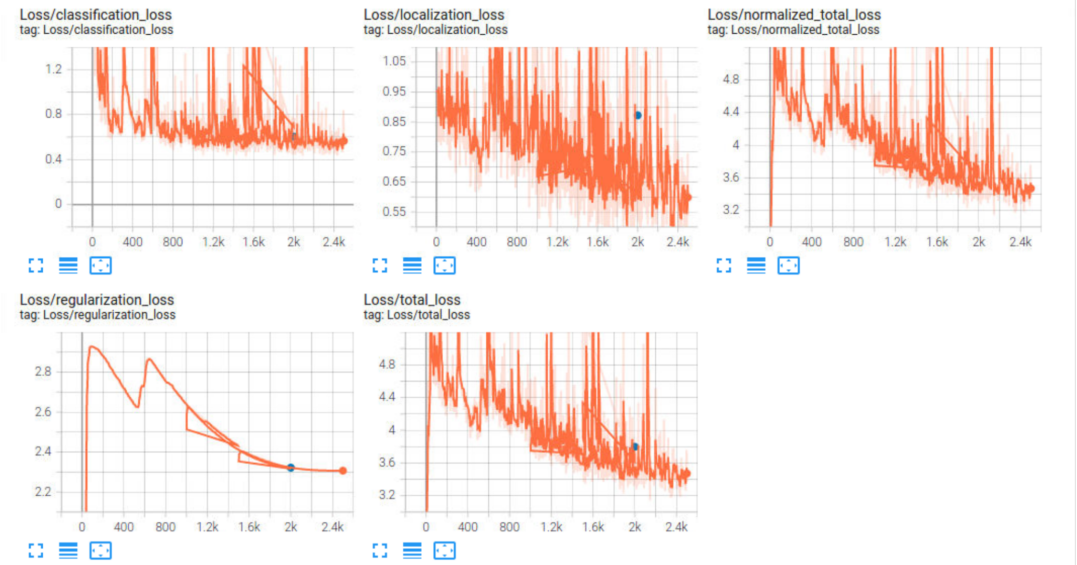
AP/mAP: During the validation we see that the model has a lower performance than the training, which proves that the model is overfitting. The main reason for this is the reduced size of the batch together with the high learning rate used. A factor that could contribute to the improvement of the model would be to increase the Batch size, decrease the learning rate, add random augmentations in the images, and train the model with many more interactions, such as 25k+.

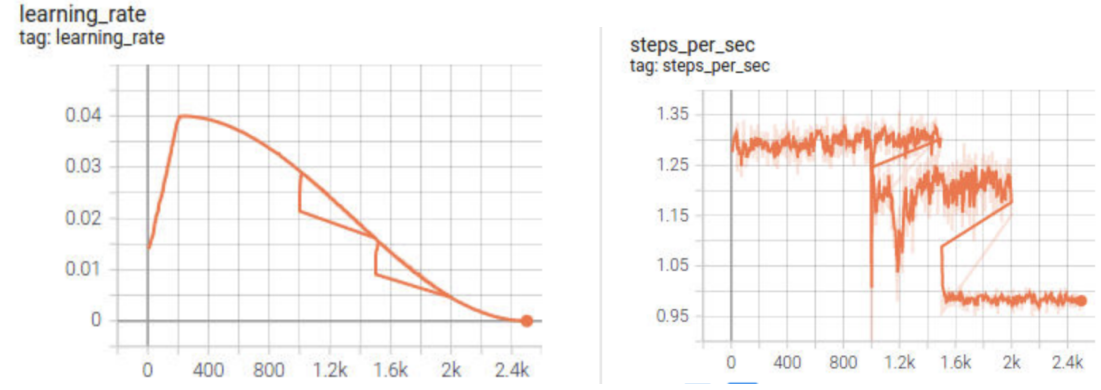
```
I1121 17:12:34.962473 140656250930944 coco_tools.py:138] DONE (t=0.03s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=32.03s).
Accumulating evaluation results...
DONE (t=0.39s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.001
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.004
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.001
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.005
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.007
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.005
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.022
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.008
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.076
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.095
```

```
INFO:tensorflow:Eval metrics at step 2000
I1121 17:13:07.433371 140656250930944 model_lib_v2.py:988] Eval metrics at step 2000
INFO:tensorflow:         + DetectionBoxes_Precision/mAP: 0.001087
I1121 17:13:07.443895 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP: 0.001087
INFO:tensorflow:         + DetectionBoxes_Precision/mAP@.50IOU: 0.003835
I1121 17:13:07.445716 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP@.50IOU: 0.003835
INFO:tensorflow:         + DetectionBoxes_Precision/mAP@.75IOU: 0.000512
I1121 17:13:07.447509 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP@.75IOU: 0.000512
INFO:tensorflow:         + DetectionBoxes_Precision/mAP (small): 0.000139
I1121 17:13:07.449330 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP (small): 0.000139
INFO:tensorflow:         + DetectionBoxes_Precision/mAP (medium): 0.004551
I1121 17:13:07.451023 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP (medium): 0.004551
INFO:tensorflow:         + DetectionBoxes_Precision/mAP (large): 0.007484
I1121 17:13:07.453898 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Precision/mAP (large): 0.007484
INFO:tensorflow:         + DetectionBoxes_Recall/AR@1: 0.000429
I1121 17:13:07.455851 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@1: 0.000429
INFO:tensorflow:         + DetectionBoxes_Recall/AR@10: 0.004931
I1121 17:13:07.457908 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@10: 0.004931
INFO:tensorflow:         + DetectionBoxes_Recall/AR@100: 0.021561
I1121 17:13:07.459864 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@100: 0.021561
INFO:tensorflow:         + DetectionBoxes_Recall/AR@100 (small): 0.008144
I1121 17:13:07.461560 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@100 (small): 0.008144
INFO:tensorflow:         + DetectionBoxes_Recall/AR@100 (medium): 0.076119
I1121 17:13:07.463185 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@100 (medium): 0.076119
INFO:tensorflow:         + DetectionBoxes_Recall/AR@100 (large): 0.094800
I1121 17:13:07.465063 140656250930944 model_lib_v2.py:991]      + DetectionBoxes_Recall/AR@100 (large): 0.094800
INFO:tensorflow:         + Loss/localization_loss: 0.872383
I1121 17:13:07.466784 140656250930944 model_lib_v2.py:991]      + Loss/localization_loss: 0.872383
INFO:tensorflow:         + Loss/classification_loss: 0.603809
I1121 17:13:07.468926 140656250930944 model_lib_v2.py:991]      + Loss/classification_loss: 0.603809
INFO:tensorflow:         + Loss/regularization_loss: 2.321912
I1121 17:13:07.473528 140656250930944 model_lib_v2.py:991]      + Loss/regularization_loss: 2.321912
INFO:tensorflow:         + Loss/total_loss: 3.798104
I1121 17:13:07.475254 140656250930944 model_lib_v2.py:991]      + Loss/total_loss: 3.798104
INFO:tensorflow:Waiting for new checkpoint at experiments/reference/
I1121 17:16:07.195711 140656250930944 checkpoint_utils.py:125] Waiting for new checkpoint at experiments/reference/
```




DetectionBoxes_Recall/AR@1
tag: DetectionBoxes_Recall/AR@1
DetectionBoxes_Recall/AR@10
tag: DetectionBoxes_Recall/AR@10
DetectionBoxes_Recall/AR@100
tag: DetectionBoxes_Recall/AR@100
DetectionBoxes_Recall/AR@100 (large)
tag: DetectionBoxes_Recall/AR@100 (large)
DetectionBoxes_Recall/AR@100 (medium)
tag: DetectionBoxes_Recall/AR@100 (medium)
DetectionBoxes_Recall/AR@100 (small)
tag: DetectionBoxes_Recall/AR@100 (small)
Name Smoothed Value Step Time Relative
eval 0.0948 0.0948 2k Mon Nov 21, 17:13:07 0s


Loss/classification_loss
tag: Loss/classification_loss
Loss/localization_loss
tag: Loss/localization_loss
Loss/normalized_total_loss
tag: Loss/normalized_total_loss
Loss/regularization_loss
tag: Loss/regularization_loss
Loss/total_loss
tag: Loss/total_loss


learning_rate
tag: learning_rate
steps_per_sec
tag: steps_per_sec

2. Improve on the reference: Let's consider how much we can improve using the same number of interactions as in the reference (because obviously if we didn't change anything and just made more interactions we would have a better result than the reference). Due to the low number of interactions we should not be able to get good loss and mPA values, but we can improve the performance of the previous model.

Increasing the Batch size could greatly improve the convergence rate and the overfitting problem of the model, however in the udacity workspace it is not possible. As we want to analyze the convergence within the same number of iterations, the things we can change are the model learning rates and image augmentation.

We will start by decreasing the learning rate values, the high learning rate values were the main reason for the low performance of the reference.

- learning_rate_base: 0.03
- warmup_learning_rate: 0.01

We will also add some random data augmentation to not overfiting the data

- random_rgb_to_gray
  - probability: 0.3
- random_adjust_saturations
  - min_delta: 0.4
  - max_delta: 1.2
- random_adjust_brightness
  - max_delta: 0.1
- random_adjust_contrasts
  - min_delta: 0.2
  - max_delta: 1.0

The augmentations to add random effects to the images, such as brightness, saturation, contrast and grayscale helped to train the model for different situations of luminosity, time and night conditions, preventing the model from overfitting with the trained data. The result obtained is shown in the figures below:

Contrast images: Grayscale images:

And the results of the model can be seen below. The mPA is still very low, the Loss however has improved from 4 to 2 . With this we can see that we can optimize the learning performance with only small configuration changes. Possibly this configuration would reach convergence in fewer interactions than the reference.

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.001
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.005
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.001
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.002
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.008
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.005
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.146
```

```
I1127 13:37:23.178438 139821371975424 model_lib_v2.py:988] Eval metrics at step 2498
INFO:tensorflow:    + DetectionBoxes_Precision/mAP: 0.000425
I1127 13:37:23.188440 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP: 0.000425
INFO:tensorflow:    + DetectionBoxes_Precision/mAP@.50IOU: 0.001019
I1127 13:37:23.190297 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP@.50IOU: 0.001019
INFO:tensorflow:    + DetectionBoxes_Precision/mAP@.75IOU: 0.000453
I1127 13:37:23.191936 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP@.75IOU: 0.000453
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (small): 0.000495
I1127 13:37:23.193607 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (small): 0.000495
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (medium): 0.000066
I1127 13:37:23.195192 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (medium): 0.000066
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (large): 0.005269
I1127 13:37:23.197014 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (large): 0.005269
INFO:tensorflow:    + DetectionBoxes_Recall/AR@1: 0.001158
I1127 13:37:23.198733 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@1: 0.001158
INFO:tensorflow:    + DetectionBoxes_Recall/AR@10: 0.002246
I1127 13:37:23.200478 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@10: 0.002246
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100: 0.007976
I1127 13:37:23.202185 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100: 0.007976
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (small): 0.000049
I1127 13:37:23.204010 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (small): 0.000049
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (medium): 0.005000
I1127 13:37:23.205709 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (medium): 0.005000
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (large): 0.145600
I1127 13:37:23.207441 139821371975424 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (large): 0.145600
INFO:tensorflow:    + Loss/localization_loss: 0.934269
I1127 13:37:23.208832 139821371975424 model_lib_v2.py:991]  + Loss/localization_loss: 0.934269
INFO:tensorflow:    + Loss/classification_loss: 0.679201
I1127 13:37:23.210283 139821371975424 model_lib_v2.py:991]  + Loss/classification_loss: 0.679201
INFO:tensorflow:    + Loss/regularization_loss: 0.566220
I1127 13:37:23.211642 139821371975424 model_lib_v2.py:991]  + Loss/regularization_loss: 0.566220
INFO:tensorflow:    + Loss/total_loss: 2.179690
```

We can also see now that the validation loss is closer to the training one, but it is still greater than the training value. Which means that the model is still overfitting. Possibly an even lower learning rate should be used.

Bonus: After finishing this part of the project I decided to do an additional test, I did a new training with a much lower learning rate using a momentum_optimizer with exponential decay,and the results were fantastic (due to lack of memory in the workspace I don't have the files related to this workout).

Exponential decay was used here as it is easier to plan the drop loss during the number of interactions used. The lower learning rate was a better match with the small batch size and allowed the model to have not only a much lower Loss but a much better AP. probably with 10k interactions maybe we can already get good results from the model.

```
exponential_decay_learning_rate {
 initial_learning_rate: 0.001
  decay_factor: 0.9
  decay_steps: 100
  burnin_learning_rate: 0.001
  burnin_steps: 100
}
```

```
INFO:tensorflow:Step 2400 per-step time 0.757s loss=0.808
I1127 13:22:19.512187 139680452658944 model_lib_v2.py:682] Step 2400 per-step time 0.757s loss=0.808
INFO:tensorflow:Step 2500 per-step time 0.738s loss=0.676
I1127 13:23:36.593325 139680452658944 model_lib_v2.py:682] Step 2500 per-step time 0.738s loss=0.676
(sdc-c1-gpu-augment) root@997c053b632f:/home/workspace#
```

```
DONE (t=0.31s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.083
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.166
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.078
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.027
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.318
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.436
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.023
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.088
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.119
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.061
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.408
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.529
```

```
INFO:tensorflow:    + DetectionBoxes_Precision/mAP: 0.082766
I1127 13:29:00.583698 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP: 0.082766
INFO:tensorflow:    + DetectionBoxes_Precision/mAP@.50IOU: 0.166026
I1127 13:29:00.585557 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP@.50IOU: 0.166026
INFO:tensorflow:    + DetectionBoxes_Precision/mAP@.75IOU: 0.078120
I1127 13:29:00.587262 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP@.75IOU: 0.078120
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (small): 0.027064
I1127 13:29:00.589400 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (small): 0.027064
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (medium): 0.317930
I1127 13:29:00.591071 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (medium): 0.317930
INFO:tensorflow:    + DetectionBoxes_Precision/mAP (large): 0.435539
I1127 13:29:00.592723 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Precision/mAP (large): 0.435539
INFO:tensorflow:    + DetectionBoxes_Recall/AR@1: 0.023301
I1127 13:29:00.594460 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@1: 0.023301
INFO:tensorflow:    + DetectionBoxes_Recall/AR@10: 0.088097
I1127 13:29:00.596153 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@10: 0.088097
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100: 0.118876
I1127 13:29:00.597882 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100: 0.118876
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (small): 0.060829
I1127 13:29:00.599608 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (small): 0.060829
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (medium): 0.407501
I1127 13:29:00.601363 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (medium): 0.407501
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (large): 0.528800
I1127 13:29:00.603193 139705300416256 model_lib_v2.py:991]  + DetectionBoxes_Recall/AR@100 (large): 0.528800
INFO:tensorflow:    + Loss/localization_loss: 0.551178
I1127 13:29:00.604676 139705300416256 model_lib_v2.py:991]  + Loss/localization_loss: 0.551178
INFO:tensorflow:    + Loss/classification_loss: 0.284711
I1127 13:29:00.606018 139705300416256 model_lib_v2.py:991]  + Loss/classification_loss: 0.284711
INFO:tensorflow:    + Loss/regularization_loss: 0.246422
I1127 13:29:00.607396 139705300416256 model_lib_v2.py:991]  + Loss/regularization_loss: 0.246422
INFO:tensorflow:    + Loss/total_loss: 1.082311
I1127 13:29:00.608673 139705300416256 model_lib_v2.py:991]  + Loss/total_loss: 1.082311
```