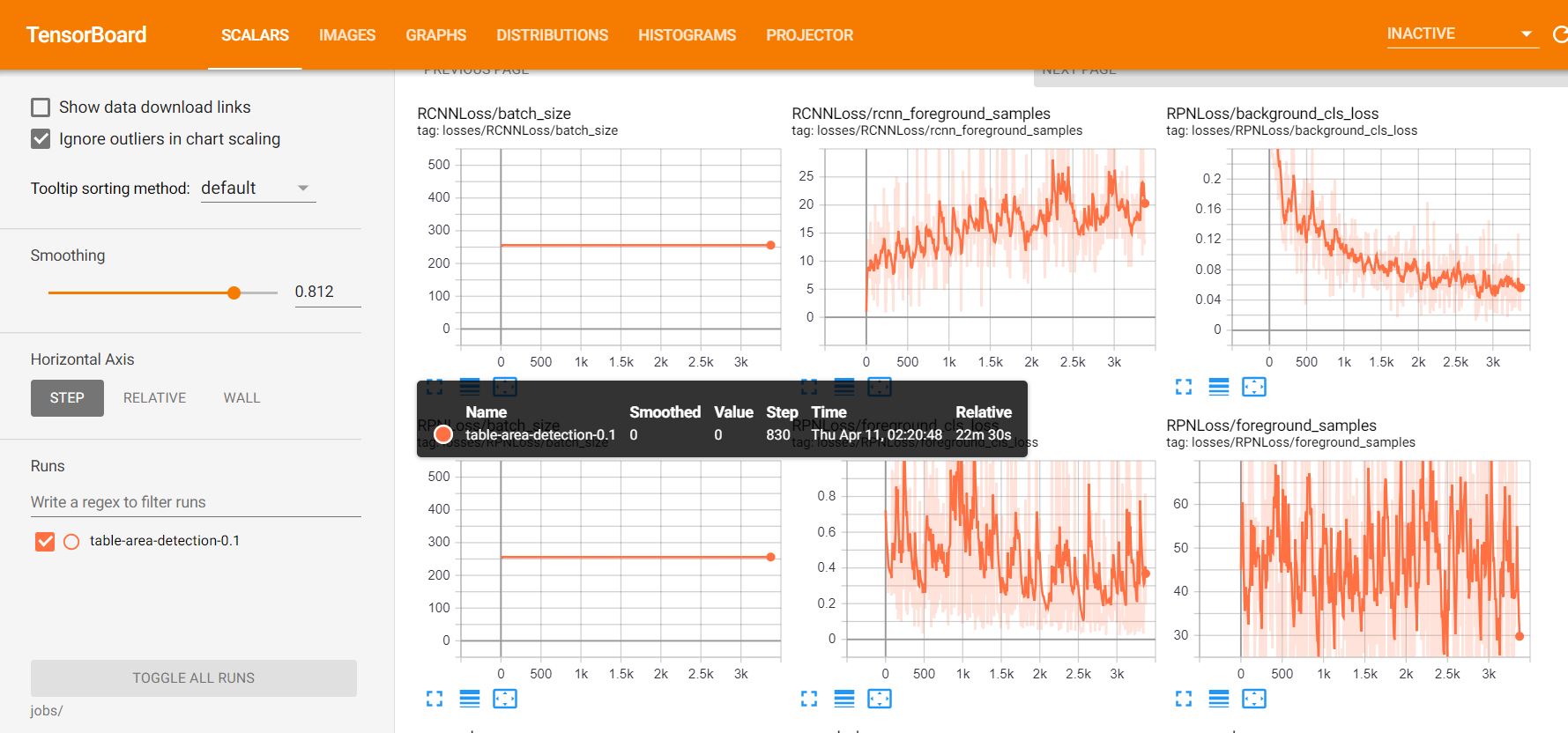
**Table Detection work visualization:**

**Tensorboard: What to look for**

Using Tensorboard for visulizing Losses, histograms and projections. First, go to the "Scalars" tab. You are going to see several *tags*, but in this case, you only should care about some of the metrics behind losses.

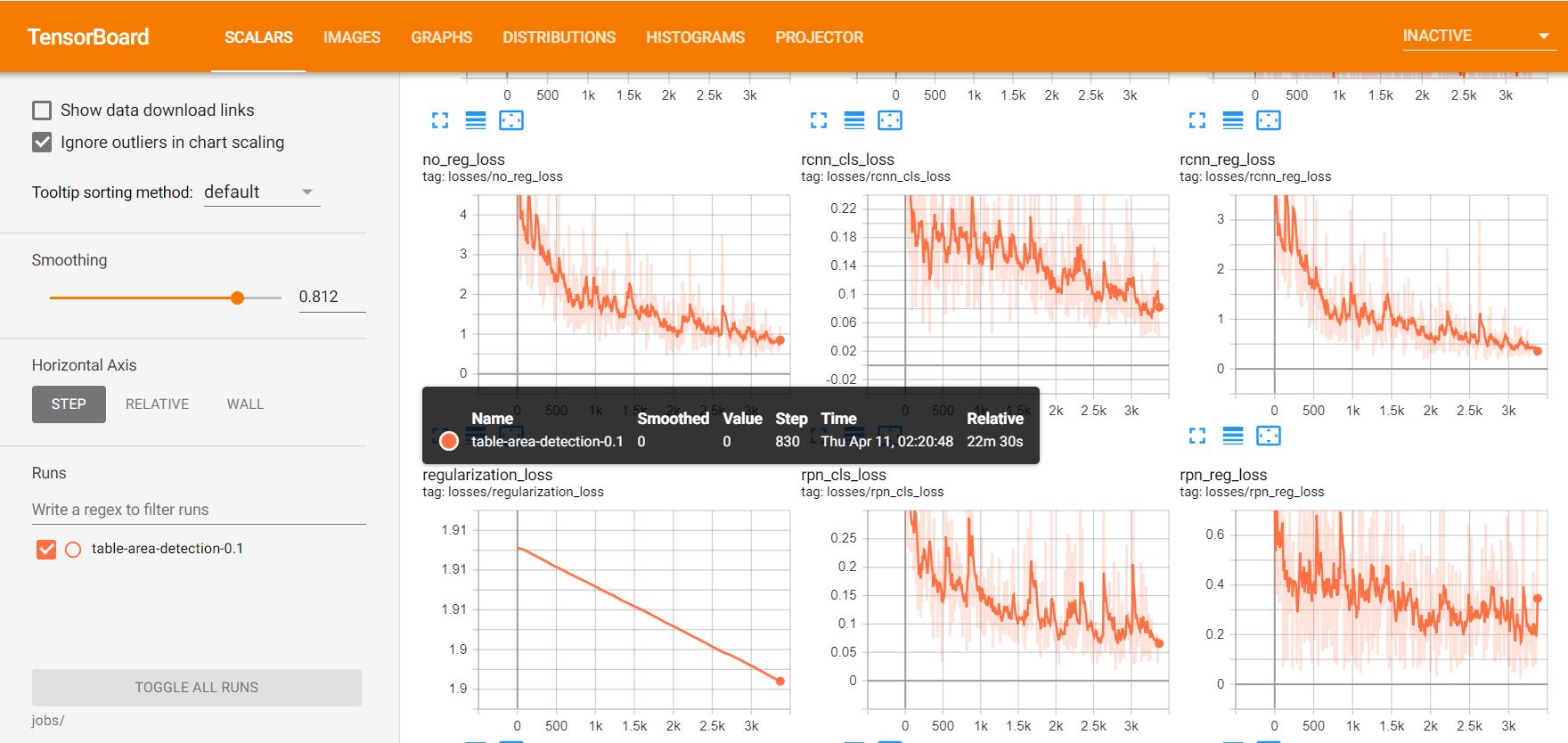
The loss is your objective function, which you want to minimize. In the case of Faster R-CNN, we have a model with a multi-objective loss, ie. the model is trying to minimize several things at the same time. This is why you will see several plots here.

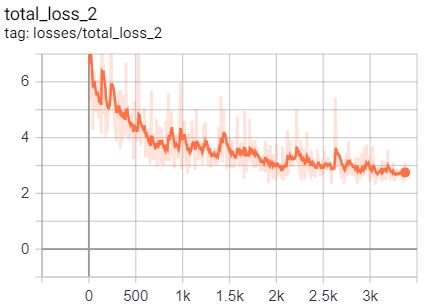


You should mostly be interested in the one called no\_reg\_loss. This is the total loss function, without including the regularization loss (which will always decrease). Therefore, it will give you a nice summary of how the training is progressing.

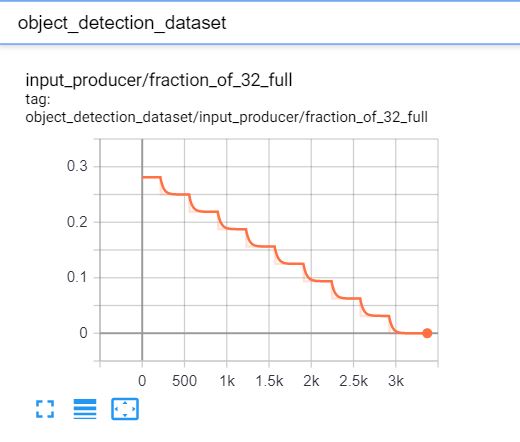
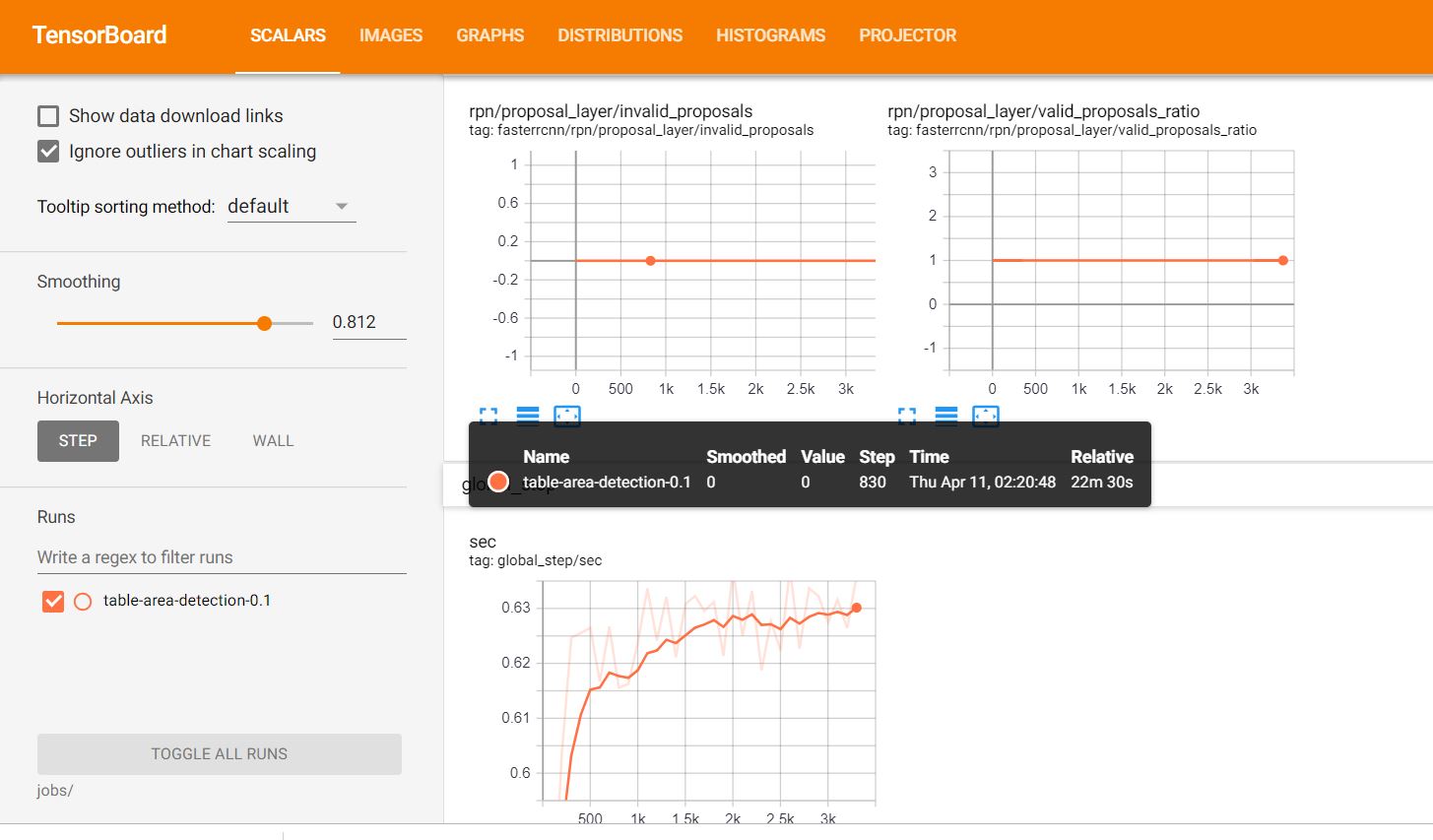
Your job is to make sure this no\_reg\_loss value is going decreasing during training.

The loss will jump around a lot, because each point corresponds to a minibatch, which in this case is a single image. A good prediction in a particular image will yield a low loss, however, if the model performed poorly another particular image, the loss will be very high.





To help you notice the trend, you can set **Smoothing** to a higher value. For example, setting it so 0.95 the plots now look like this:



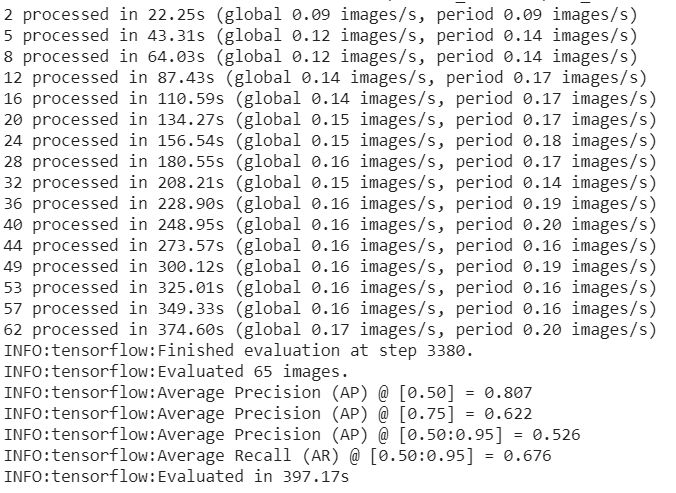
**Evaluating models:**

Mean Average Precision (mAP) is the metric commonly used to evaluate object detection task. It computes how well your classifier works across all classes. What you need to know is that mAP will be a number between 0 and 1, and the higher the better. Moreover, it can be calculated across different IoU (Intersection over Union) thresholds. For example, Pascal VOC challenge metric uses 0.5 as threshold (notation mAP@0.5), and COCO dataset uses mAP at different thresholds and averages them all out (notation mAP@[0.5:0.95]). Luminoth will print out several of these metrics, specifying the thresholds that were used under this notation.

In Luminoth, lumi eval will make a run through your chosen dataset split (ie. validation or test), and run the model through every image, and then compute metric mAP. Luminoth should now load the last available checkpoint, and start processing images.

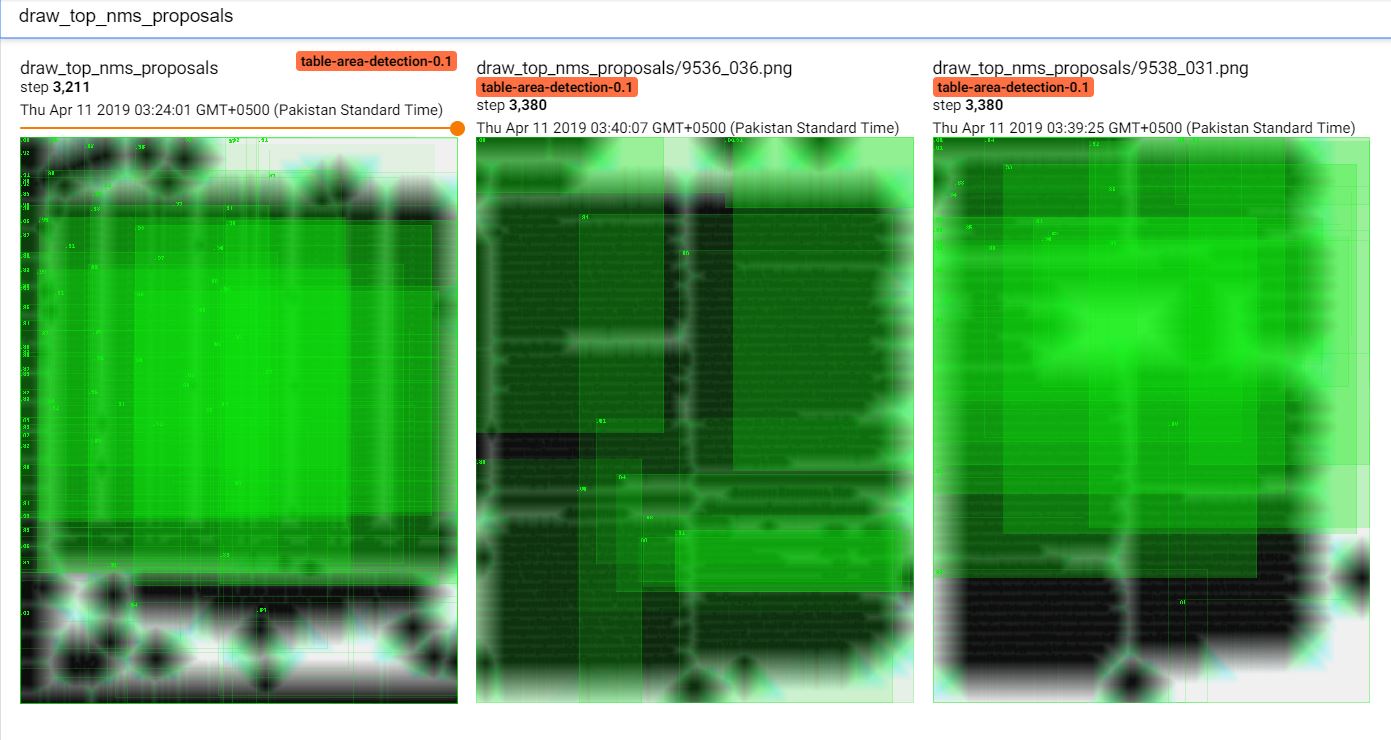
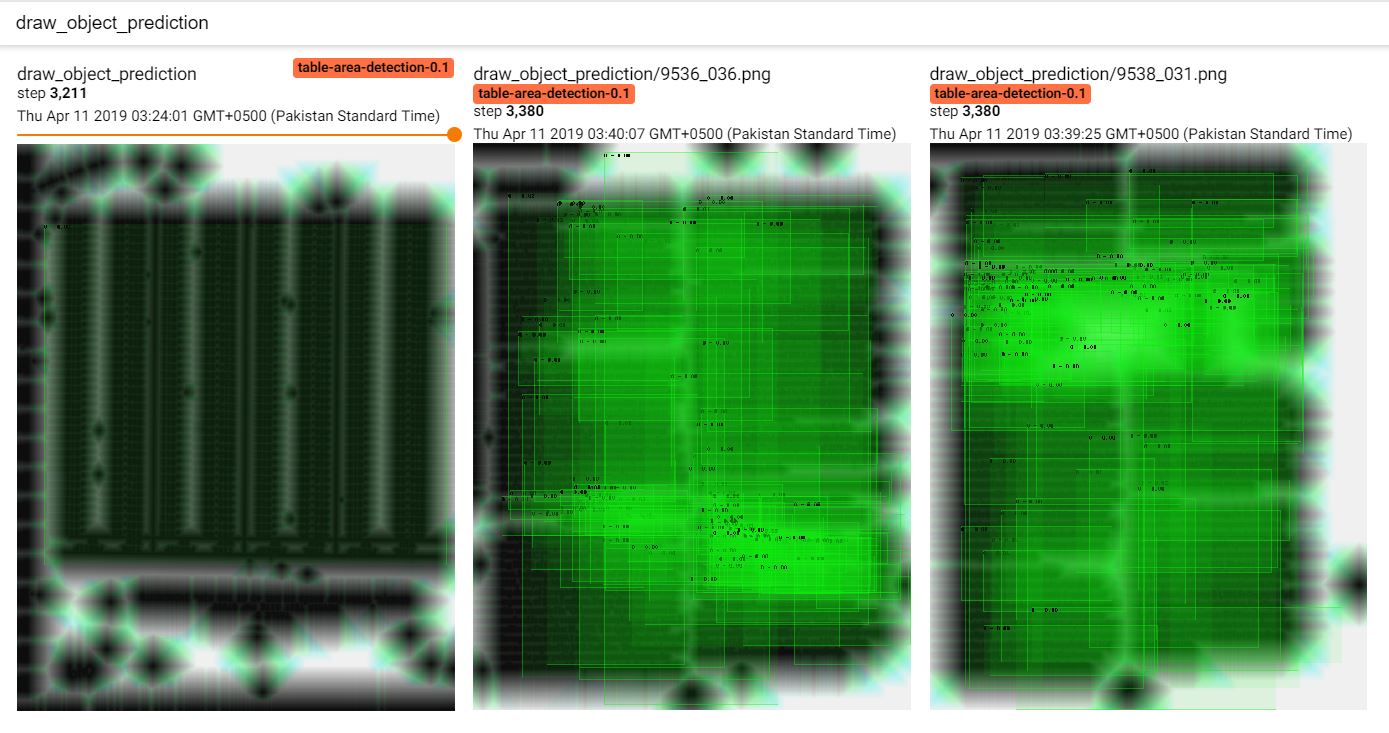
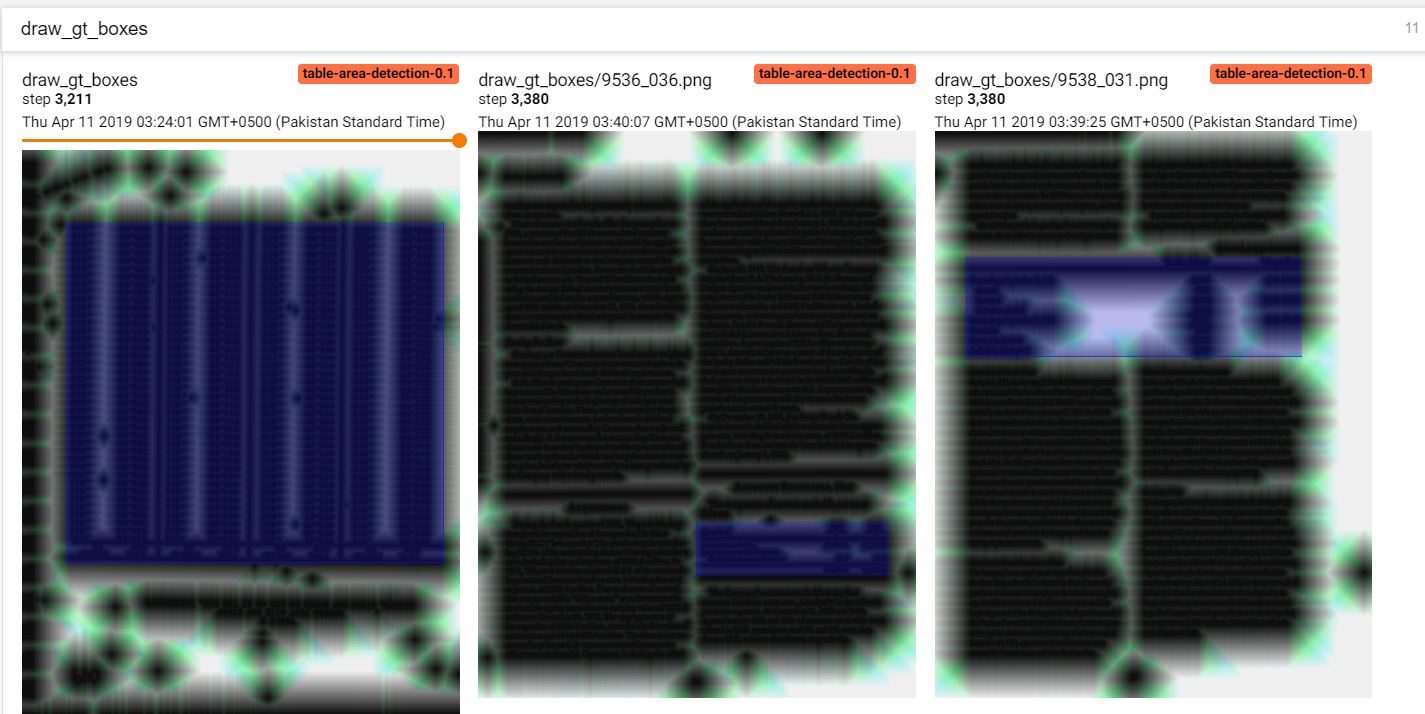
You can check it for more details. [https://medium.com/@jonathan\_hui/map-mean-average-precision-for-object-detection-45c121a31173]

mAP (mean Average Precision) for Object Detection





Images Input Boxes vs predicted:



Gradient Steps throughout Iterations:



Trained Model Points Projection:

