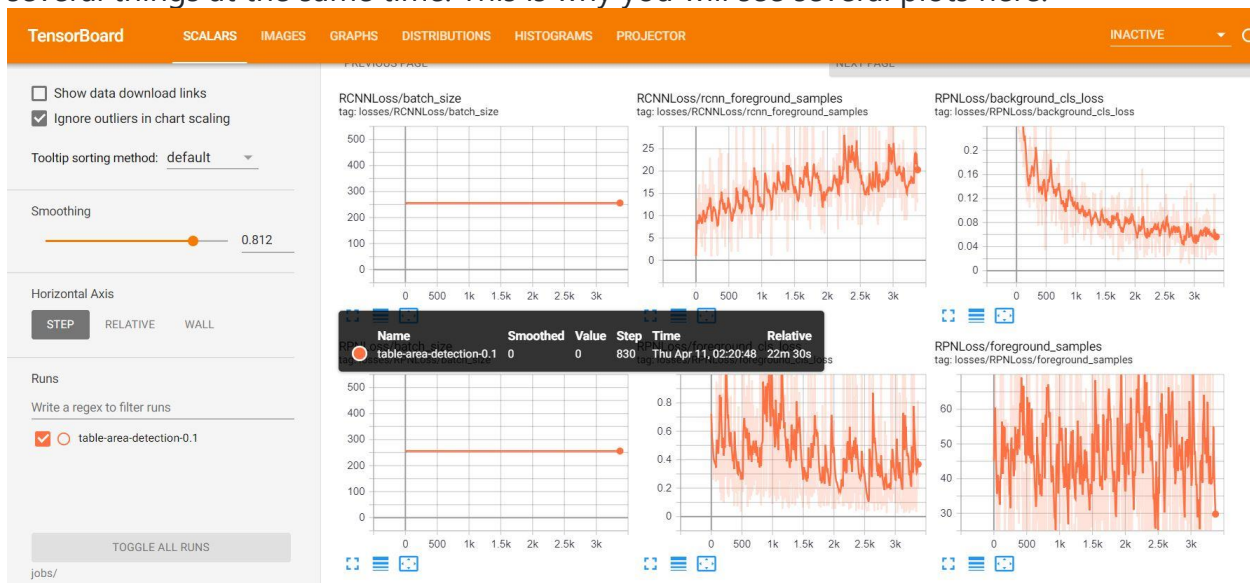


Table Detection work visualization:

Tensorboard: What to look for

Using Tensorboard for visualizing 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.

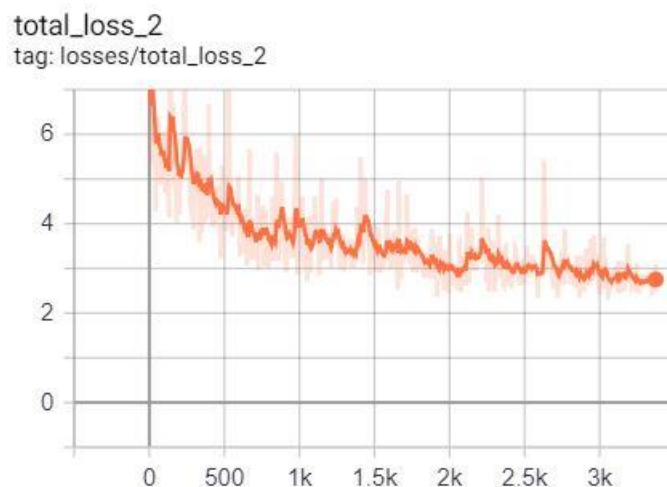
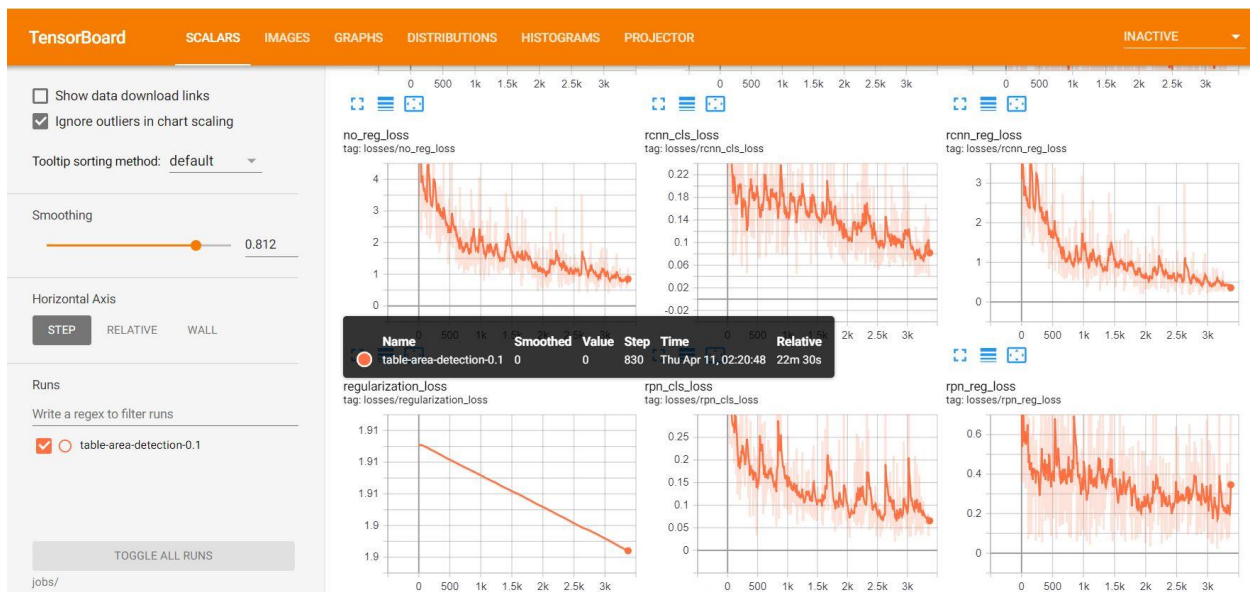


You should mostly be interested in the one called `no_reg_loss`. This is the total loss function, without including the regularization loss (which will always decrease).

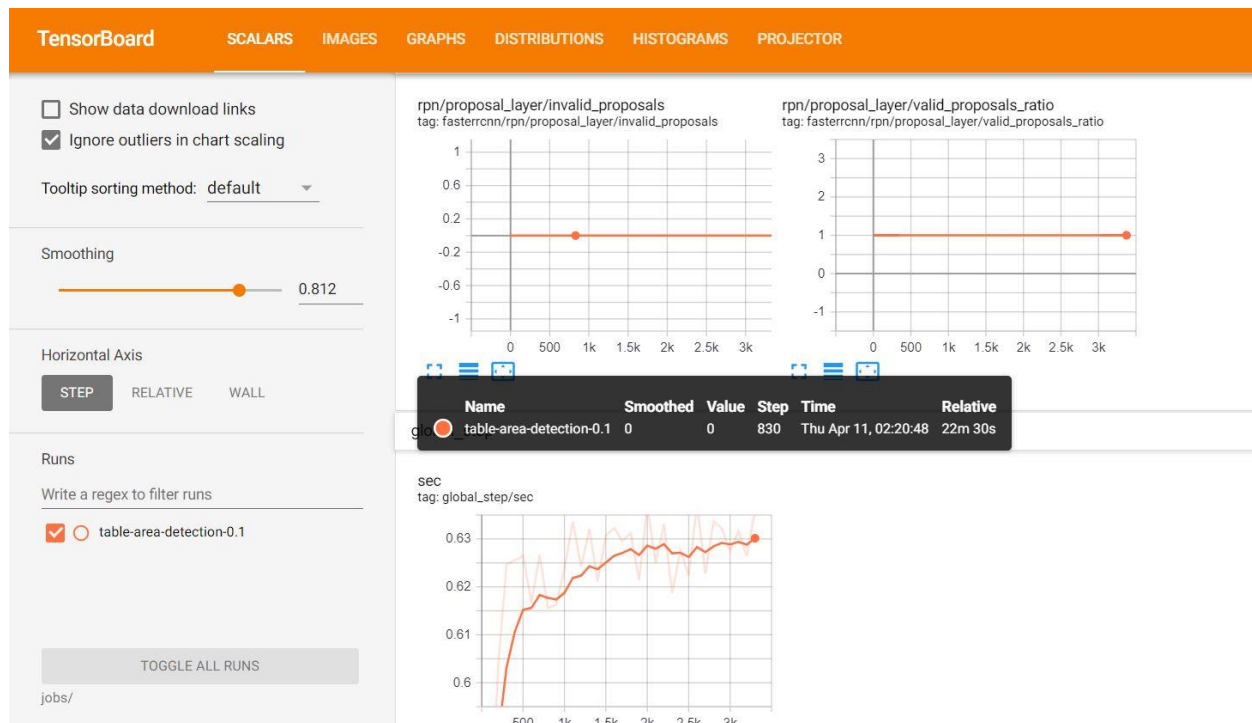
Therefore, it will give you a nice summary of how the training is progressing.

