**CS771 - Homework 3, FCOS implementation**

Satya Sai Srinath (sgnamburi@wisc.edu), Debarshi Deka (ddeka@wisc.edu)

**Implementation details:**

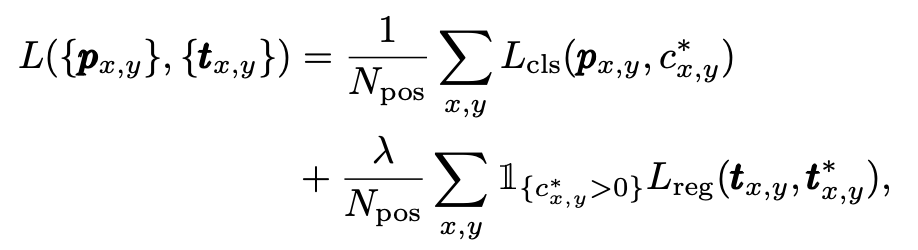
**FCOS Classification Head:**

This is very straightforward. All we did is to pass the features of a particular feature pyramid to the convolutional layers and stack all the outputs.

**FCOS Regression Head:**

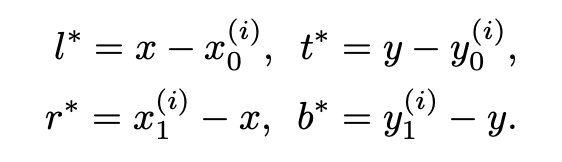
This is also very straightforward. Here, we will have to take care of both the regression output (which has 4 values) and the centerness output (which comes from 0-1). All we did is to pass the features of a particular feature pyramid to the convolutional layers and stack all the outputs into 2 separate lists.

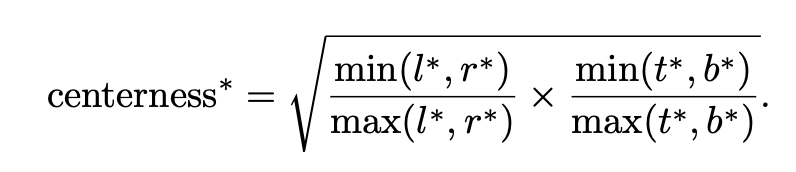
**FCOS Compute loss:**



This is pretty tricky. It basically has 3 terms, one classification loss (sigmoid focal loss), one regression loss (GIoU loss) and one for centerness loss (BCE loss).

Only when the class is predicted i.e (c\*x,y > 0), then only the prediction of regression targets and centerness makes sense. Else, the point will be considered as background. When there are multiple predicted boxes for one ground truth, we take the targets which have a minimum area.

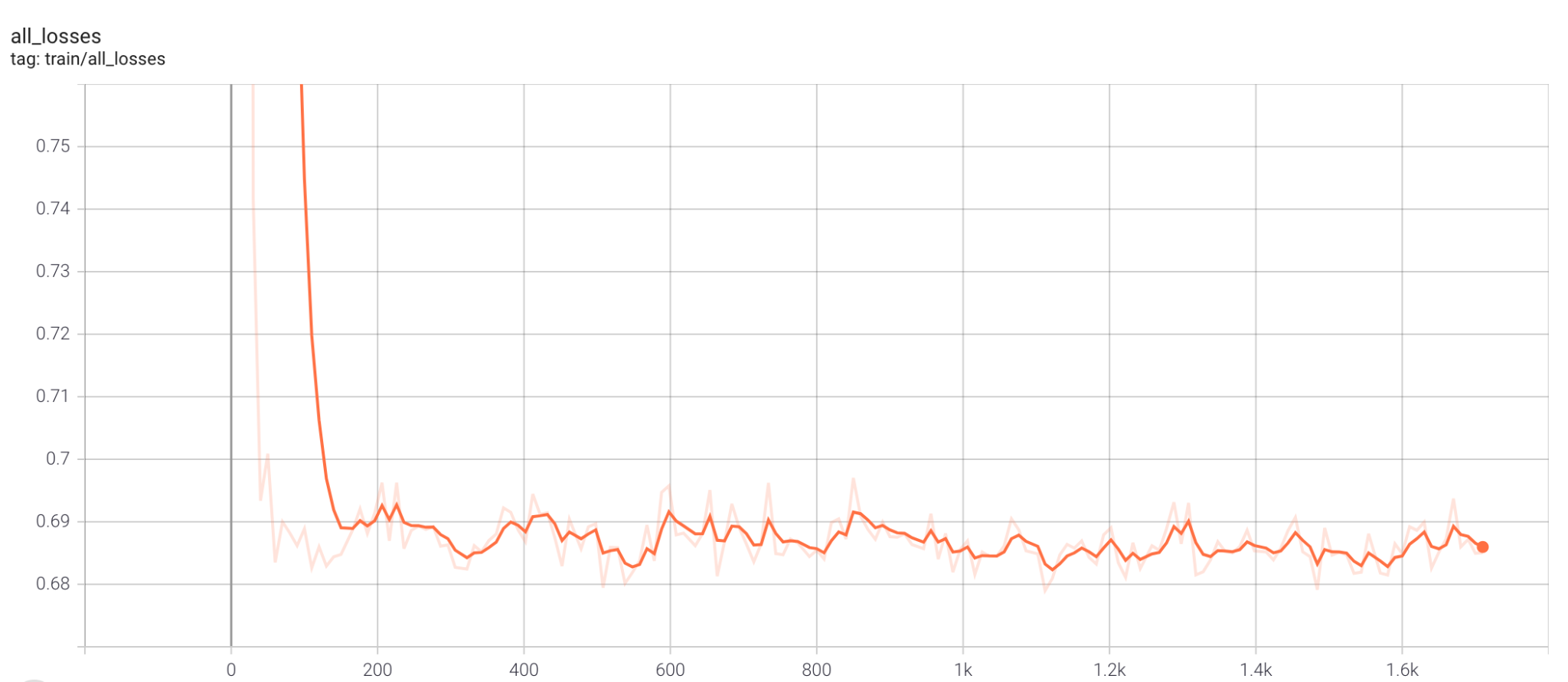
**FCOS Inference:**

****

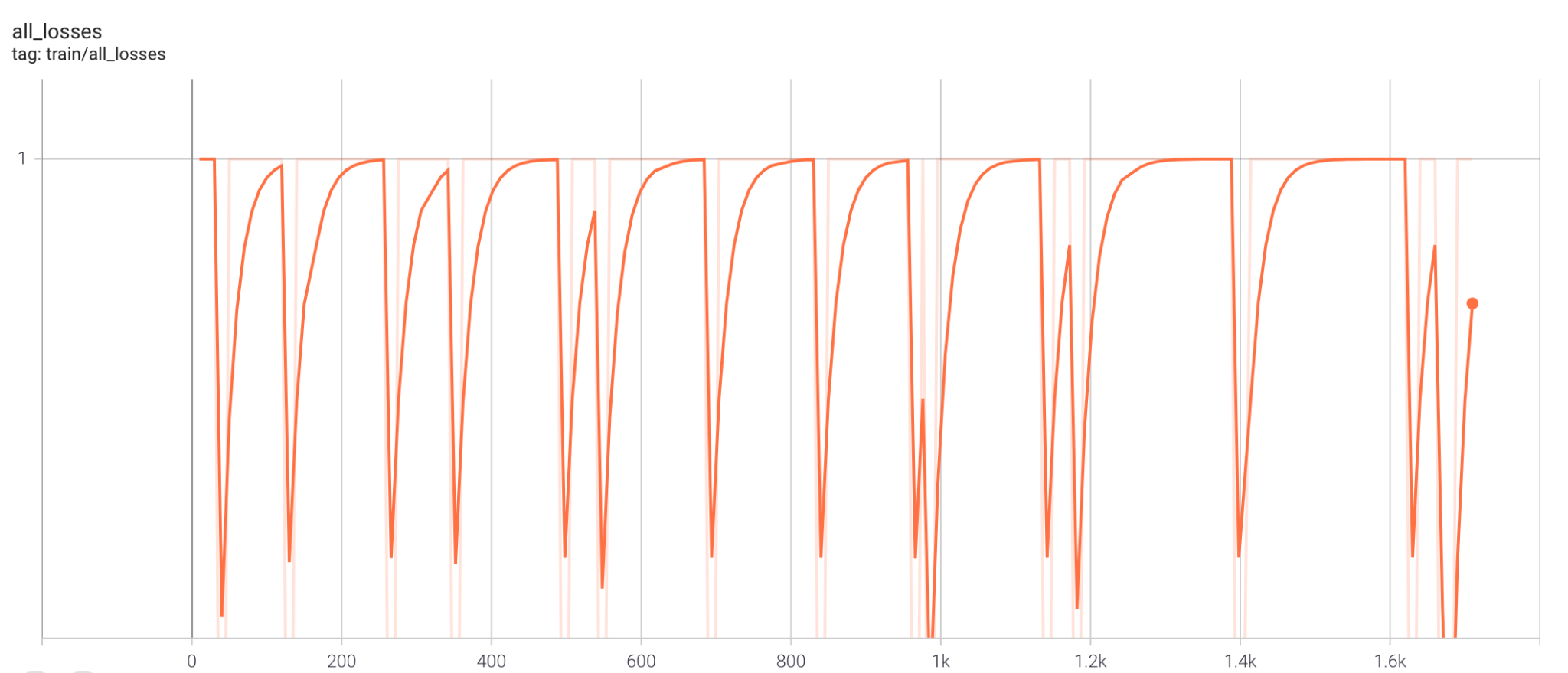
Given an input image, we forward it through the network and obtain the classification scores and the regression prediction for each location on the feature maps Fi . Following [15], we choose the location with classification score > 0.05 as positive samples and invert the above equation to obtain the predicted boxes (where \* are the predicted coordinates, x,y are centers of the box, (i) correspond to the ground truth labels for a box). While inferencing, non-max suppression is performed to remove the boxes which has threshold < nms\_threshold initialized in code.

**Results (visualization of plots):**

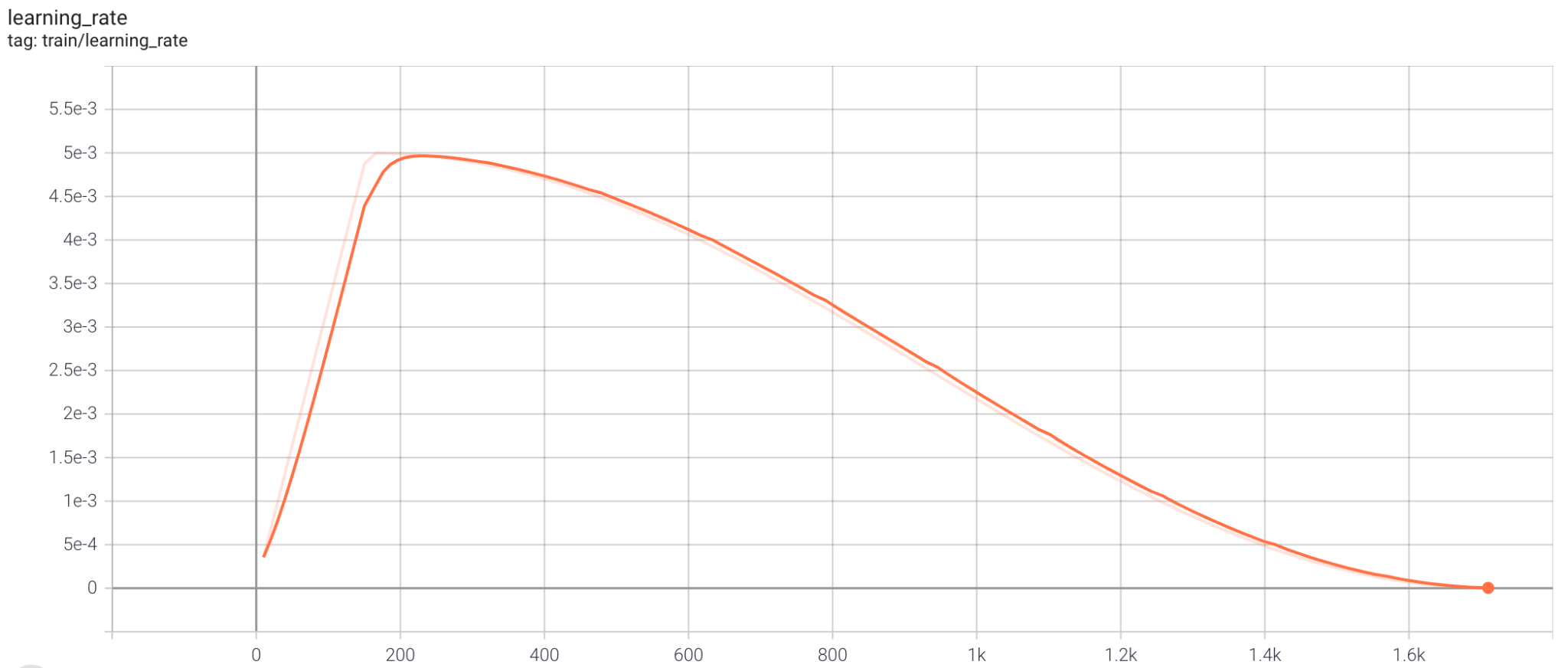
Classification loss



Centerness loss



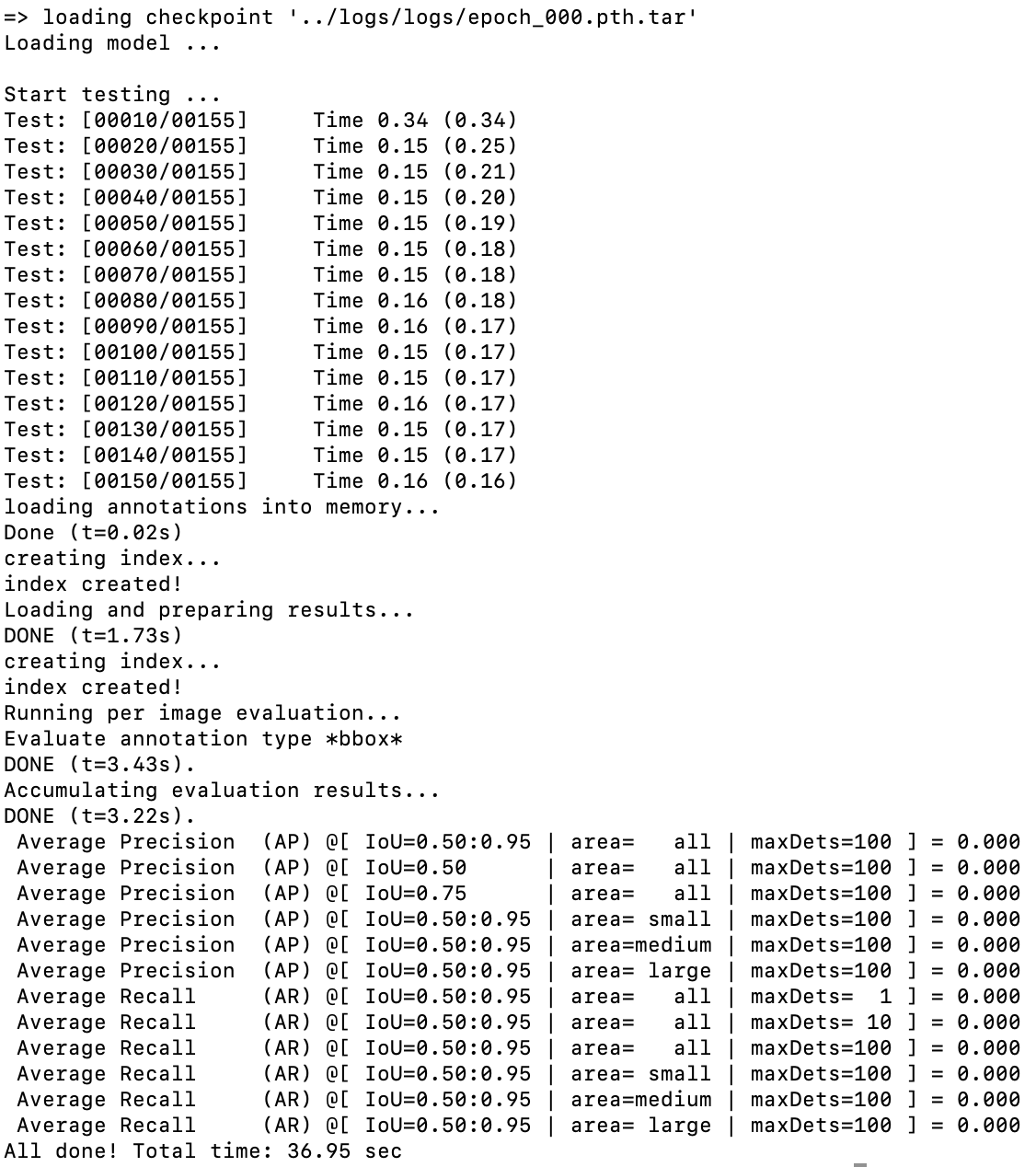
Regression loss



Learning rate (decaying)

**Evaluation:**

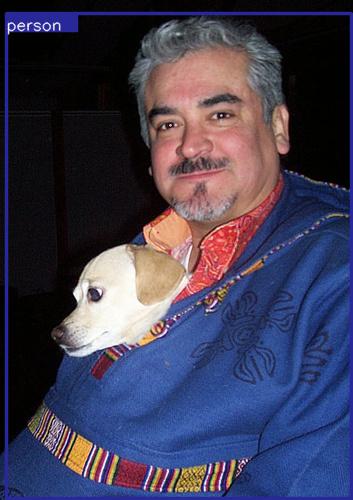
The evaluation command is run on the .pth.tar file and the json file is able to get generated. An example output from terminal when epoch\_0000.pth.tar is given as input to evaluate and achieved an **AP of 0.95 at IOU@0.5**



**Bonus Question (Visualization of the bounding boxes):**

For the bonus question, we tried to visualize the bounding boxes of the images present in COCO dataset. For that, we chose Pytorch version of FCOS Resnet 50 with standard weights. The draw boxes function takes the coordinates and joins them thus forming a box. Refer to ***visualize\_boxes.py*** for more details. And here are some results.

**Note:** We can also use the results\_json to obtain the bounding boxes by passing the coordinates to draw boxes function.





**Bonus Question (Aggressive Augmentations):**

Before giving the images to the model, CutMix and Mixup can be performed to see how the performance varies as the percentage of the augmentation changes.

**Contributions:**

**Satya Sai Srinath** - FCOS classification and regression forward functions, FCOS Loss function, Inference function, Bonus question, Report

**Debarshi Deka** - Critique writeup, FCOS Loss function, Inference function, Report