Object Detection in an Urban Environment

In this project, I applied the skills I've gained in this course to use a pretrained neural network to detect and classify objects using data from Waymo.

I monitored the training with TensorBoard and generated the losses, precision and recall graphs to decide which model was best for our dataset after experimenting with different hyperparameters to improve the model's performance in the pipeline.

The models I've used were:

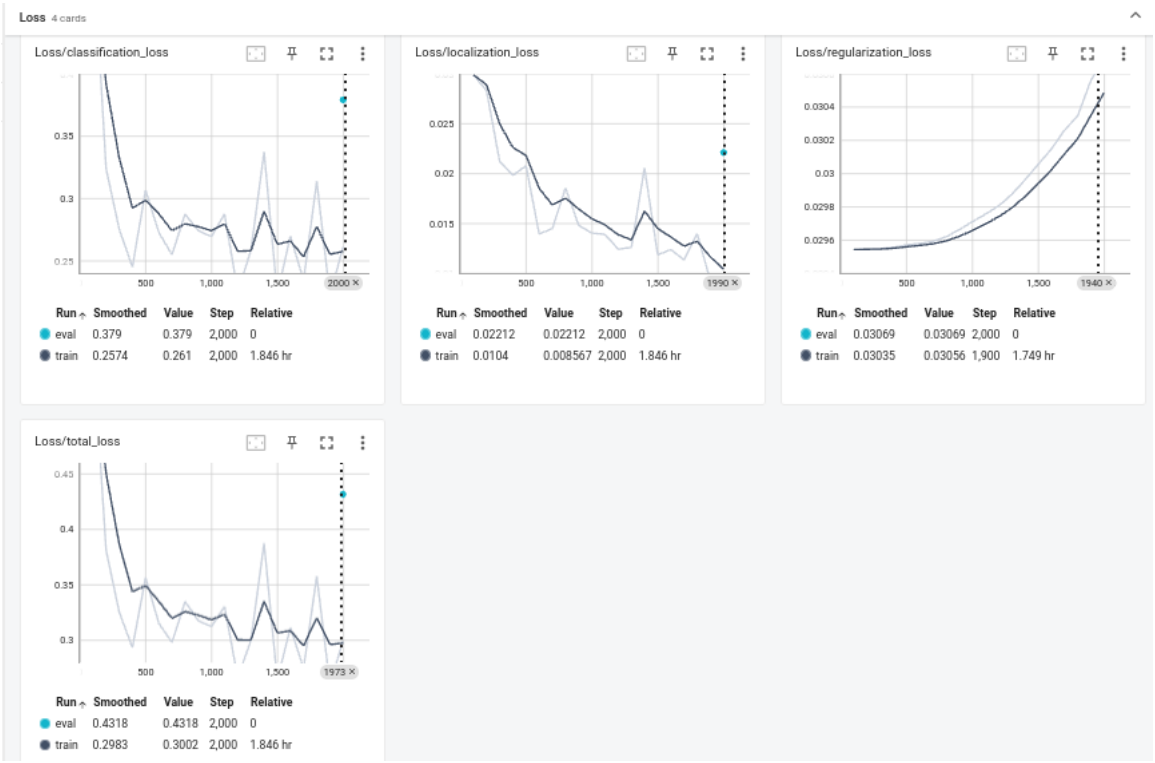
Model 1: EfficientDet D1 640x640

Model 2: SSD MobileNet V2 FPNLite 640x640

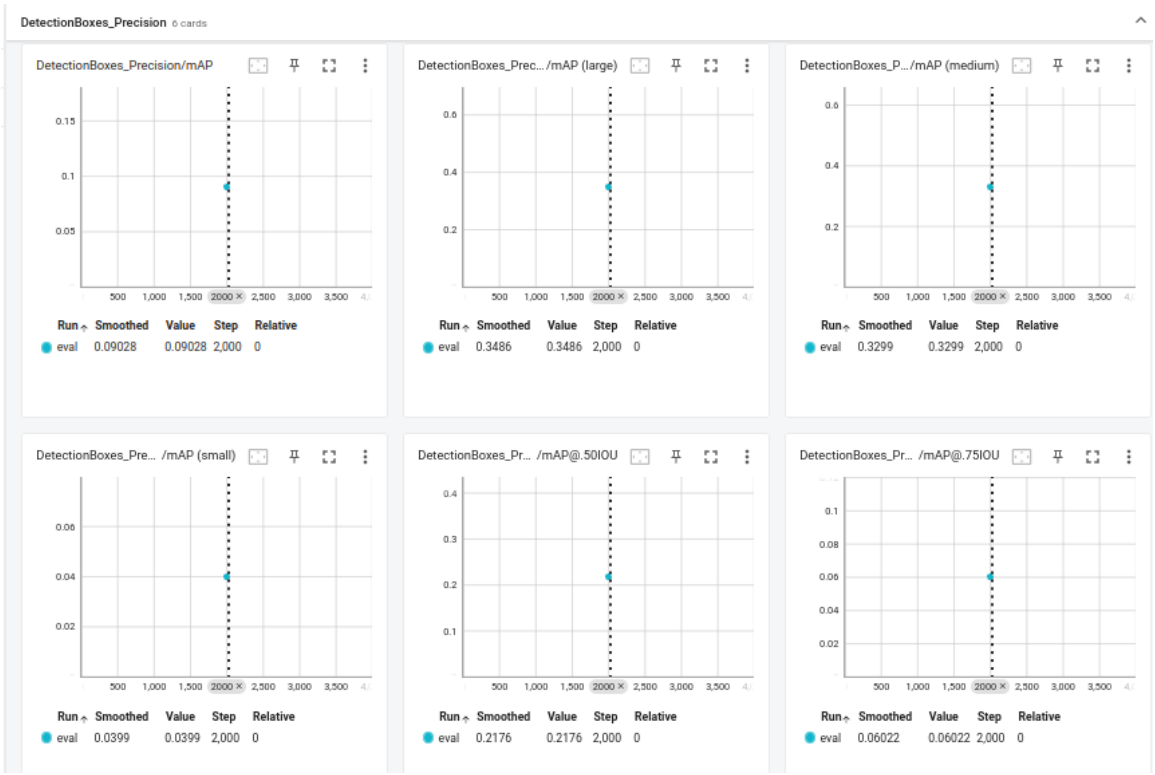
Model 3: SSD ResNet50 V1 FPN 640x640 (RetinaNet50)

Model 1:

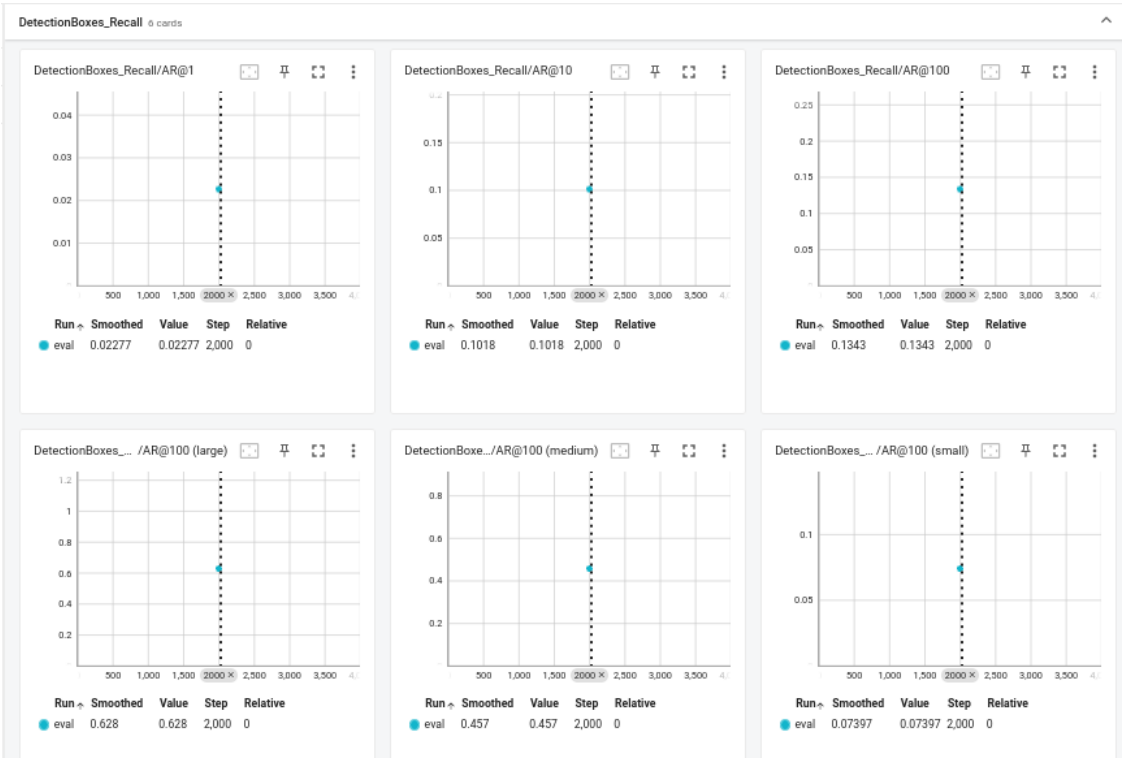
Losses:



Precision:

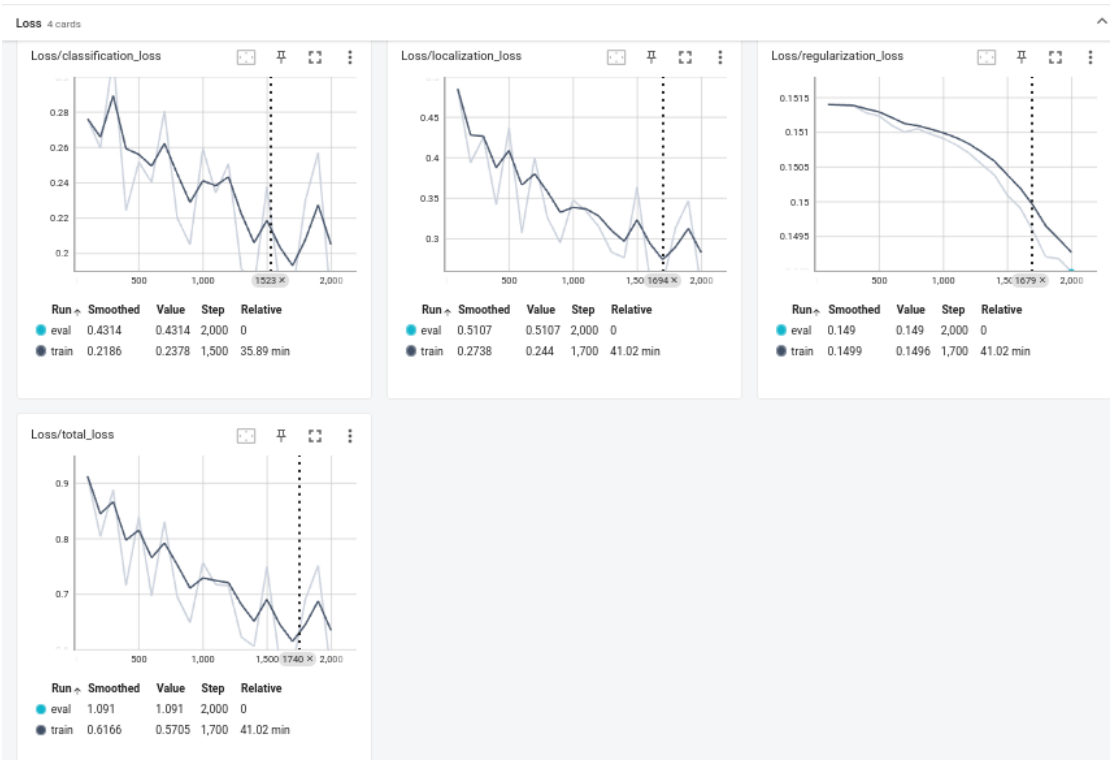


Recall:

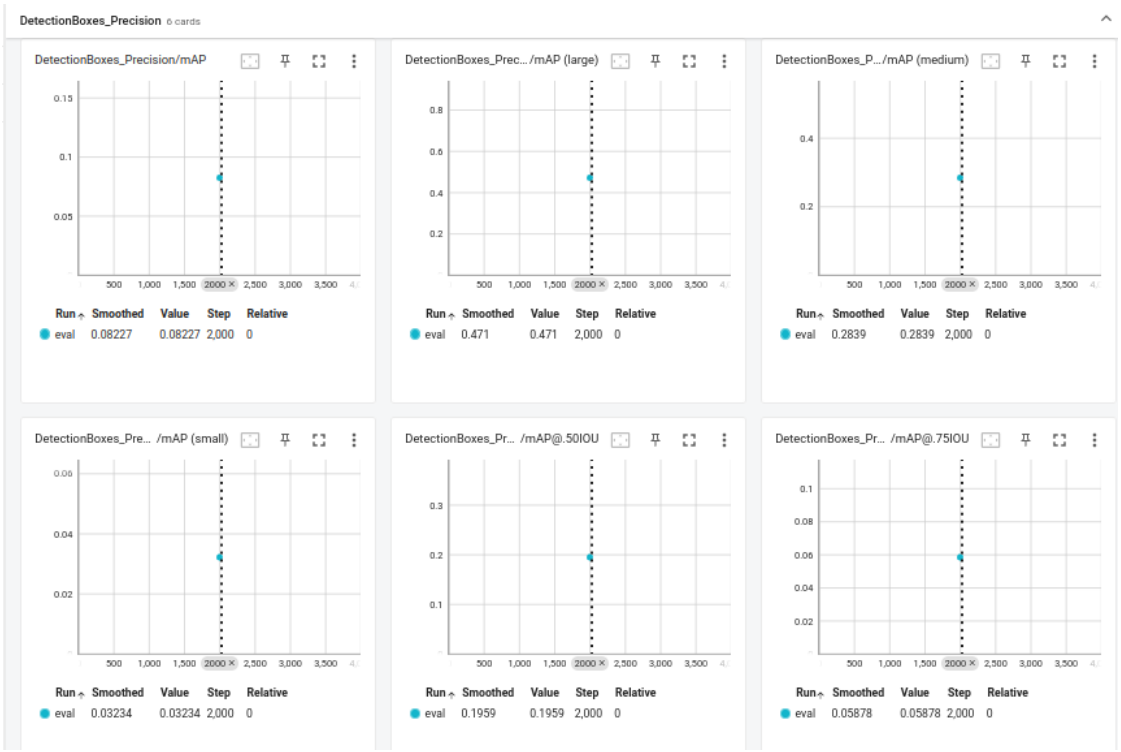


Model 2:

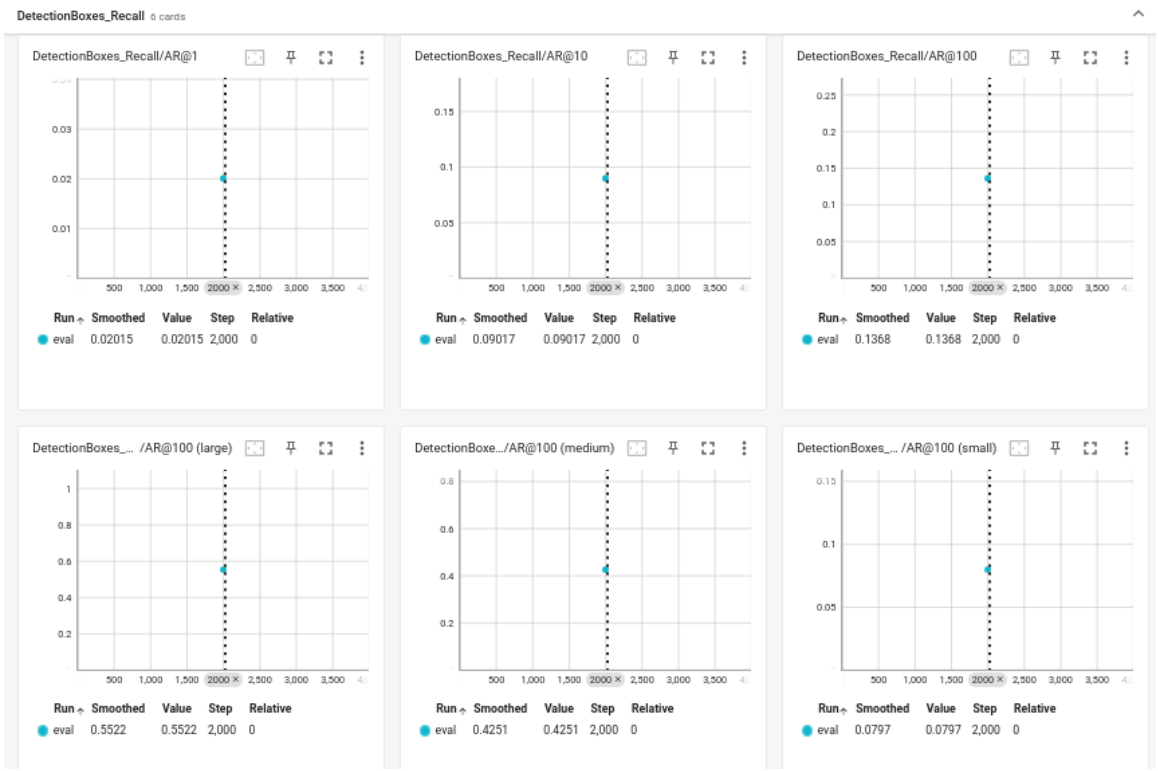
Losses:



Precision:

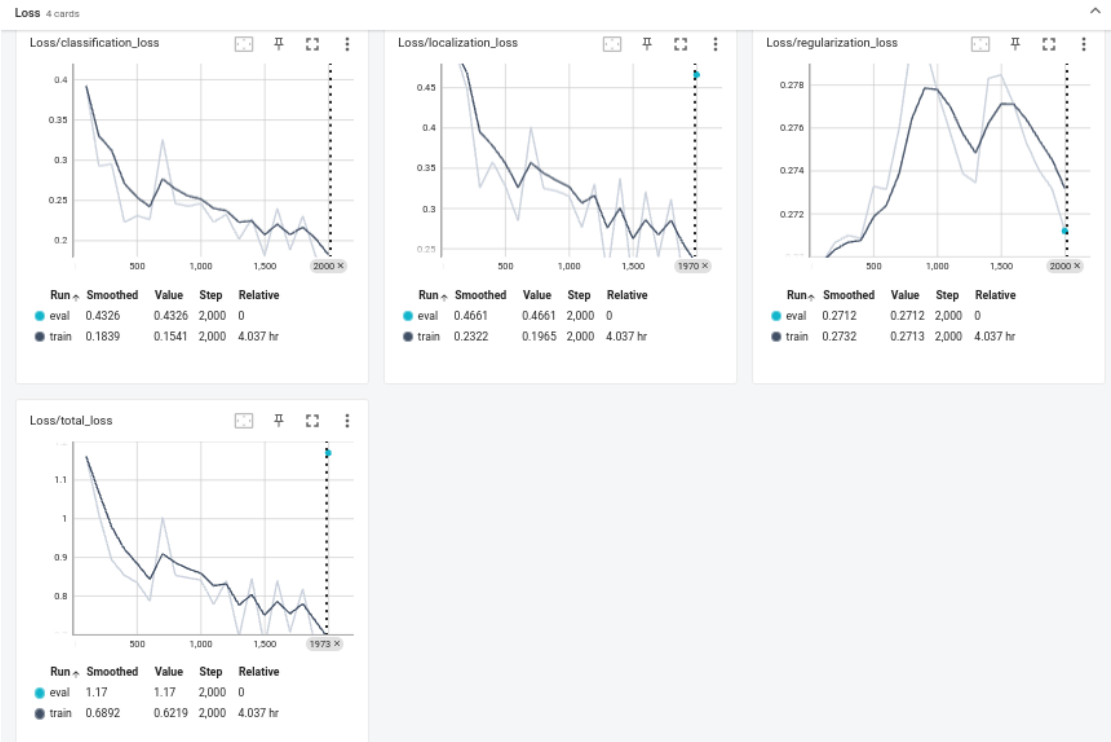


Recall:

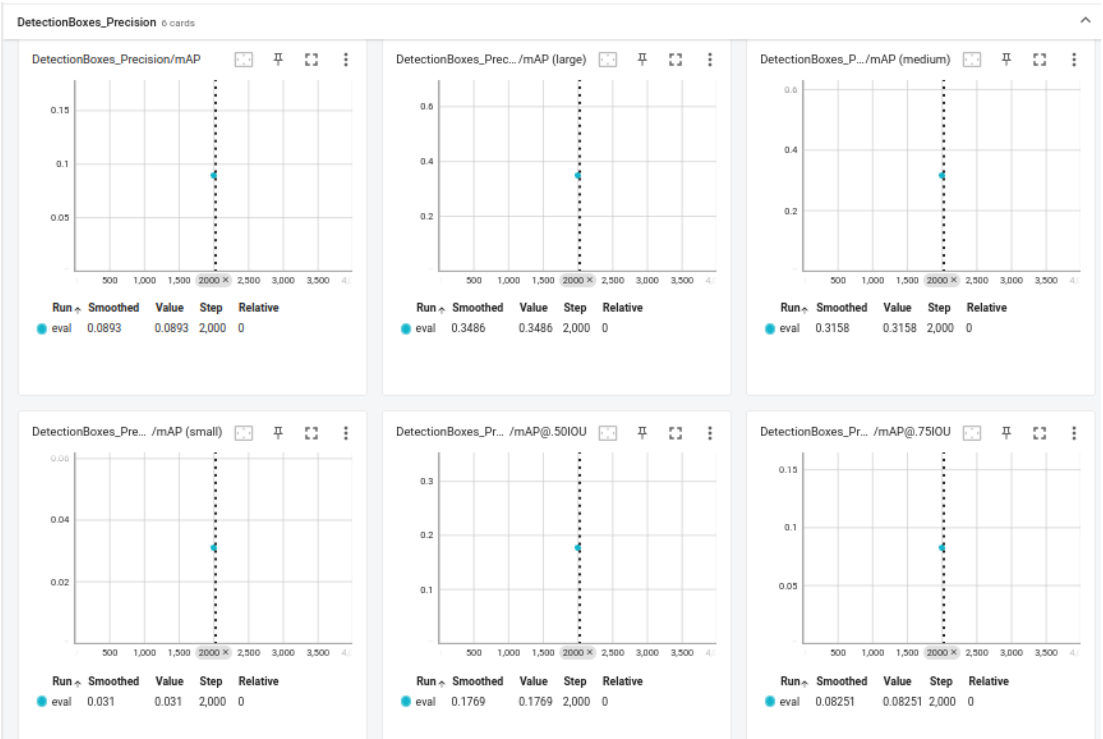

DetectionBoxes_Recall 6 cards
DetectionBoxes_Recall/AR@1
Run Smoothed Value Step Relative
eval 0.02015 0.02015 2,000 0
DetectionBoxes_Recall/AR@10
Run Smoothed Value Step Relative
eval 0.09017 0.09017 2,000 0
DetectionBoxes_Recall/AR@100
Run Smoothed Value Step Relative
eval 0.1368 0.1368 2,000 0
DetectionBoxes_... /AR@100 (large)
Run Smoothed Value Step Relative
eval 0.5522 0.5522 2,000 0
DetectionBoxe.../AR@100 (medium)
Run Smoothed Value Step Relative
eval 0.4251 0.4251 2,000 0
DetectionBoxes_... /AR@100 (small)
Run Smoothed Value Step Relative
eval 0.0797 0.0797 2,000 0

Model 3:

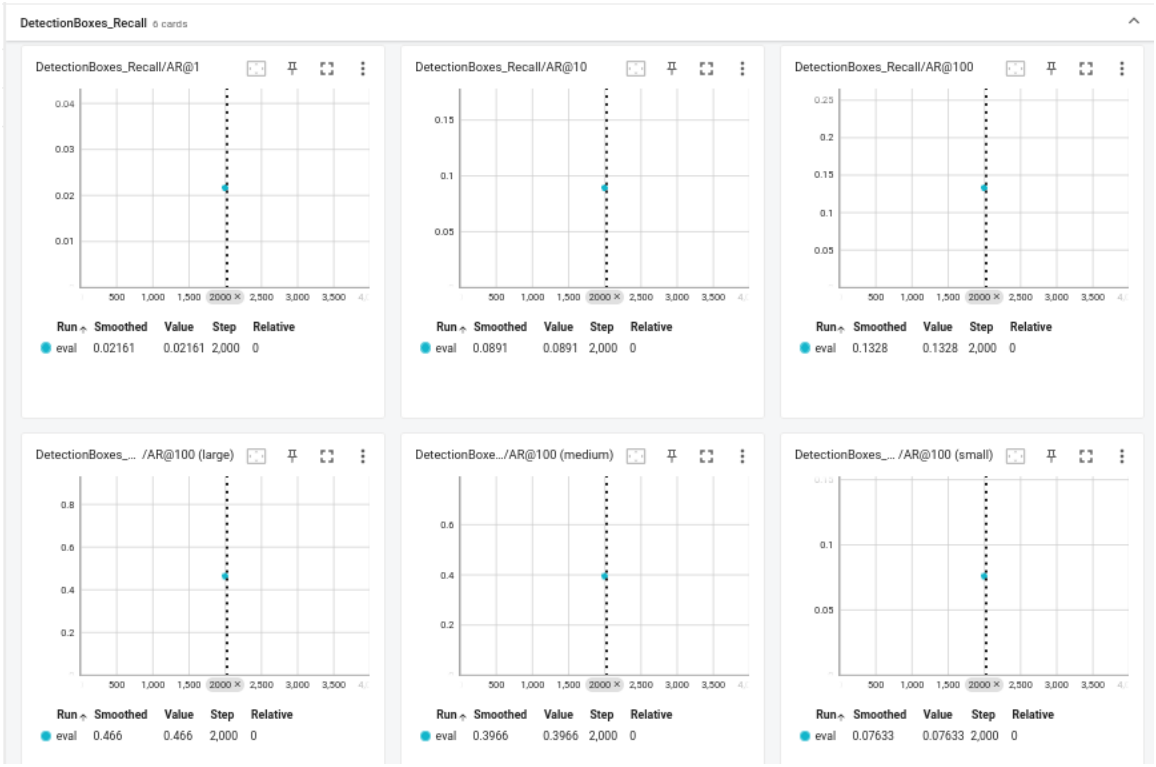
Losses:



Precision:



Recall:


DetectionBoxes_Recall 6 cards
DetectionBoxes_Recall/AR@1
DetectionBoxes_Recall/AR@10
DetectionBoxes_Recall/AR@100
DetectionBoxes_... /AR@100 (large)
DetectionBoxe.../AR@100 (medium)
DetectionBoxes_... /AR@100 (small)
Run Smoothed Value Step Relative
eval 0.02161 0.02161 2,000 0
eval 0.0891 0.0891 2,000 0
eval 0.1328 0.1328 2,000 0
eval 0.466 0.466 2,000 0
eval 0.3966 0.3966 2,000 0
eval 0.07633 0.07633 2,000 0

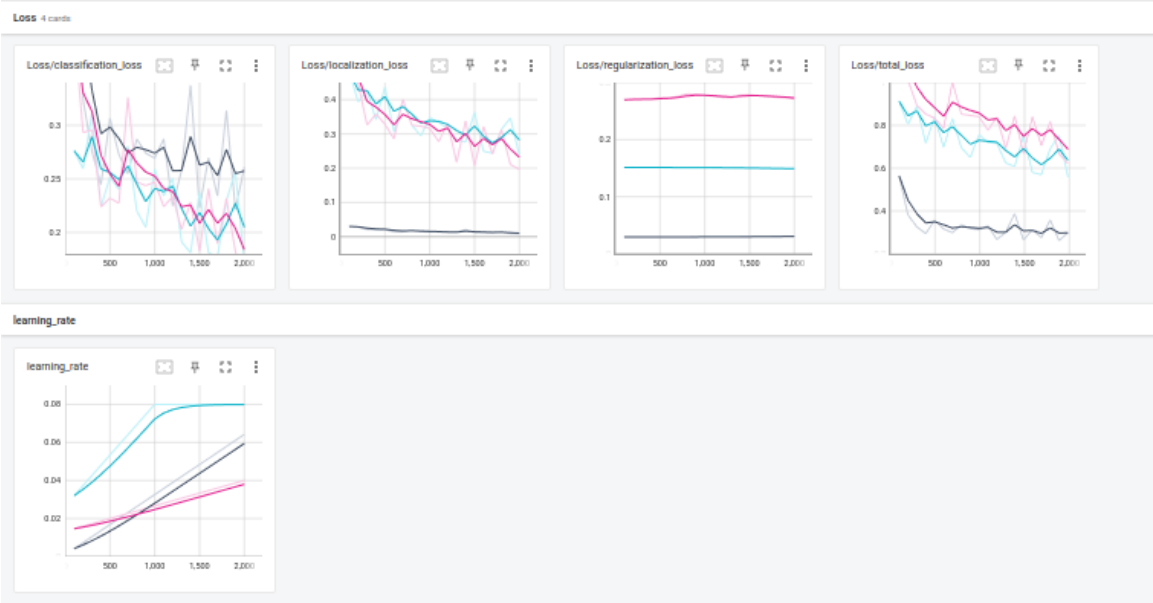
So, after analyzing we will notice that in the 3 models the training loss is less than the validation loss which is good and means that the model is trained on the data set and can distinguish and classify new data, but the most important metric to us here is the precision and recall to indicate how well the model can detect the objects.

As shown in the graphs for the models, according to the numbers the best model for this data set is model 2 the mobile net.

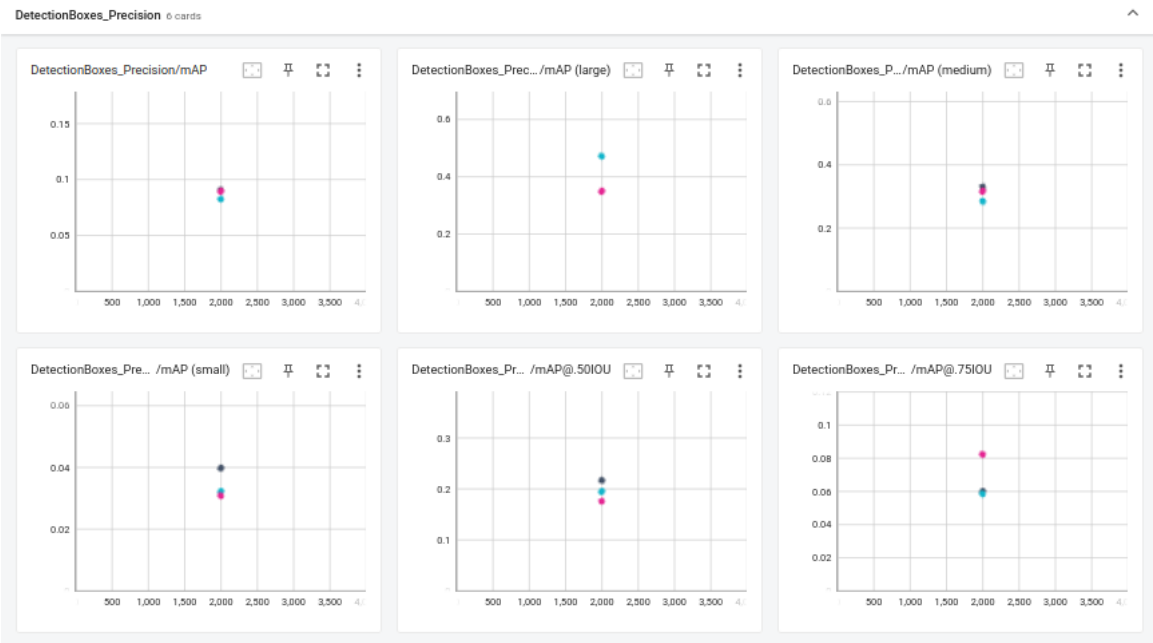
And to compare the three models and see for ourselves there are the 3 graphs for them:

For the 3 Models:

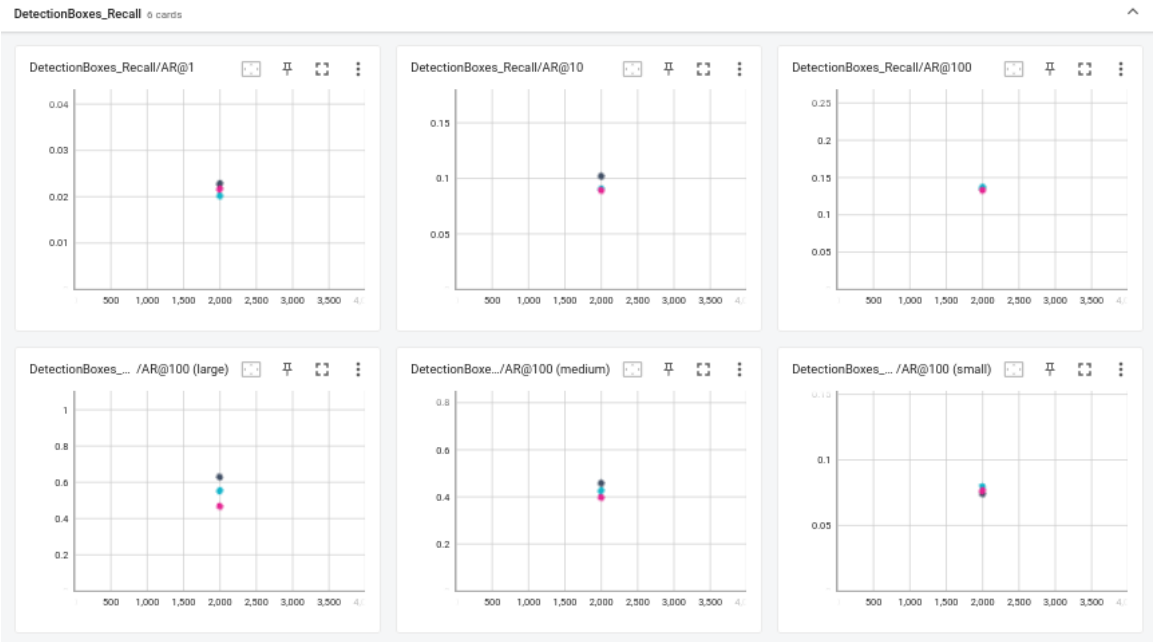
Loss:



Precision:



Recall:



How can we improve the performance of the tested models further?

To further improve the performance of the tested models we can perform data augmentation with more than one method to increase the data and train the model more, we can also increase the number of epochs until we reach the breaking point where the overfitting occurs, we can use other optimizers.