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:

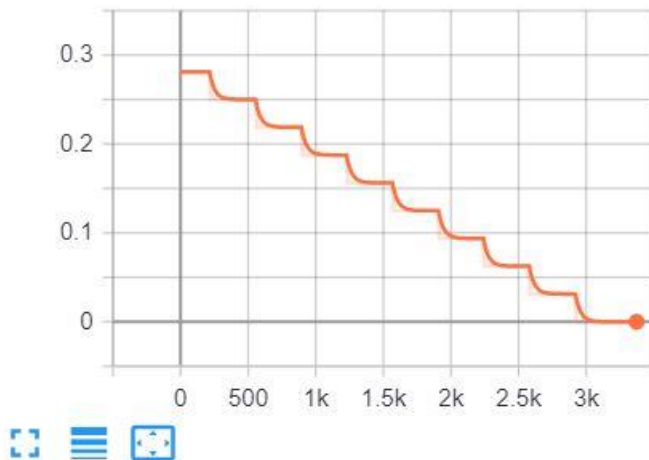


object_detection_dataset

input_producer/fraction_of_32_full

tag:

object_detection_dataset/input_producer/fraction_of_32_full



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.

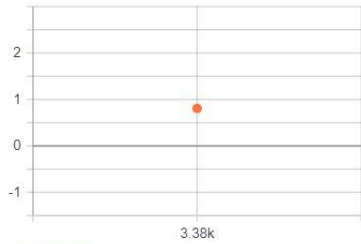
In Luminoth, `lumi eval` will make a run through your chosen dataset split (ie. validation or test), and run the model through every image, and then compute metric mAP. Luminoth should now load the last available checkpoint, and start processing images.

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

```
2 processed in 22.25s (global 0.09 images/s, period 0.09 images/s)
5 processed in 43.31s (global 0.12 images/s, period 0.14 images/s)
8 processed in 64.03s (global 0.12 images/s, period 0.14 images/s)
12 processed in 87.43s (global 0.14 images/s, period 0.17 images/s)
16 processed in 110.59s (global 0.14 images/s, period 0.17 images/s)
20 processed in 134.27s (global 0.15 images/s, period 0.17 images/s)
24 processed in 156.54s (global 0.15 images/s, period 0.18 images/s)
28 processed in 180.55s (global 0.16 images/s, period 0.17 images/s)
32 processed in 208.21s (global 0.15 images/s, period 0.14 images/s)
36 processed in 228.90s (global 0.16 images/s, period 0.19 images/s)
40 processed in 248.95s (global 0.16 images/s, period 0.20 images/s)
44 processed in 273.57s (global 0.16 images/s, period 0.16 images/s)
49 processed in 300.12s (global 0.16 images/s, period 0.19 images/s)
53 processed in 325.01s (global 0.16 images/s, period 0.16 images/s)
57 processed in 349.33s (global 0.16 images/s, period 0.16 images/s)
62 processed in 374.60s (global 0.17 images/s, period 0.20 images/s)
INFO:tensorflow:Finished evaluation at step 3380.
INFO:tensorflow:Evaluated 65 images.
INFO:tensorflow:Average Precision (AP) @ [0.50] = 0.807
INFO:tensorflow:Average Precision (AP) @ [0.75] = 0.622
INFO:tensorflow:Average Precision (AP) @ [0.50:0.95] = 0.526
INFO:tensorflow:Average Recall (AR) @ [0.50:0.95] = 0.676
INFO:tensorflow:Evaluated in 397.17s
```

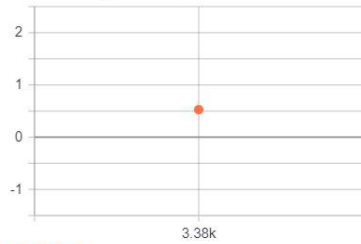
AP@0.50
tag: val_metrics/AP@0.50



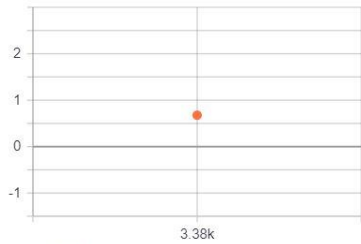
AP@0.75
tag: val_metrics/AP@0.75



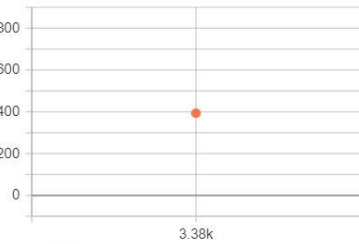
AP@[0.50:0.95]
tag: val_metrics/AP@[0.50:0.95]



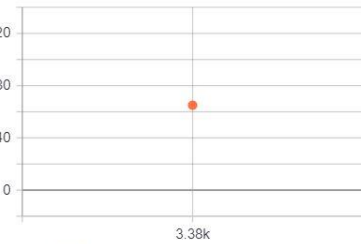
AR@[0.50:0.95]
tag: val_metrics/AR@[0.50:0.95]



evaluation_time
tag: val_metrics/evaluation_time



total_evaluated
tag: val_metrics/total_evaluated



Images Input Boxes vs predicted:

draw_gt_boxes

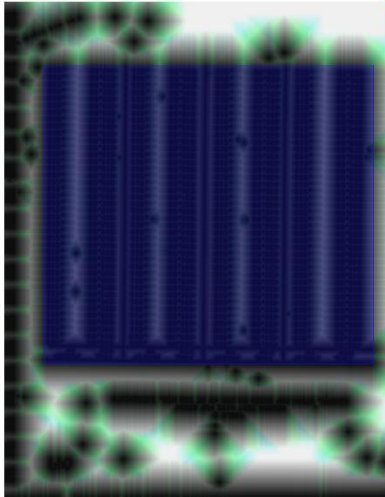
11

draw_gt_boxes

step 3,211

Thu Apr 11 2019 03:24:01 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1




draw_gt_boxes/9536_036.png

step 3,380

Thu Apr 11 2019 03:40:07 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1

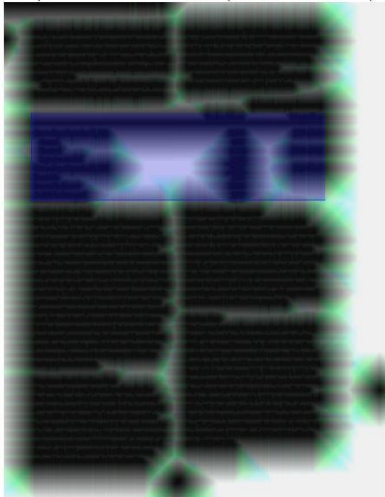


draw_gt_boxes/9538_031.png

step 3,380

Thu Apr 11 2019 03:39:25 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1

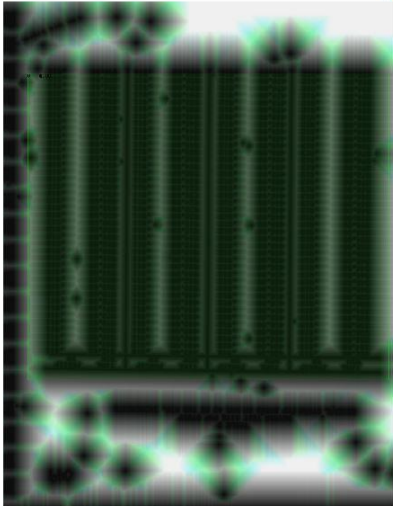


draw_object_prediction

step 3,211

Thu Apr 11 2019 03:24:01 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1

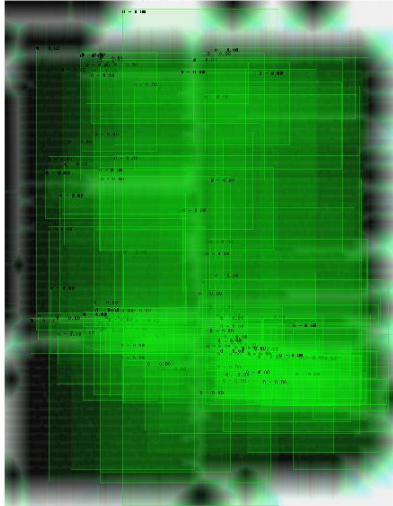


draw_object_prediction/9536_036.png

step 3,380

Thu Apr 11 2019 03:40:07 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1

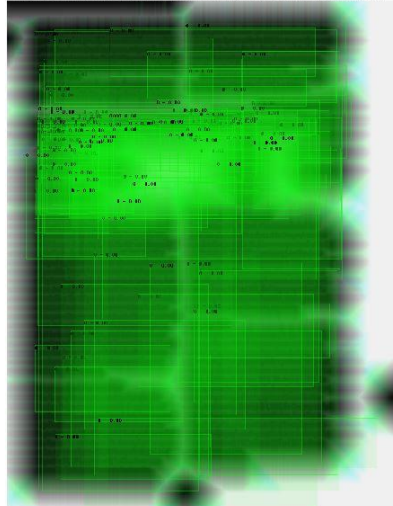


draw_object_prediction/9538_031.png

step 3,380

Thu Apr 11 2019 03:39:25 GMT+0500 (Pakistan Standard Time)

table-area-detection-0.1

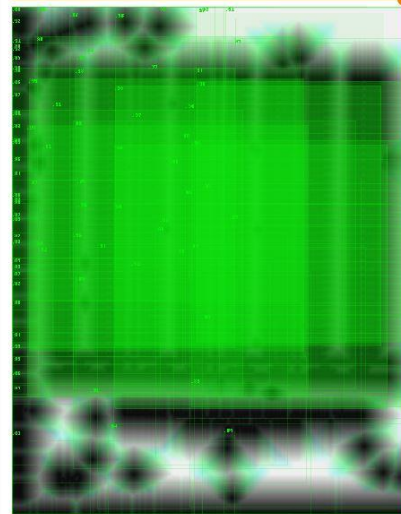


draw_top_nms_proposals

draw_top_nms_proposals
step 3,211

table-area-detection-0.1

Thu Apr 11 2019 03:24:01 GMT+0500 (Pakistan Standard Time)

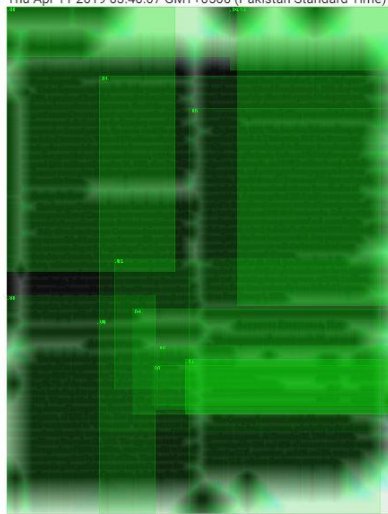


draw_top_nms_proposals/9536_036.png

table-area-detection-0.1

step 3,380

Thu Apr 11 2019 03:40:07 GMT+0500 (Pakistan Standard Time)

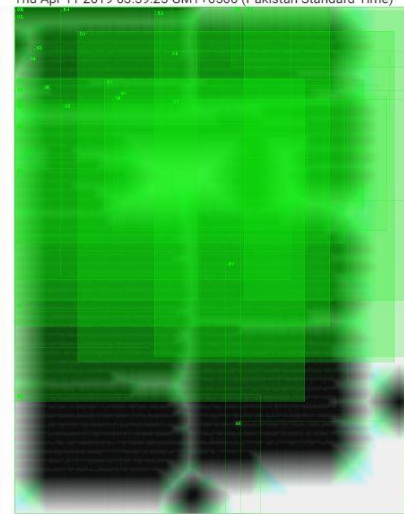


draw_top_nms_proposals/9538_031.png

table-area-detection-0.1

step 3,380

Thu Apr 11 2019 03:39:25 GMT+0500 (Pakistan Standard Time)

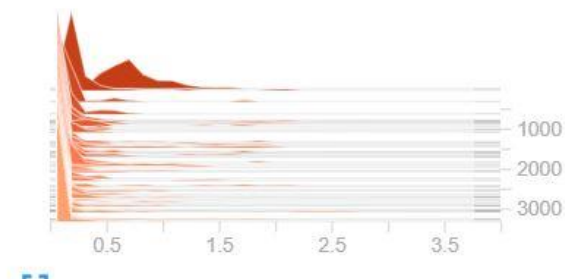


Gradient Steps throughout Iterations:

losses

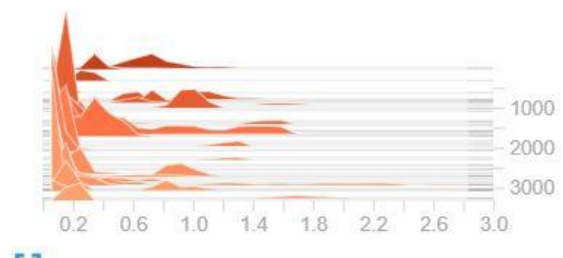
losses/RPNLoss/background_cls_loss_1

table-area-detection-0.1



losses/RPNLoss/foreground_cls_loss_1

table-area-detection-0.1



Trained Model Points Projection:

