

16-824 Visual Learning and Recognition (HW2)

1 Question 0

Answer all parts of Question 0 from README.md

Q0.1

Train class

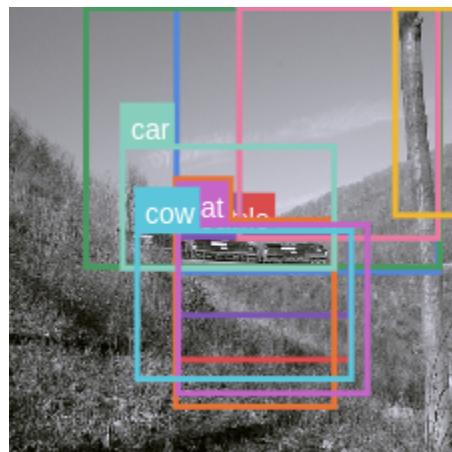
Q0.2

Select the top n highest scoring boxes from rois. For task_0 we select the top 10 scoring proposals

Q0.3



Q0.4



1

Question 1

Q1.1

For alexnet class: features is the features part of alexnet, I used torchvision models for this, removing the last maxpool layer. In the forward function the input image is passed through classifiers and features. The function of this is to extract features from images. The final maxpool layer is to extract the location of the object, which will be the maximum value in the heatmap.

In the localizer_alexnet function, I initialize the weights with xavier for features and classifier if the pretrained argument is false, else pretrained weights are applied to the features part while xavier weights are applied to the classifier layer.

In tasks2.py, the loss function used is BCELoss(), optimizer used is SGD with lr=0.01, momentum=0.9

In forward function the output of the net is passed through a sigmoid to make it between 0 and 1, and BCE loss is calculated.

WANDB is used to plot metric1, metric2, loss etc.

Q1.2

The output resolution of heatmap before max pool: (20, 29, 29)

Output after max pool: 20*100*100

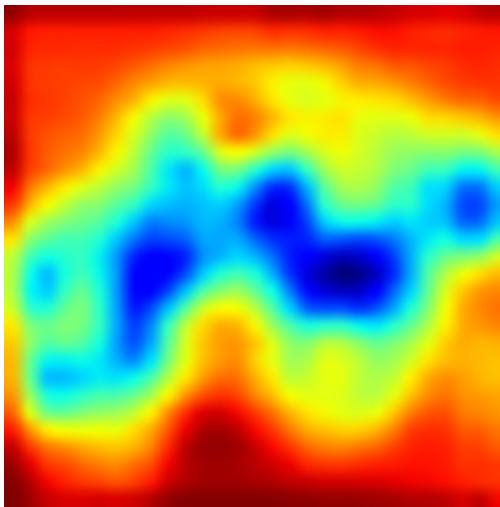
Q1.3

Plot the following

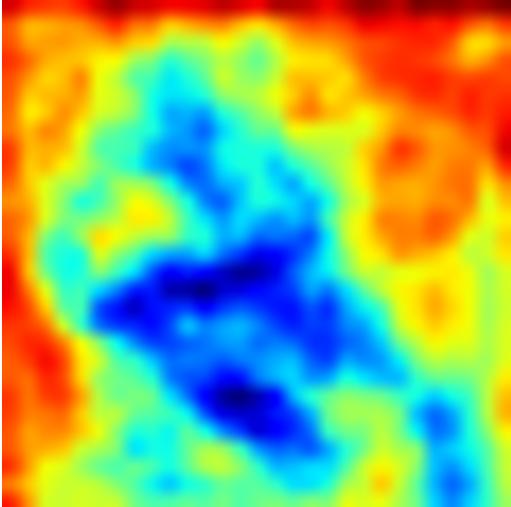
- Plot training loss curve.



Epoch 1

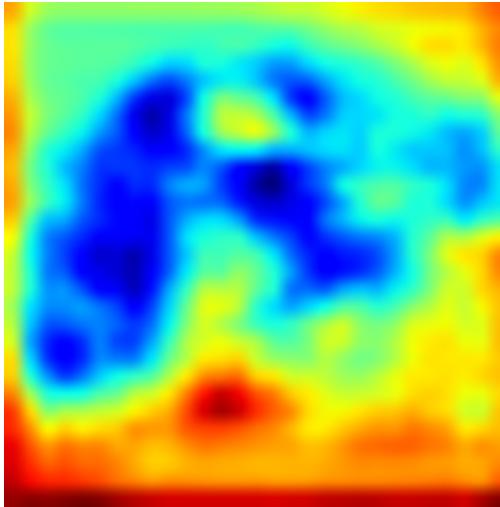


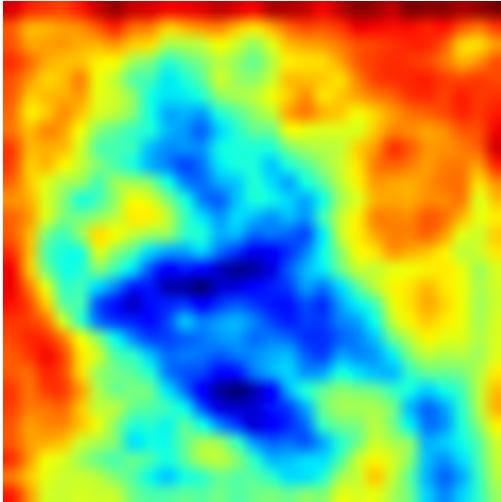
car



Person

Epoch 2





Q1.4

The loss being used here is BCE Loss, all the most of labels are 0 compared to one(because only 2 or 3 classes present in an image out of 20, so 2 or 3, 1's in labels and other 17 zeroes. So when correct predictions become stronger, error decreases. As multiplying by 0 does not affect error, BCE error does not reflect the amount of misclassification.

Q1.5

Answer and describe functionality of the completed TODO blocks

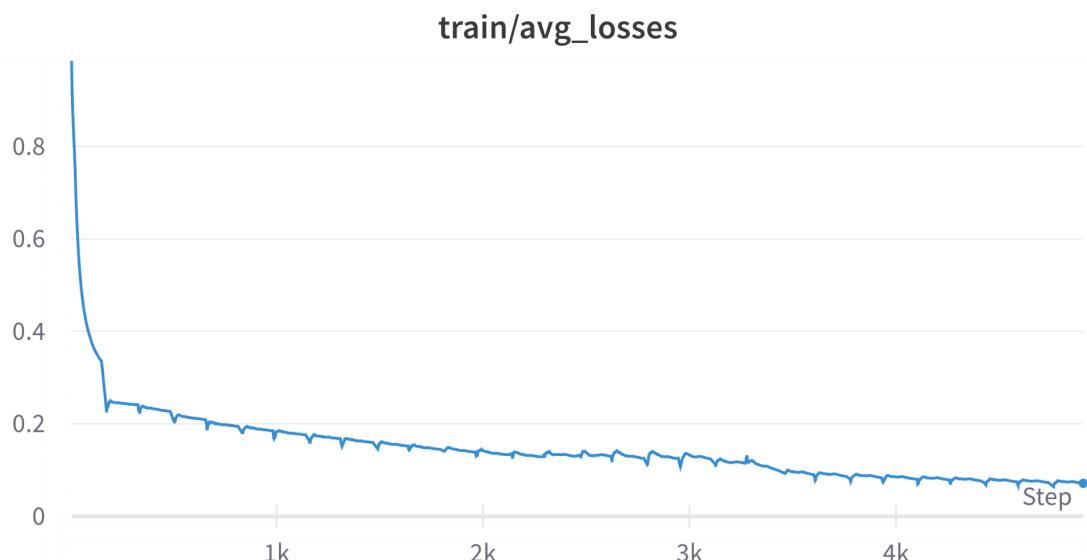
So metric 1 is mean average precision, which calculates the average of average precision among 20 classes. Average precision for a class is the area under the precision recall curve, an increase in map(metric1) shows the quality of detection.

Metric2 measures the recall, that is $TP/(TP+FN)$, it is 0 initially because of no FN, but then it decreases and increases. Recall shows that the network is learning although the loss does not change much, when recall increases. Loss only reflects the TP and TN as FN, FP have labels zero which is not reflected in BCE loss.

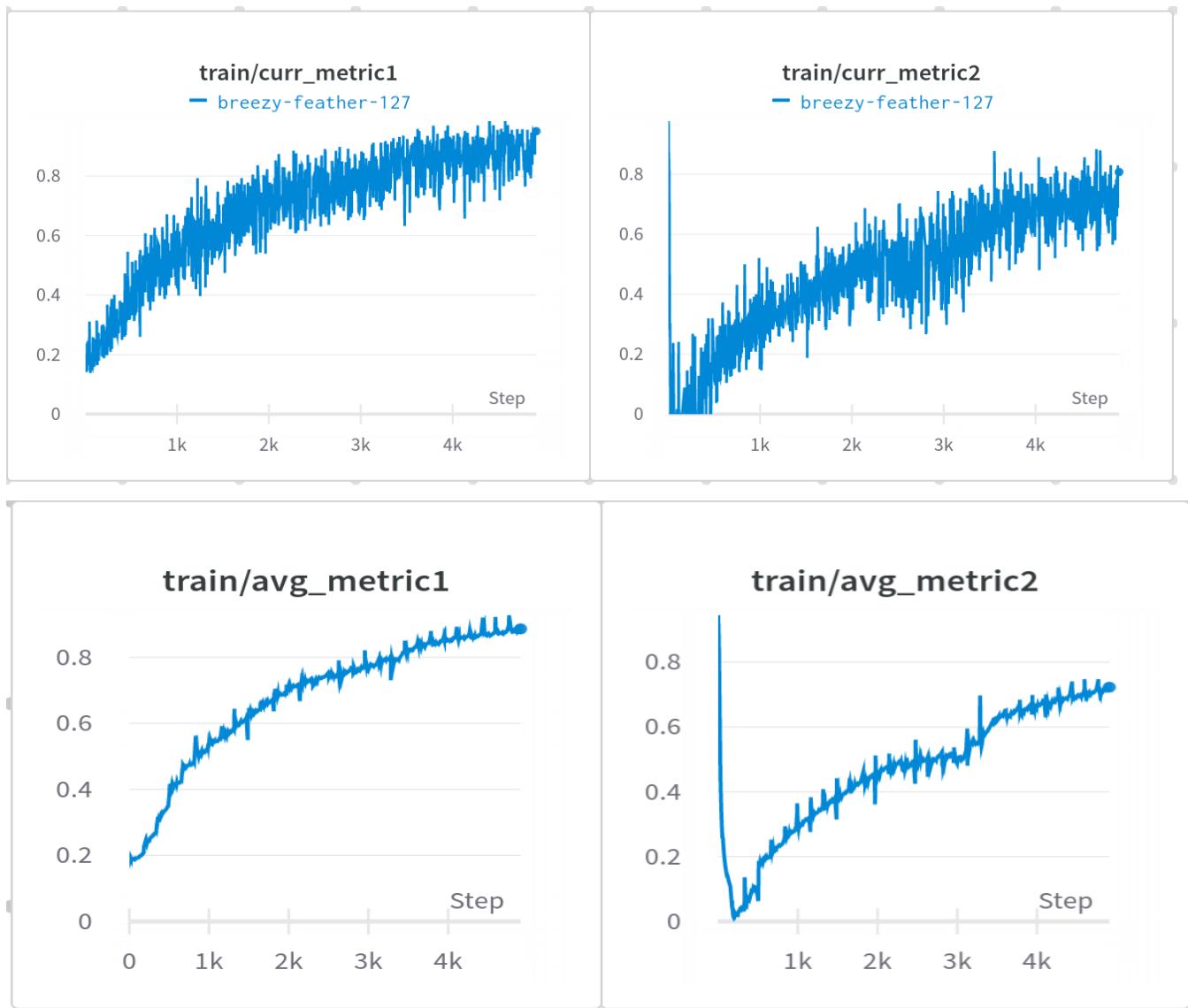
Q1.6 Part 1

Plot the following figures.

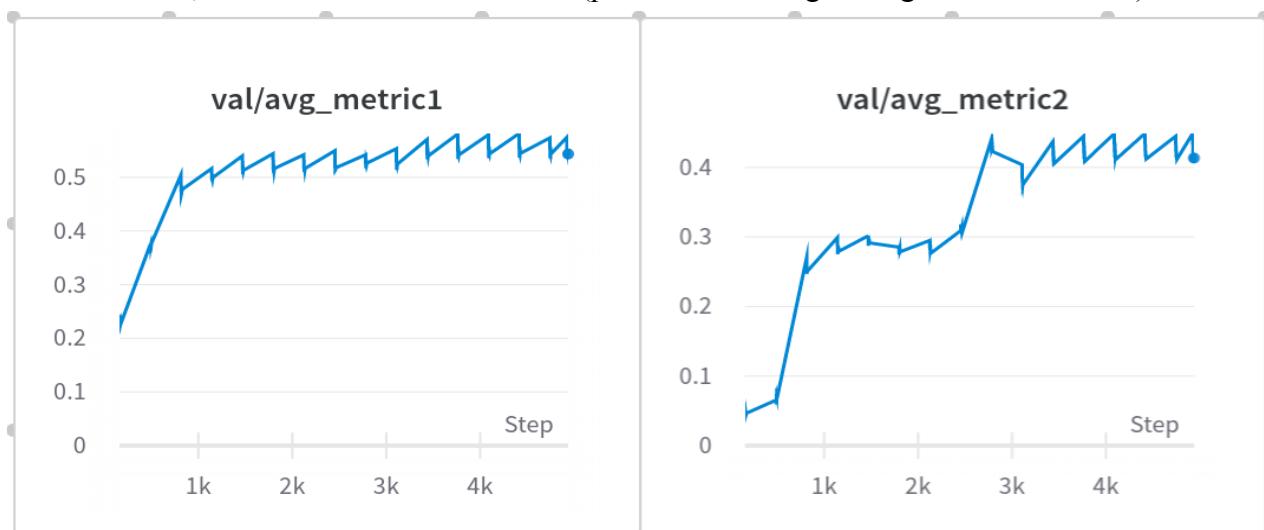
1. The train loss against iterations.



2. Plot metric1, metric2 on the training set (plot both in a single image with 2 columns)

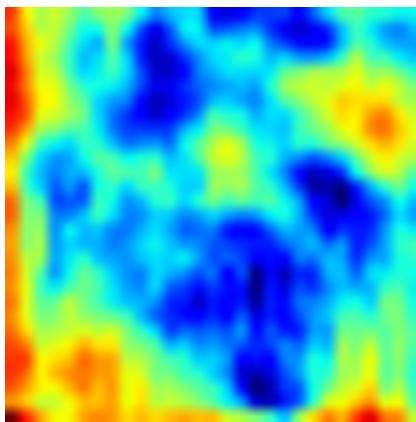


3. Plot metric1, metric2 on the validation set (plot both in a single image with 2 columns)

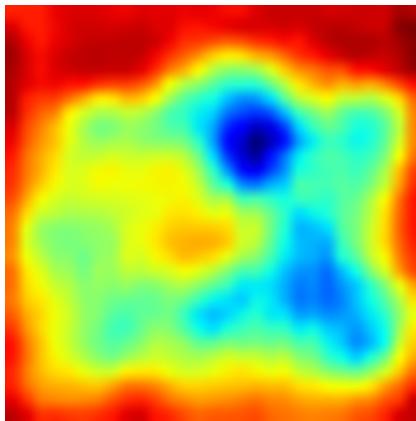


Q1.6 Part 2

Plot wandb images for heat maps for the first logged epoch. Please plot the image and it's heatmap as one image



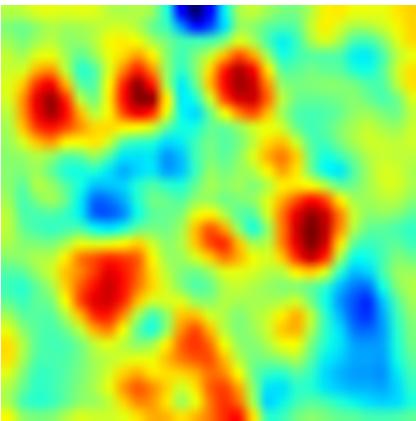
person



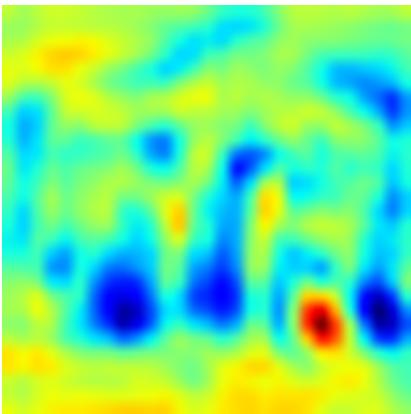
dog

Q1.6 Part 3

Plot wandb images for heat maps for the last logged epoch. Please plot the image and it's heatmap as one image with multiple columns.



person

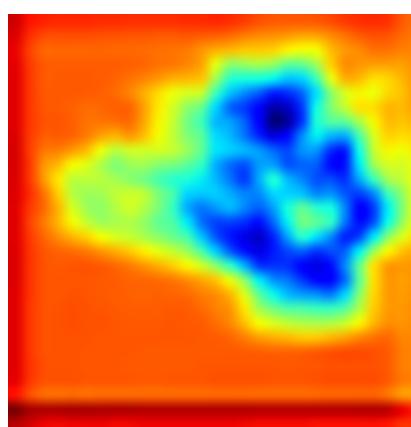


dog

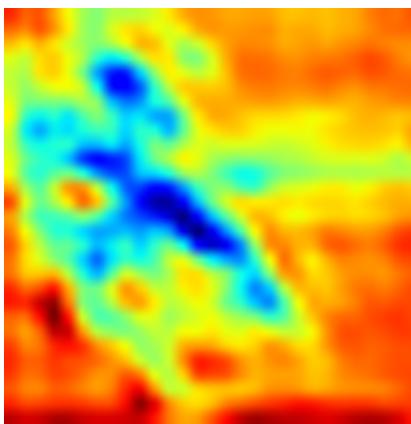
2

Q1.6 Part 4

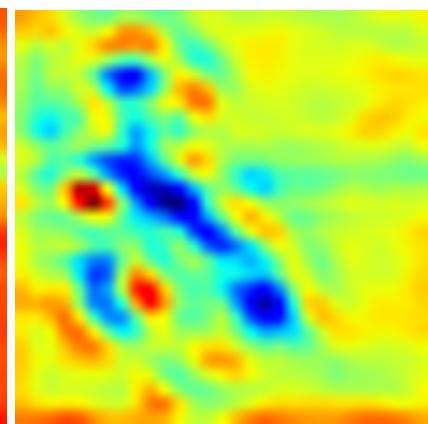
Plot 3 randomly chosen validation images and heat maps. As before plot each image in one row with multiple columns, hence a total of 3 rows.



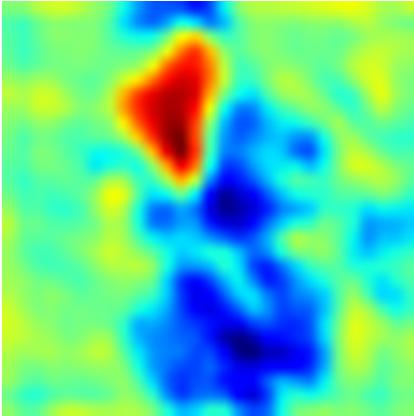
aeroplane



Motorbike



person



person

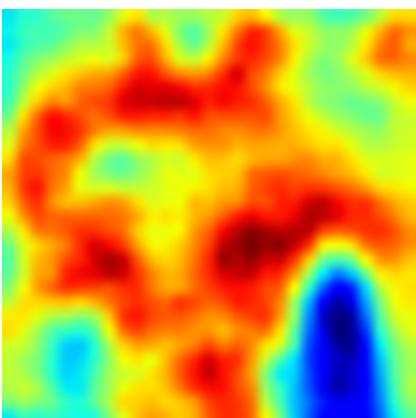
Q1.6 Part 5

- Report training loss: **0.07992**
 - Training metric 1: **0.8718**
 - Training metric 2: **0.68891**
 - Validation metric 1: **0.55752**
 - Validation metric 2: **0.41691**
-

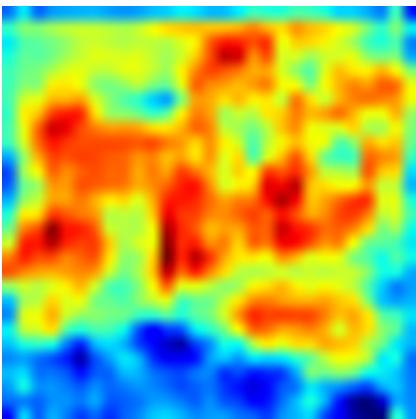
Q1.7 Part 1

Plot images and heatmaps using wandb at similar intervals as before (ensure that the same images are plotted)

Epoch 0

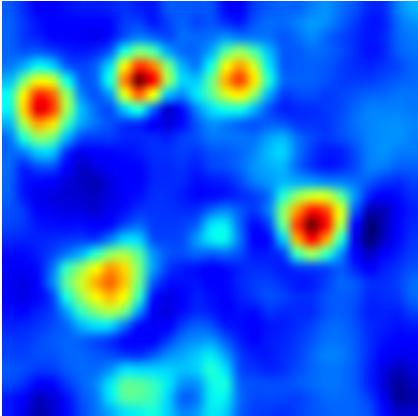


person

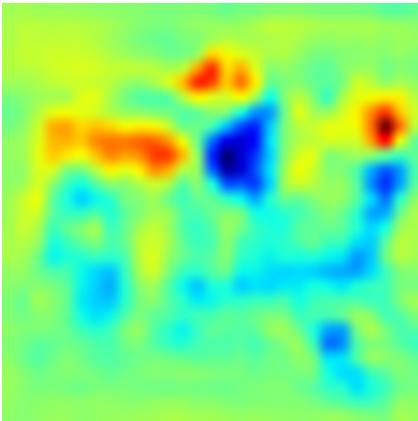


Dog

Last Epoch



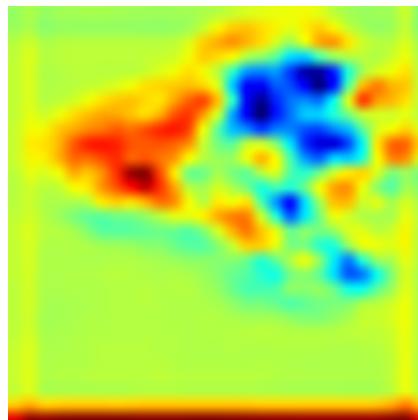
person



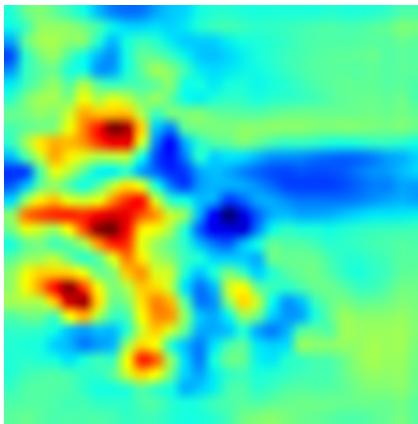
dog

Q1.7 Part 2

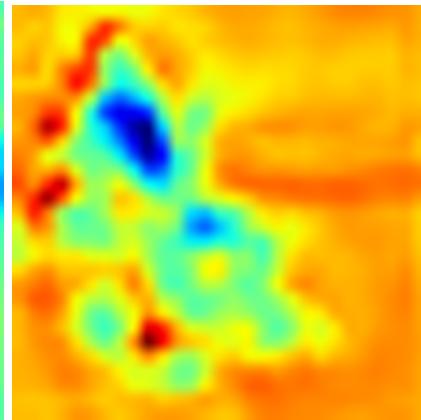
Plot 3 randomly chosen images (same images as Q1.6) and corresponding heatmaps from the validation set.



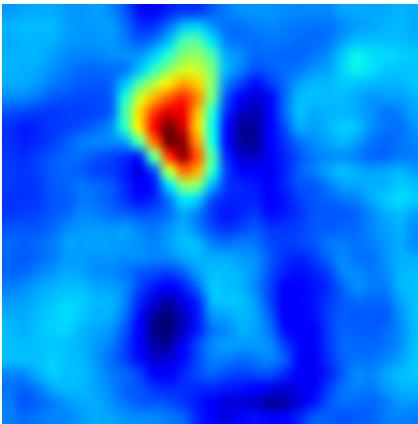
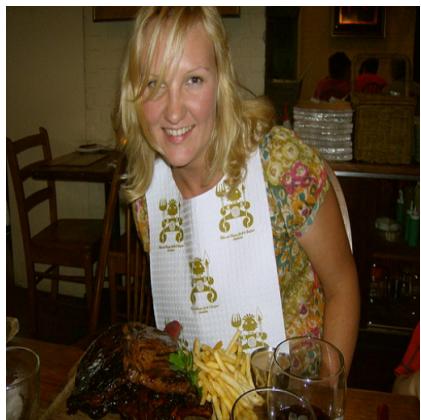
aeroplane



Motorbike



person



person

Q1.7 Part 3

Report the training loss, training and validation metric1 and metric2 achieved at the end of Training.

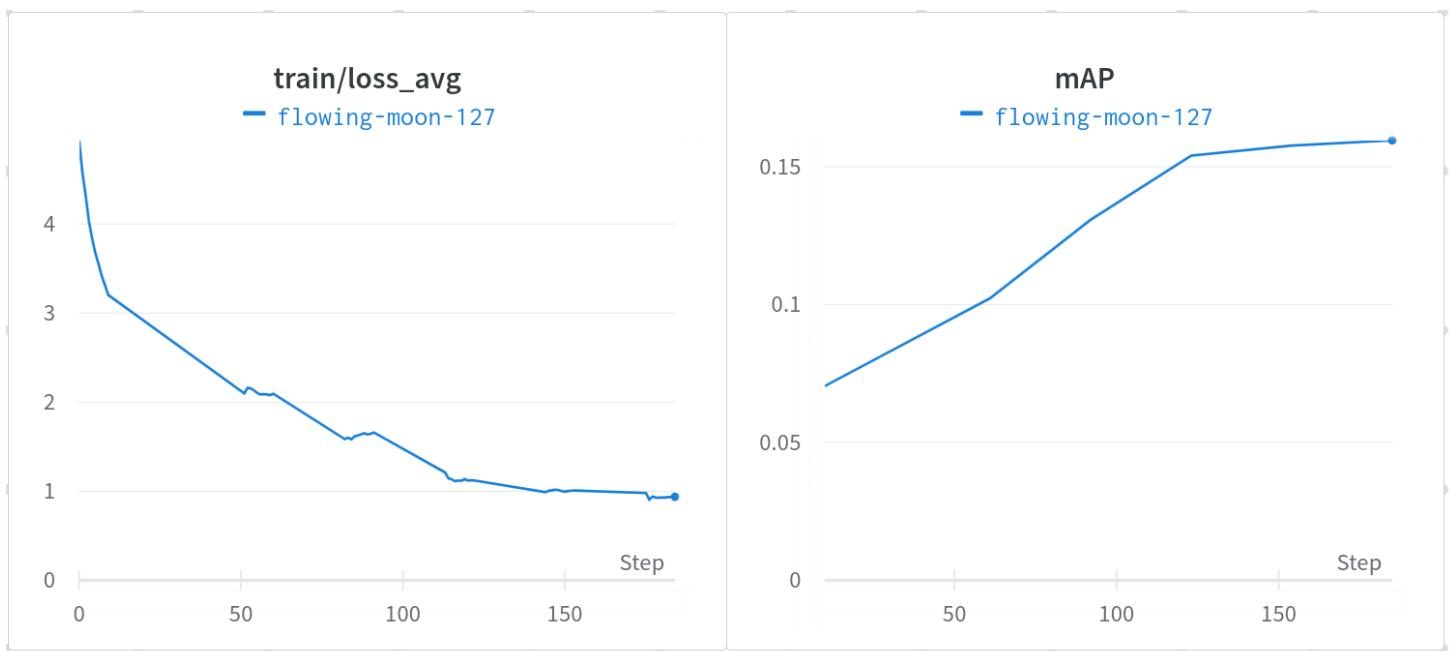
- Training loss: **0.03916**
- Training metric 1: **0.9644**
- Training metric 2: **0.87567**
- Validation metric 1: **0.70629**
- Validation metric 2: **0.5634**

3

Question 2

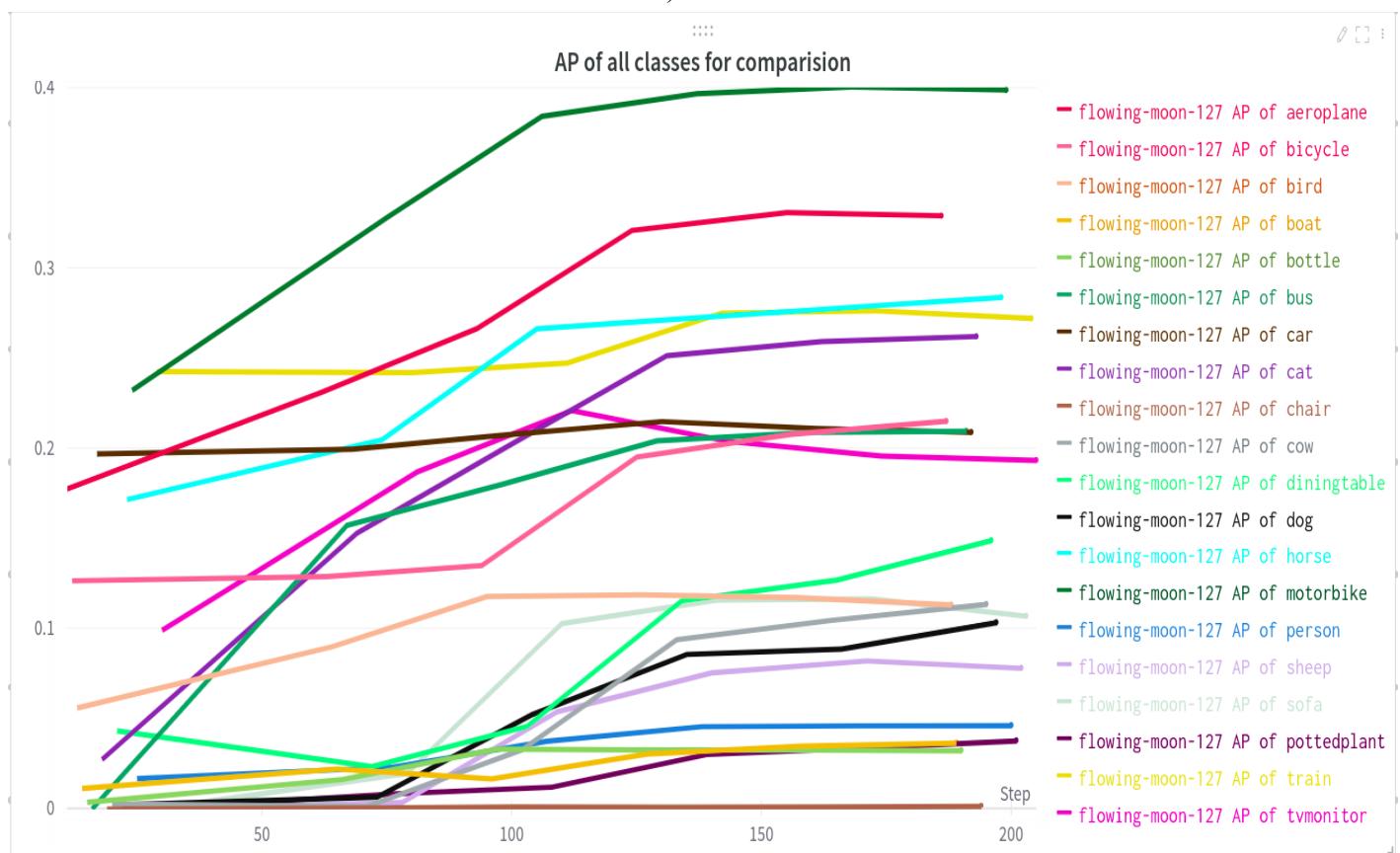
Q2.4 Part 1

Plot wandb downloaded image of training loss vs iterations. Use wandb to plot mAP on the test set every epoch. Please post this as one image with 2 columns.

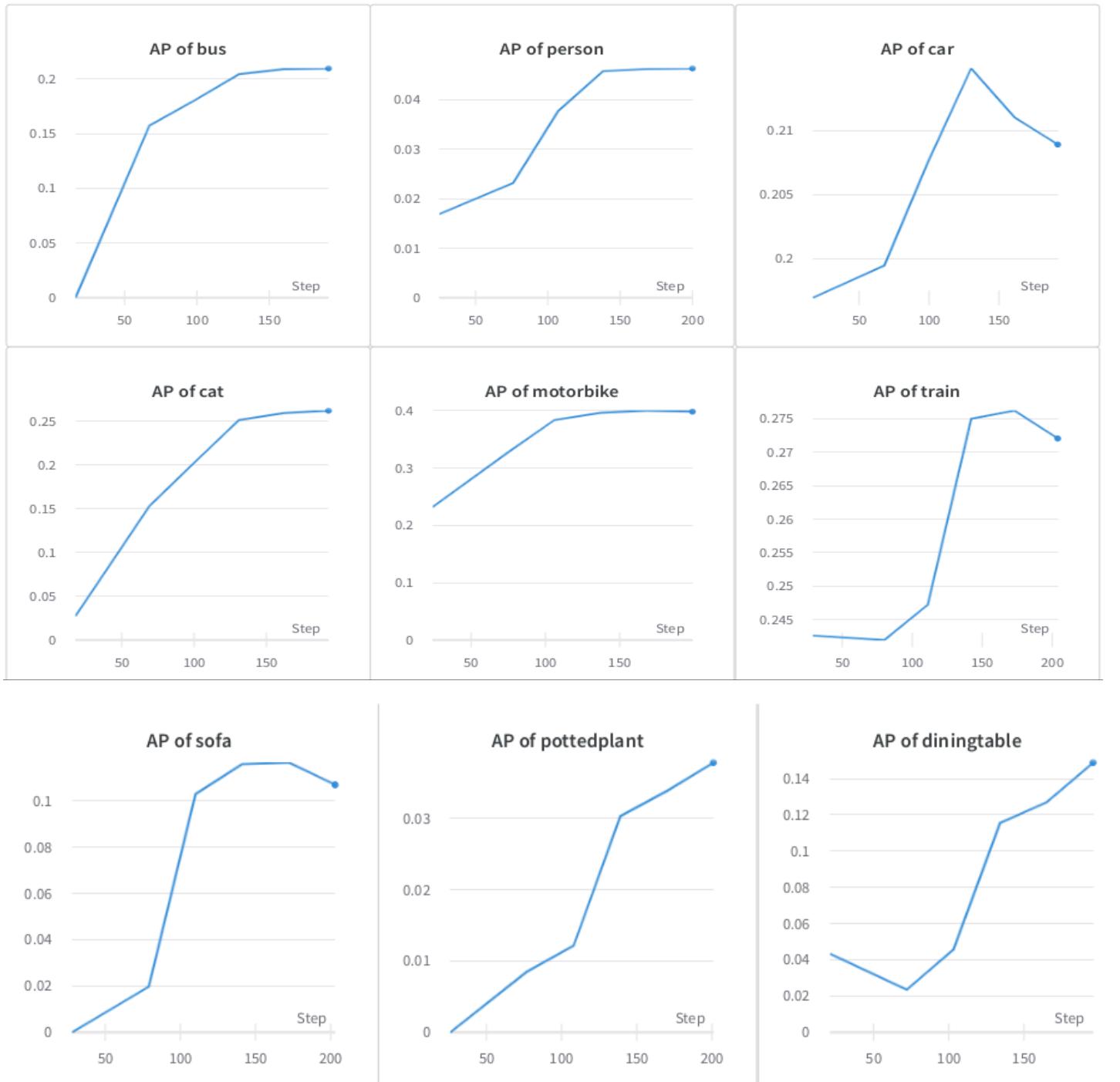


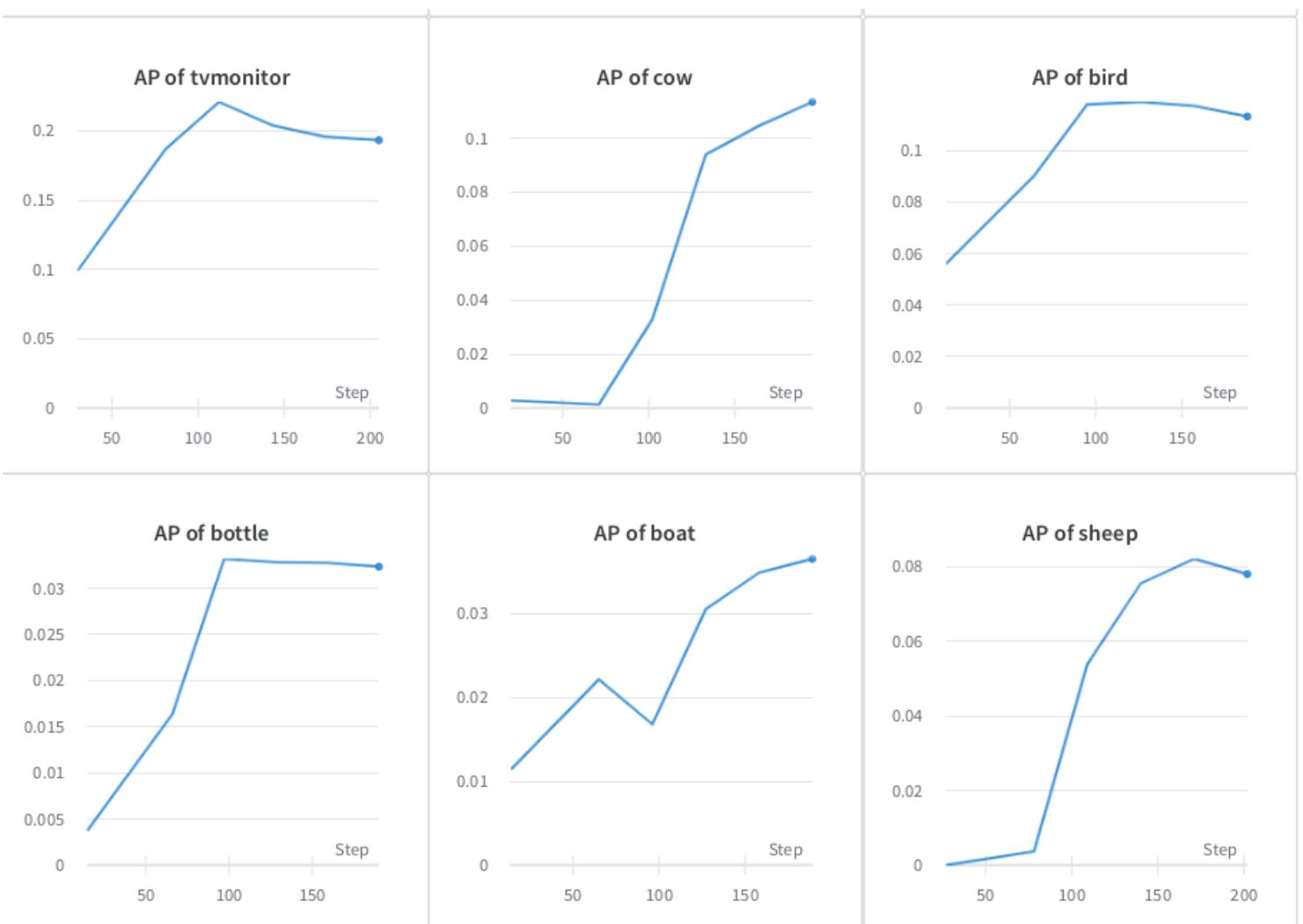
Q2.4 Part 2

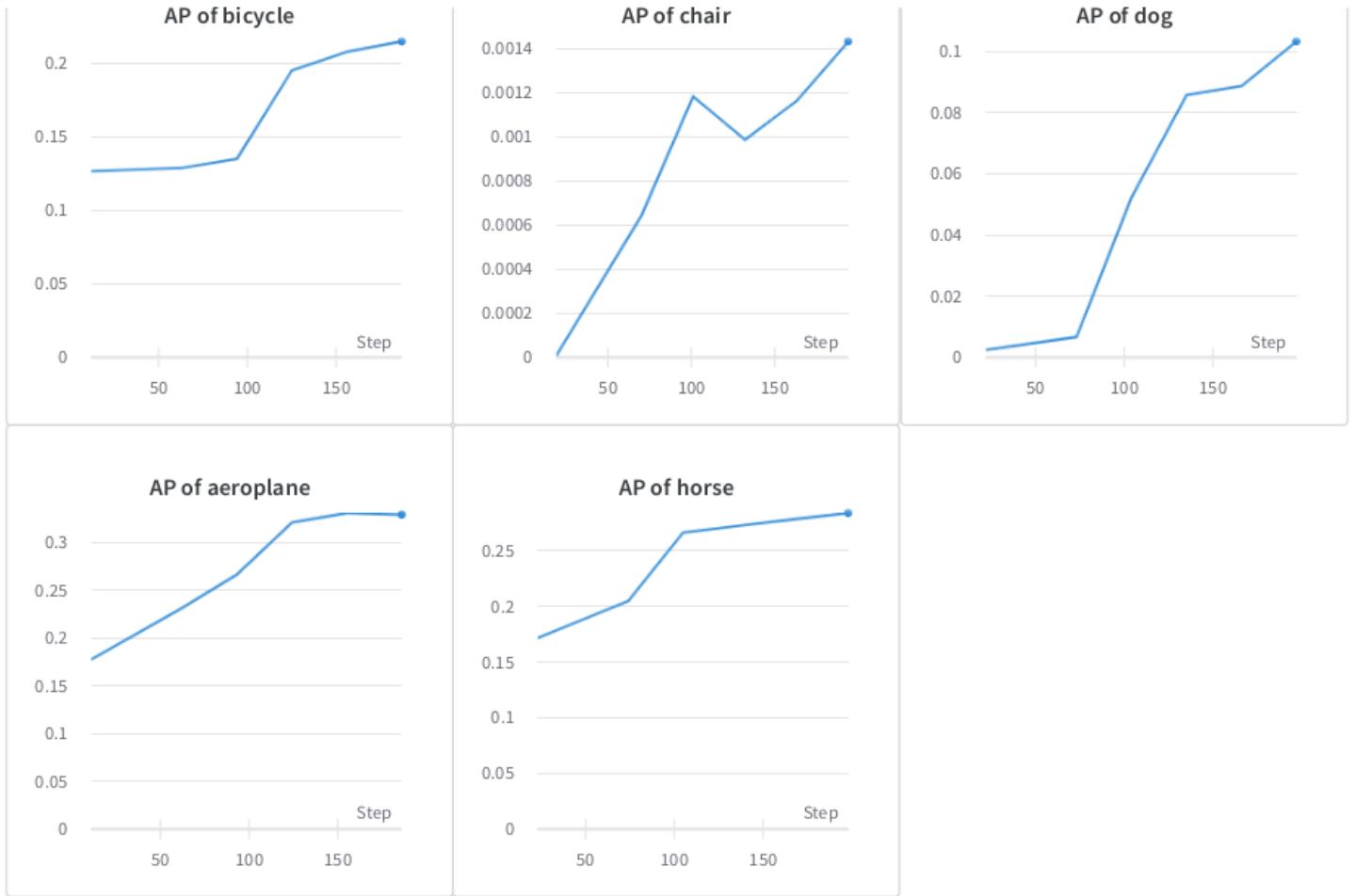
Plot downloaded image of class-wise test-mAPs vs iterations plot. Please make sure you plot all images (3 or more) for class-wise test APs in a single image so that it is easy to compare. You can use matplotlib or latex (or any other tool you prefer) to create a NxM image where N is the number of rows and M is the number of columns).



AP for individual classes

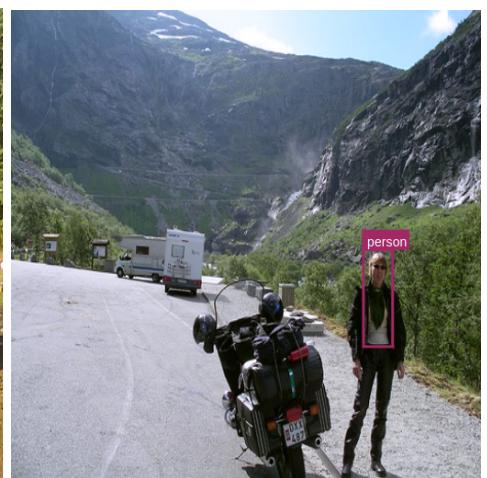


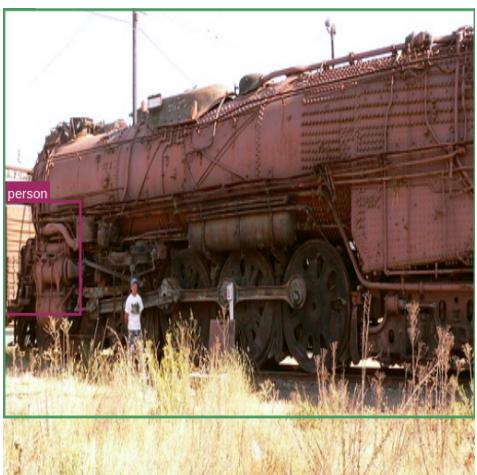
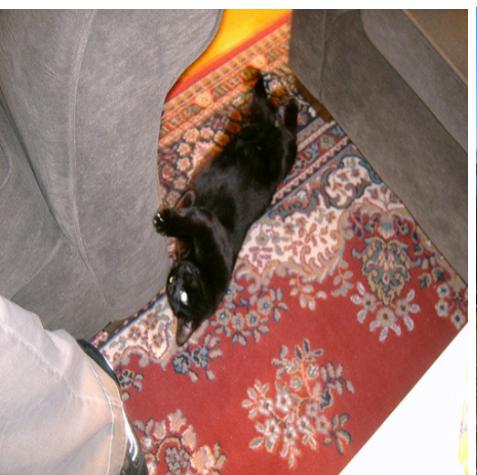
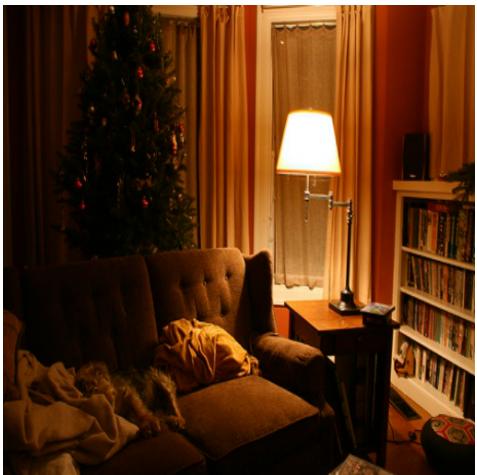
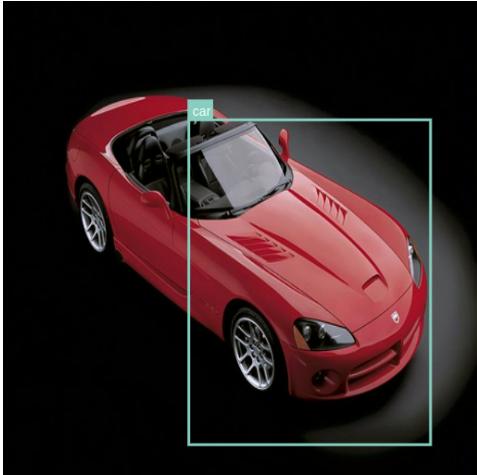




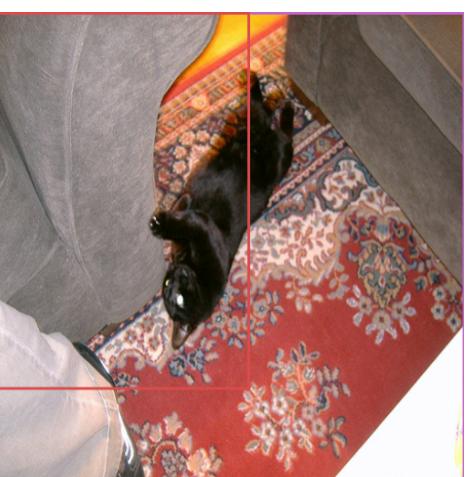
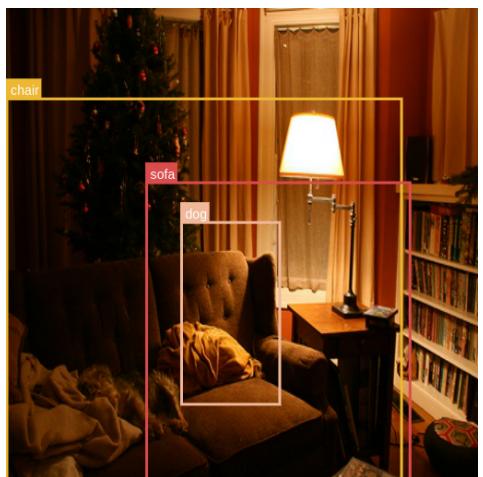
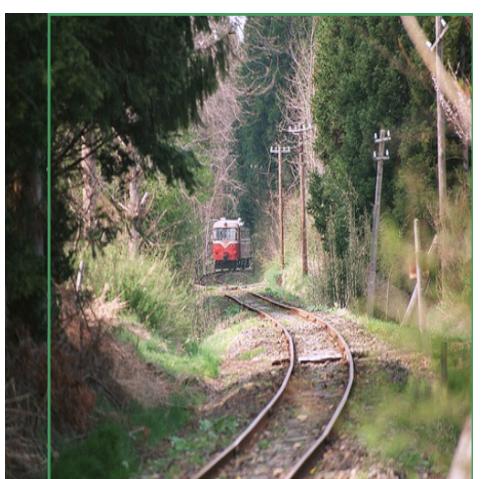
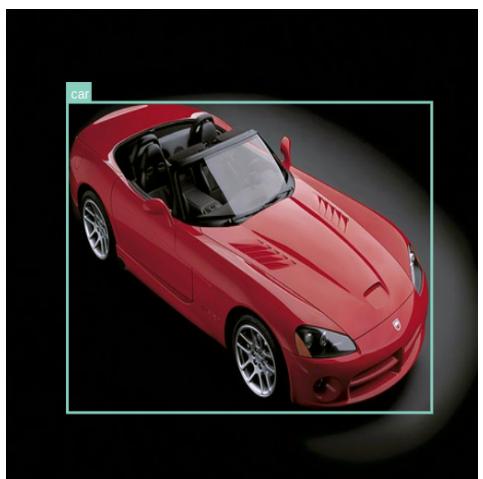
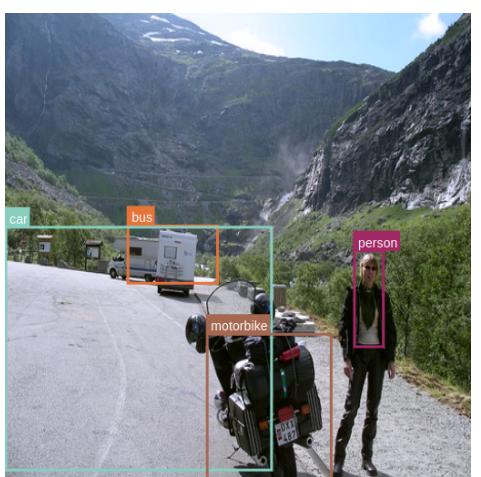
Q2.4 Part 3

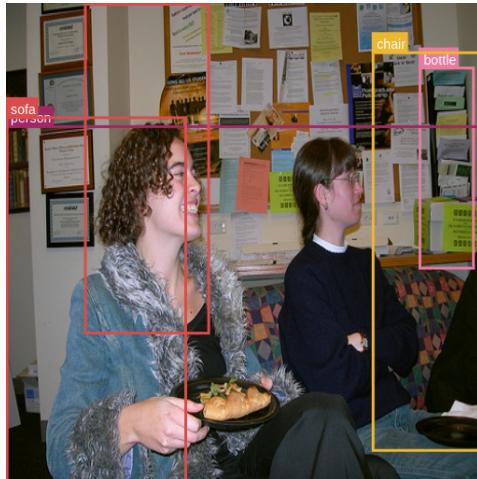
Plot bounding boxes on 9 random images at the end of the first epoch, and at the end of the last epoch. For this question please plot a 3x3 image instead of pasting 10 images one after the other.
After 1st epoch





After 5 epochs





Legend of bounding box classes as some labels are outside the image.

aeroplane	bicycle	bird	boat	bottle	bus
car	cat	chair	cow	diningtable	
horse	motorbike		person	pottedplant	
sheep	sofa	train	tvmonitor		

Q2.5 Part 4

Report the final class-wise AP on the test set and the mAP. Truncate your results to 4 decimal Places.

Class wise AP:

1. AP of aeroplane 0.3289
2. AP of bicycle 0.2150
3. AP of bird 0.1131
4. AP of boat 0.03656
5. AP of bottle 0.0323
6. AP of bus 0.2094
7. AP of car 0.2089
8. AP of cat 0.2620
9. AP of chair 0.0014
10. AP of cow 0.1134
11. AP of diningtable 0.1489
12. AP of dog 0.1033

13. AP of horse 0.2837
14. AP of motorbike 0.3987
15. AP of person 0.0463
16. AP of pottedplant 0.0378
17. AP of sheep 0.07804
18. AP of sofa 0.1069
19. AP of train 0.2720
20. AP of tvmonitor 0.1934

mAP 0.15952