

HOMEWORK 2: WEAKLY SUPERVISED OBJECT LOCALIZATION

16-824 Visual Learning
Recognition (Spring 2021)
Carnegie Mellon University

[Homework PDF Link](#)

BY: Feng Xiang*

DUE: Monday, March 29, 2021 11:59 PM

Notes

- N/A

*Compiled on Friday 2nd April, 2021 at 01:24

Task 0

Q0.1

Prompt What classes does the image at index 2020 contain (index 2020 is the 2021-th image due to 0-based numbering)?

Answer The 2020-th image shown below (see Figure 0.1) displays the train class which is class number 18.



Figure 0.1: 2020-th Image

References

- N/A

Q0.2

Prompt What is the easiest way to select the most informative regions? (Hint: look at the scores corresponding to each proposal in *voc_2007_trainval.mat*.

Answer The easiest way to select the most informative regions is to order the region proposal based on the highest *boxScores*. Once the proposals are ordered, one can return the top-N proposals for a given image.

References

- N/A

Q0.3

Prompt Use Wandb to visualize the ground-truth bounding box and the class for the image at index 2020.

Answer The figure shown below (see Figure 0.2) displays the ground truth label for the 2020-th image.

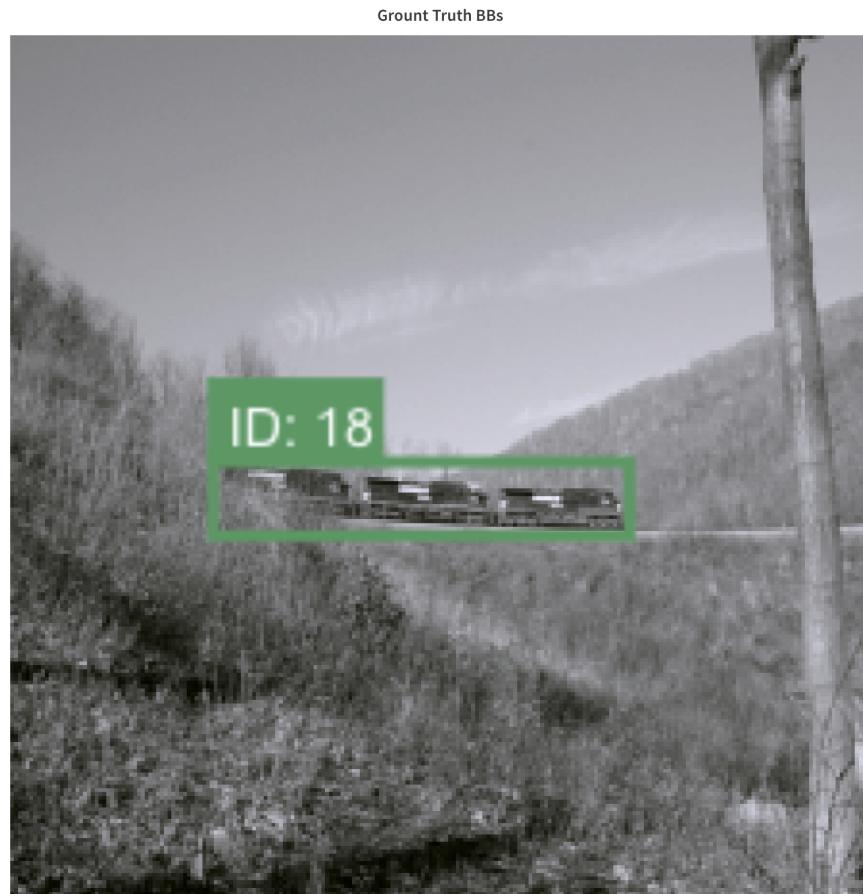


Figure 0.2: Ground Truth: 2020-th Image

References

- N/A

Q0.4

Prompt Use Wandb to visualize the top ten bounding box proposals for the image at index 2020.

Answer The figure shown below (see Figure 0.3) displays the top ten region proposals for the 2020-th image.



Figure 0.3: Region Proposals: 2020-th Image

References

- N/A

Task 1

Q1.1

Prompt For each TODO in *AlexNet.py* and *task_1.py* (except for the functions *metric1*, *metric2*, *LocalizerAlexNetRobust*), describe the functionality of that part.

Answer

AlexNet.py

- LocalizerAlexNet: define model

The model is defined by two overarching layers: the feature and classifier layers. The architecture of both overarching layers are based on the AlexNet architecture. The features layers is identical to the features layer of AlexNet, except for the *pool5* layer. The classifier layers contain three convolutional layers, where the last two have kernel sizes of one. Comparing this layer to the AlexNet classifier layer, the last two FC layers are replaced with 1-D convolutional layers, and the dropout layers were removed.

- LocalizerAlexNet: define forward pass

The forward pass through the model is simply passing the input image through the features layer then classifier layer.

- localizer_alexnet: initialize weights correctly based on whether it is pretrained or not

The function will initialize an instance of the *LocalizerAlexNet* class and, depending on the inputs to the function, will set the weights and biases of the features layers based on the ImageNet pretrained AlexNet version. In addition to setting the weights and biases in the features layers, if the input is set to *pretrained = True*, the classifier layers' weights will be Xavier Initialized and the biases will be set to zero. If the input is set to *pretrained = False*, then the model will be Xavier Initialized with the biases being set to zero.

task_1.py

- main: define loss function (criterion), optimizer, and LR scheduler

The loss function selected was binary cross-entropy. To note, the output from the model first passes through a Sigmoid function before being inputted to the loss function. The selected optimizer was stochastic gradient descent with momentum and weight decay. The LR scheduler selected was a stepLR function.

- main: create datasets and dataloaders using VOCDataset

Two separate dataloader instances were created: train and validation. The training dataloader called the 'trainval' dataset. Images are resized to 512x512 and the top 30 proposals are chosen. The validation dataloader called the 'test' dataset. Images are resized to 512x512 and the top 30 proposals are chosen.

- main: create loggers for wandb

The *wandb.log* function was used to both plot scalar values over time (i.e. loss, metric1, and amtric2) as well asl plotting image data.

- train: you can add input arguments if you wish

The default inputs were chosen for this function.

- train: get inputs from the data dict

Inside the training loop of the *train* function, the image batch is extracted from the data iteration. The batch size is predetermined as a hyperparameter before starting the training loop.

- train: get output from the model

The extracted image batch at that iteration of the for loop is passed through the *LocalizerAlexNet* model. The output from the model is a 3D spatial feature map stack.

- train: perform any necessary functions on the output such as clamping

The output from the model is sent into a spatial max pooling function. In the function, the maximum value at each spatial layer is extracting and created into an return matrix that is batch size by class size large.

- train: compute loss using the defined criterion

Outputs from the max pooling function are passed into the criterion as the predictions. The ground truth input is the target vector for that dataset multiplied by the weights vector.

- train: compute gradient descent and do SGD step

To start this code block, the optimizer is reset back to a zero gradient. Backpropagation is performed through the network to compute gradients based on the results of the loss function. An additional step was performed afterwards, where gradient values are clipped to have a max value of 1.0. Finally, the optimizer takes a 'step', thus updating the weights and biases from the values determined during backpropagation.

- train: visualize at appropriate intervals

Wandb logs are set in this code block. This includes training losses and metrics that are being sent to the specified *Wandb* repository for this particular run.

- validate: get inputs from the data dict

Similar to the same *TODO* in the *train* function, the image batch is extracted from the data iteration. The batch size is predetermined as a hyperparameter before starting the training loop.

- validate: get output from model

Similar to the same *TODO* in the *train* function, the extracted image at that iteration of the for loop is passed through the model. The output from the model is a 3D spatial feature map stack.

- validate: perform any necessary functions on the input

Similar to the same *TODO* in the *train* function, the output from the model is sent through a max pooling function. In the function, the maximum value at each spatial layer is extracting and created into a return matrix that is batch size by class size large.

- validate: compute loss using the defined criterion

Similar to the same *TODO* in the *train* function, outputs from the max pooling function are passed into the criterion as the predictions. The ground truth input is the target vector for that dataset multiplied by the weights vector.

- validate: visualize at appropriate intervals

Similar to the same *TODO* in the *train* function, training losses and metrics are sent and logged into the specified *Wandb* repository for this particular run.

References

- Is object localization for free? - Weakly-supervised learning with convolutional neural networks

Q1.2

Prompt What is the output resolution of the model?

Answer The output resolution of the model, given a 512x512 input image in a batch of 32 is (32, 20, 29, 29). There are 32 images in one batch, 20 class labels, and a 29x29 spatial output resolution.

References

- N/A

Q1.3

Prompt Initialize and train the model with the given parameters. Plot the training loss curve. Plot images and heatmaps for 2 batches (1 image in each batch).

Answer The figure shown below (starting at Figure 0.4) displays the training loss plot through the 2 epoch training period.

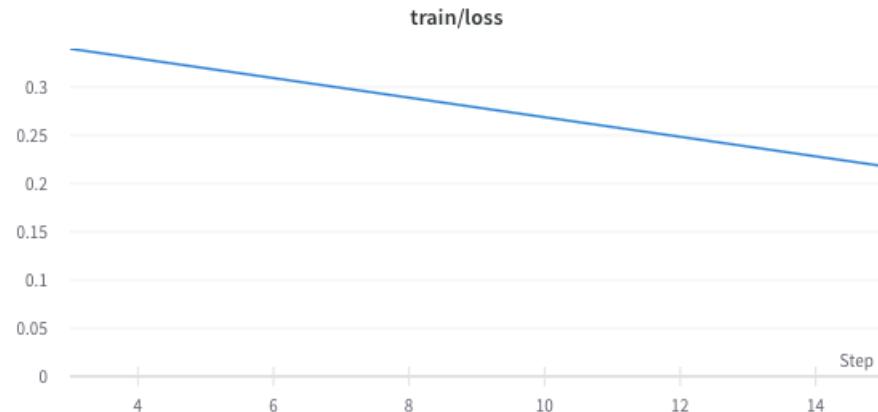


Figure 0.4: Training loss over 2 epochs

The figure shown below (see Figure 0.5) displays the reference image as well as two heat maps for the first sampled image during the first epoch.

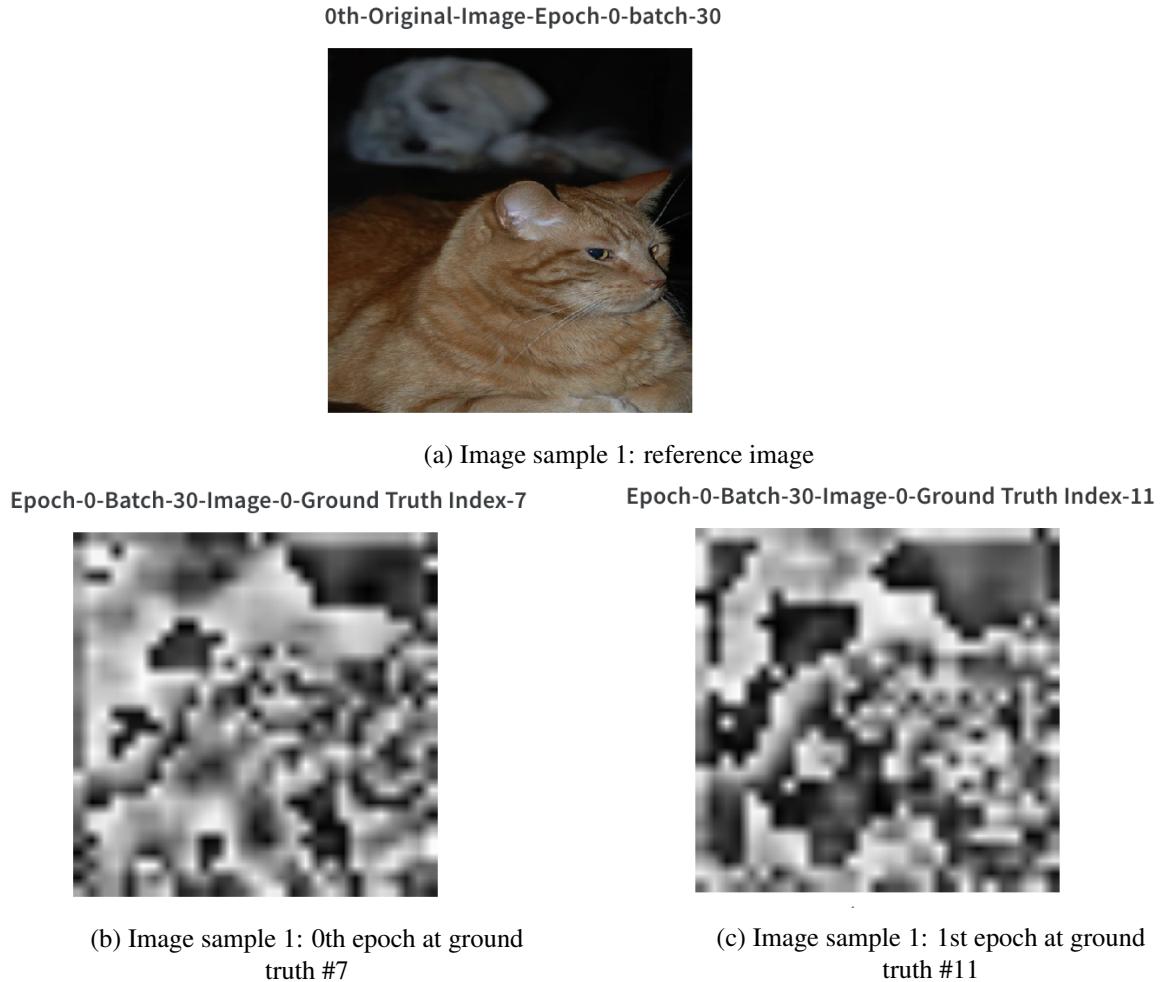


Figure 0.5: Image sample 1: Q1.3 reference image and heatmaps at 0th epoch

The figure shown below (see Figure 0.6) displays the two heatmaps for the first sampled image at the second epoch.

Epoch-1-Batch-30-Image-0-Ground Truth Index-7



(a) Image sample 1: 2nd epoch at ground truth #7

Epoch-1-Batch-30-Image-0-Ground Truth Index-11



(b) Image sample 1: 2nd epoch at ground truth #11

Figure 0.6: Image sample 1: Q1.3 heatmaps at last epoch

The figure shown below (see Figure 0.7) displays the reference image as well as the heatmap for the second sampled image at the first epoch.

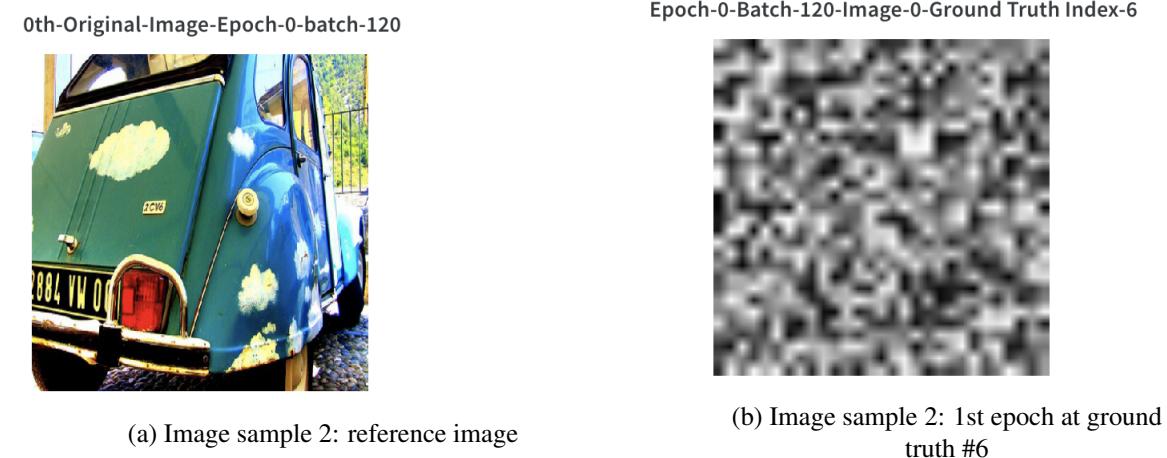


Figure 0.7: Image sample 2: Q1.3 reference image and heatmap at last epoch

The figure shown below (see Figure 0.8) displays the heat map for the second sampled image at the second epoch.

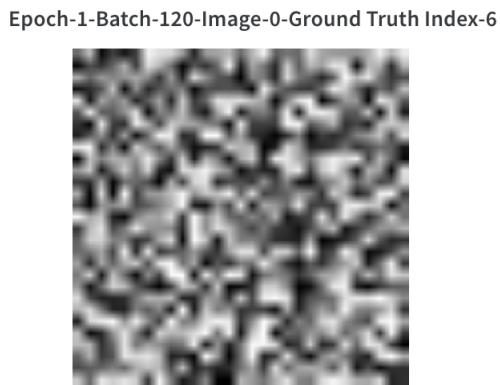


Figure 0.8: Image sample 2: 2nd epoch at ground truth #6

References

- N/A

Q1.4

Prompt In the first few iterations, you should observe a steep drop in the loss value. Why does this happen? (Hint: Think about the labels associated with each image).

Answer At the beginning of the training loop, the model's classifier layer was randomly initialized based on Xavier initialization. On the first pass through the model, the model has not built a basis for inferences yet. Therefore, the model will either output all 1's or all 0's. Once the loss function is called, the function will output low AP values. As the model continues to train, however, the loss function will output significantly lower and lower training loss values.

References

- N/A

Q1.5

Prompt The first metric is a standard metric for multi-label classification. Do you remember what this is? Write the code for this metric in the TODO block for *metric1* and *metric2*.

Answer The standard metric used in *metric1* is Mean-Average-Precision (mAP).

The functionality of *metric1* consists of computing the Average-Precision (AP) for each of the 20 classes in the output matrix. The inputs to *metric1* are the output from the model and the ground truth classifier labels. The output from the model serves as the prediction vector. The prediction vector is fed into a Sigmoid function. Afterwards, the AP was computed for that particular class based on the function **nn.Sigmoid**. The output from the sigmoid function is the class probabilities predictions for each image in the batch. The ground truth and predictions are then sent through the **sklearn.metrics.average_precision_score** function where the AP value will be return. The AP value is then appended to a python list, where, at the end of the function, the mean value is calculated from that list and returned as the mAP.

The functionality of *metric2* consists of computing the recall score between the ground truth labels and the class predictions. The beginning segment of the function is similar to the *metric1* function, in that the output from the model is passed through the **nn.Sigmoid** function to return the class probabilities. The difference is that afterwards, the class probabilities are converting to a one-hot vectors where the threshold is at 0.5. The one-hot class vector and ground truth labels are sent to the **sklearn.metrics.recall_score** function where the recall value at that particular class is returned. The recall score is appended to a python list, where, at the end of the function, the mean value is calculated from that list and returned as the recall score value.

References

- N/A

Q1.6

Prompt The first metric is a standard metric for multi-label classification. Do you remember what this is? Write the code for this metric in the TODO block for *metric1* and *metric2*.

Answer The figure shown below (see Figure 0.9) shows the training loss of the model after 30 epochs.

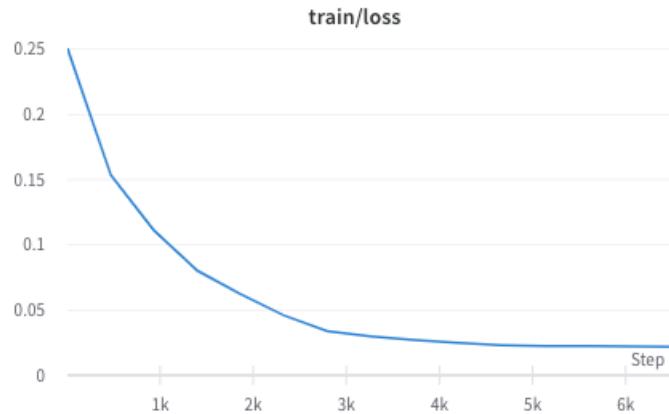


Figure 0.9: Training loss over 30 epochs

The figures shown below (see Figure 0.10) show the training mAP and recall scores over 30 epochs.

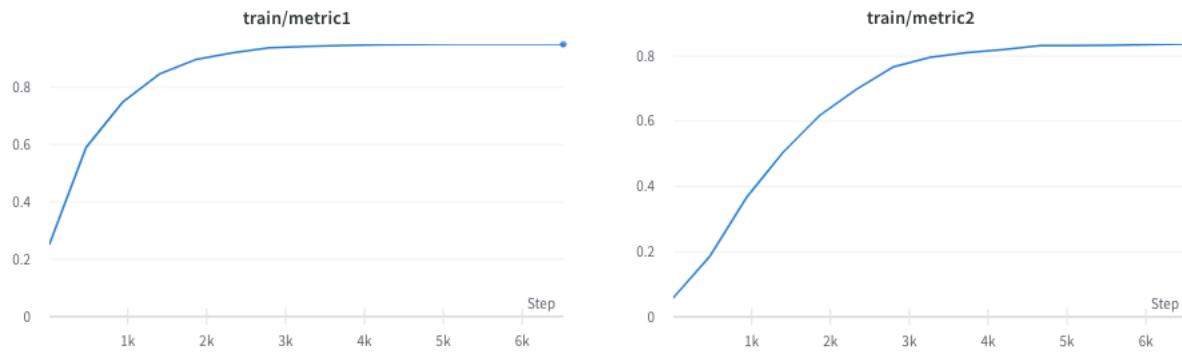
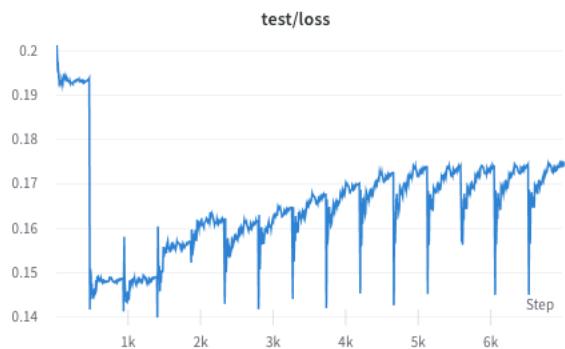
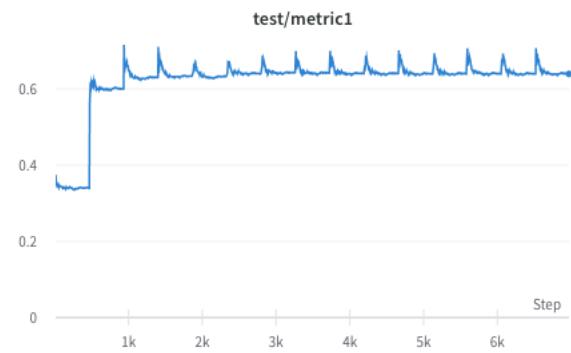


Figure 0.10: Training metric1 and metric 2 over 30 epochs

The figures shown below (see Figure 0.11) show the validation mAP and recall scores over 30 epochs.



(a) Testing mAP over 30 epochs



(b) Testing recall score over 30 epochs

Figure 0.11: Validation metric1 and metric 2 over 30 epochs

The figure shown below (see Figure 0.12) display the image and heatmaps of the first sampled image at the first epoch.

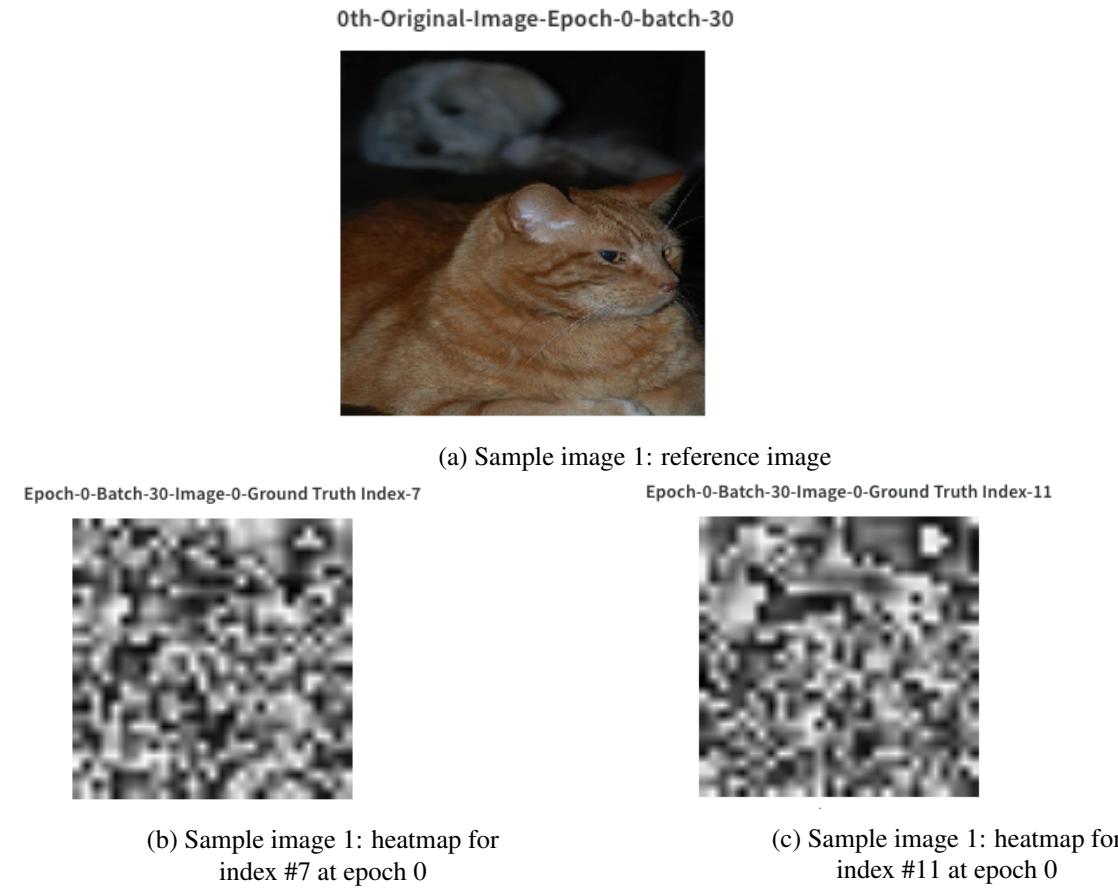


Figure 0.12: Sample image 1: reference image and heatmaps at epoch 0

The figure shown below (see Figure 0.13) display the image and heatmap for the second sampled image at the first epoch.

0th-Original-Image-Epoch-0-batch-120



(a) Sample image 2: reference image

Epoch-0-Batch-120-Image-0-Ground Truth Index-6



(b) Sample image 2: heatmap for
index #6 at epoch 0

Figure 0.13: Sample image 2: reference image and heatmaps for epoch 0

The figure shown below (see Figure 0.14) display heatmap comparisons of the first sampled image at the last epoch.

Epoch-29-Batch-30-Image-0-Ground Truth Index-7



(a) Sample image 1: heatmap for
index #7 at epoch 29

Epoch-29-Batch-30-Image-0-Ground Truth Index-11



(b) Sample image 1: heatmap for
index #11 at epoch 29

Figure 0.14: Sample image 1: heatmaps at epoch 29

The figure shown below (see Figure 0.15) display the image and heatmap for the second sampled image at the last epoch.

Epoch-29-Batch-120-Image-0-Ground Truth Index-6



Figure 0.15: Sample image 2: heatmap for index #6 at epoch 29

The figure shown below (see Figure 0.16) display a randomized image and the associated heatmap for the specified index.

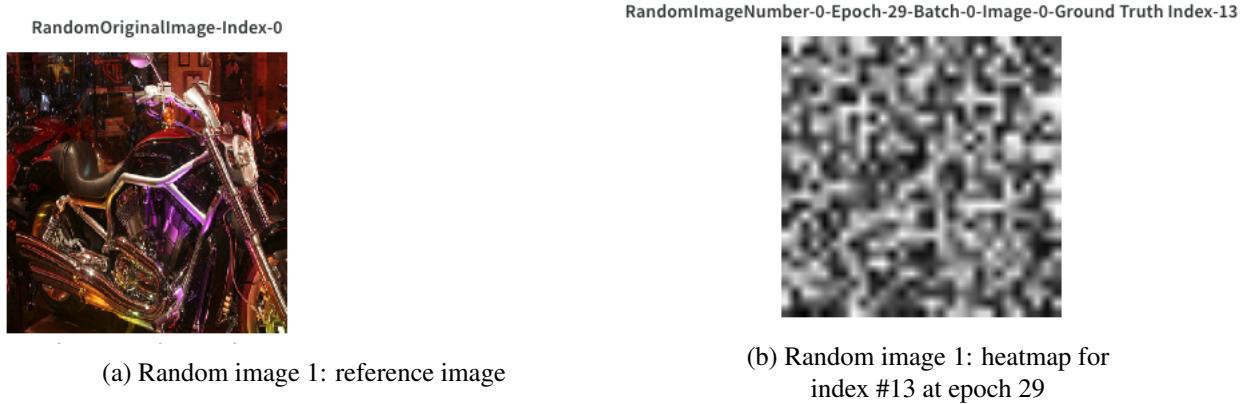


Figure 0.16: Random image 1: reference image and heatmaps at epoch 29

The figure shown below (see Figure 0.17) display a randomized image and the associated heatmap for the specified index.

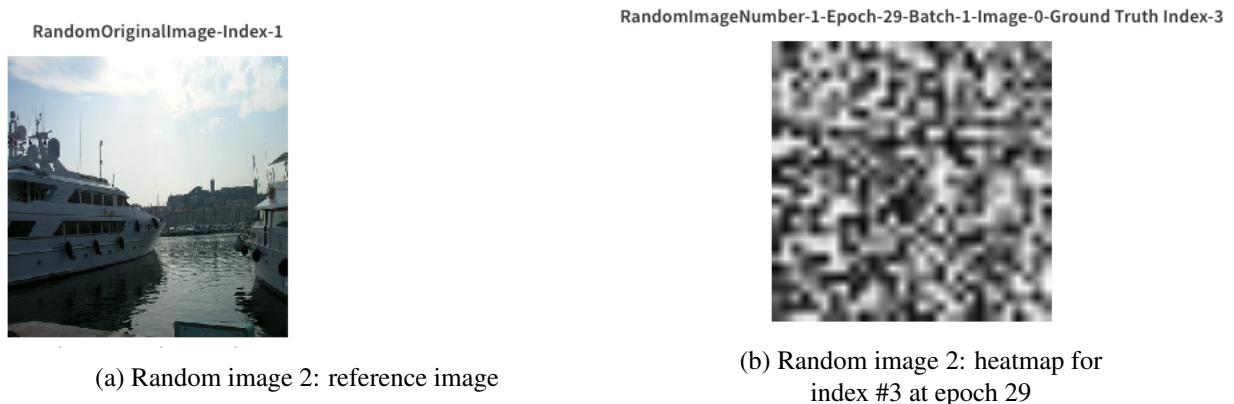


Figure 0.17: Random image 2: reference image and heatmaps at epoch 29

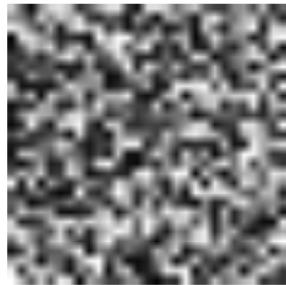
The figure shown below (see Figure 0.18) display a randomized image and the associated heatmaps for the specified index.

RandomOriginalImage-Index-2



(a) Random image 3: reference image

RandomImageNumber-2-Epoch-29-Batch-2-Image-0-Ground Truth Index-7 RandomImageNumber-2-Epoch-29-Batch-2-Image-0-Ground Truth Index-11



(b) Random image 3: heatmap for index #7 at epoch 29



(c) Random image 3: heatmap for index #11 at epoch 29

Figure 0.18: Random image 3: reference image and heatmaps at epoch 29

The final reported statistics of the best training session are the following:

- learning rate: 0.1
 - batch size: 32
 - epochs: 30
 - train/loss: 0.0220
 - train/metric1: 0.951
 - train/metric2: 0.840
 - validation/loss: 0.1745
 - validation/metric1: 0.639
 - validation/metric2: 0.352

For reference, the reported statistics for a learning rate of 0.01 is shown below for comparison:

- learning rate: 0.01

- train/loss: 0.17064
- train/metric1: 0.46741
- train/metric2: 0.09792
- validation/loss: 0.17776
- validation/metric1: 0.41057
- validation/metric2: 0.09091

References

- N/A

Q1.7

Prompt In the heatmap visualization you observe that there are usually peaks on salient features of the objects but not on the entire objects. How can you fix this in the architecture of the model? Implement this new model in *LocalizerAlexNetRobust* and also implement the corresponding *localizer_alexnet_robust()*.

Answer The figure shown below (see Figure 0.19) displays the first sampled image.

0th-Original-Image-Epoch-0-batch-30



Figure 0.19: Sample image 1: reference image

The figure shown below (see Figure 0.20) displays the first heat map comparison between the two AlexNet models for the first sampled image at the first epoch.

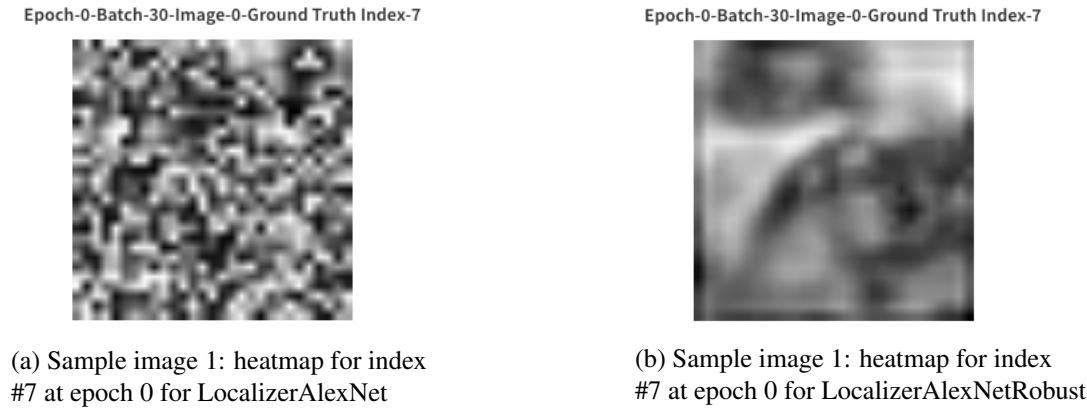


Figure 0.20: Sample image 1: heatmap comparison for index 7 at epoch 0

The figure shown below (see Figure 0.21) displays the second heat map comparison between the two AlexNet models for the first sampled image at the first epoch.

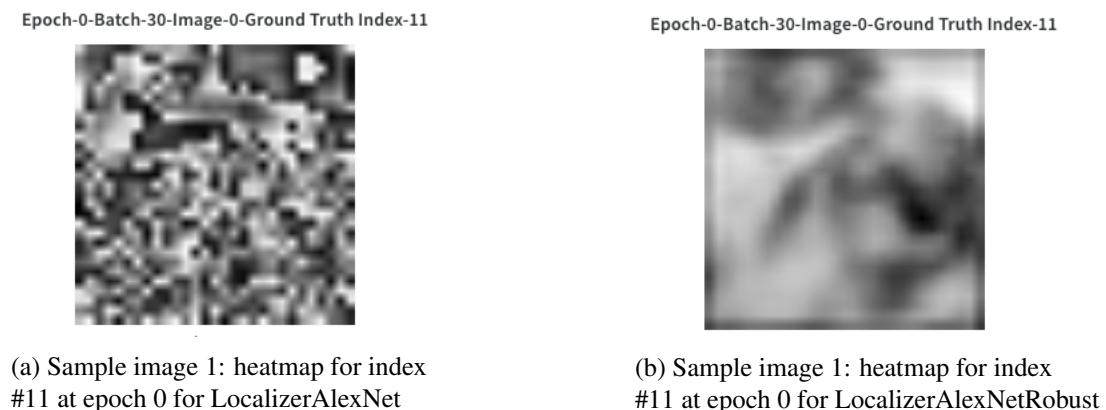


Figure 0.21: Sample image 1: heatmap comparison for index 11 at epoch 0

The figure shown below (see Figure 0.22a) displays the second sampled image.

0th-Original-Image-Epoch-0-batch-120



(a) Sample image 2: reference image

The figure shown below (see Figure 0.23) displays the heatmap comparison for the second sampled image at the first epoch.

Epoch-0-Batch-120-Image-0-Ground Truth Index-6



(a) Sample image 2: heatmap for index #6 at epoch 0 for LocalizerAlexNet

Epoch-0-Batch-120-Image-0-Ground Truth Index-6



(b) Sample image 2: heatmap for index #6 at epoch 0 for LocalizerAlexNetRobust

Figure 0.23: Sample image 2: heatmap comparison for index 6 at epoch 0

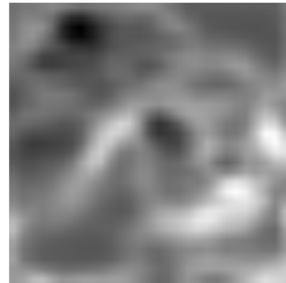
The figure shown below (see Figure 0.24) displays the heatmap comparison for the first sampled image at the last epoch

Epoch-29-Batch-30-Image-0-Ground Truth Index-7



(a) Sample image 1: heatmap for index #7 at epoch 29 for LocalizerAlexNet

Epoch-44-Batch-30-Image-0-Ground Truth Index-7



(b) Sample image 1: heatmap for index #7 at epoch 44 for LocalizerAlexNetRobust

Figure 0.24: Sample image 1: heatmap comparison for index #7 at last epoch

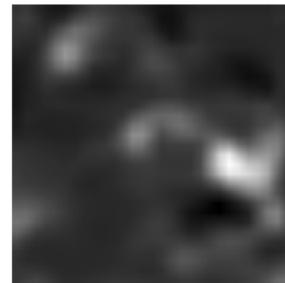
The figure shown below (see Figure 0.25) displays the second heatmap comparison for the first sampled image at the last epoch.

Epoch-29-Batch-30-Image-0-Ground Truth Index-11



(a) Sample image 1: heatmap for index #11 at epoch 29 for LocalizerAlexNet

Epoch-44-Batch-30-Image-0-Ground Truth Index-11

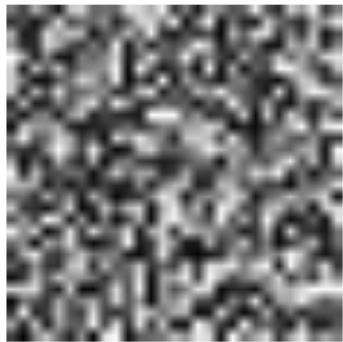


(b) Sample image 1: heatmap for index #11 at epoch 44 for LocalizerAlexNetRobust

Figure 0.25: Sample image 1: heatmap comparison for index #11 at last epoch

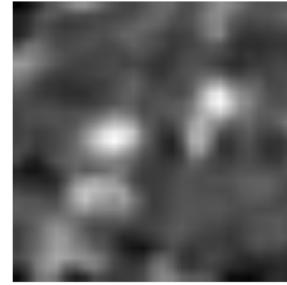
The figure shown below (starting at Figure 0.26) displays the heatmap comparision for the second sampled image at the last epoch.

Epoch-29-Batch-120-Image-0-Ground Truth Index-6



(a) Sample image 2: heatmap for index #6 at epoch 29 for LocalizerAlexNet

Epoch-44-Batch-120-Image-0-Ground Truth Index-6



(b) Sample image 2: heatmap for index #6 at epoch 44 for LocalizerAlexNetRobust

Figure 0.26: Sample image 2: heatmap comparison for index #6 at last epoch

The figure shown below (see Figure 0.17) displays the first randomly sampled image and its corresponding heatmap at the ground truth index.

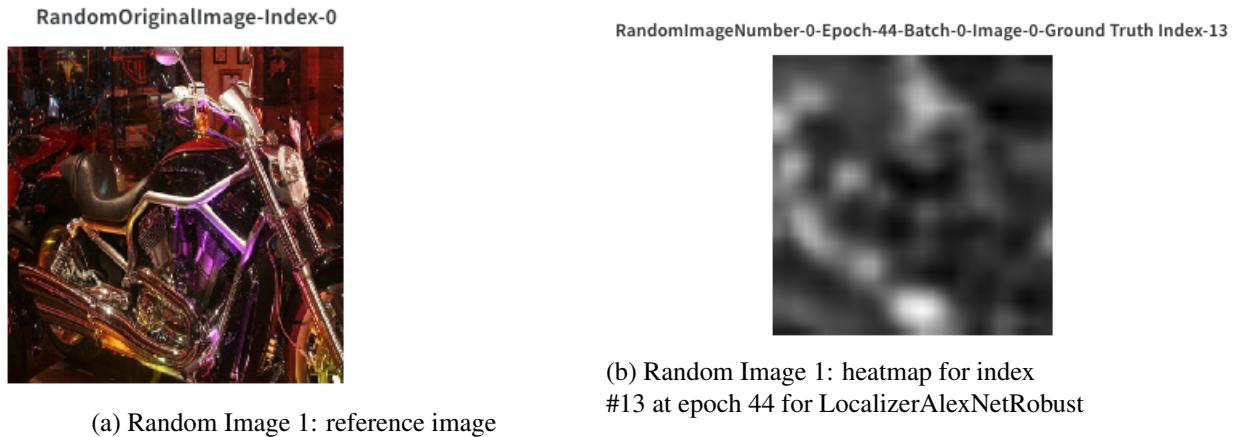


Figure 0.27: Random Image 1: reference image and heatmap for index #13 at last epoch

The figure shown below (see Figure 0.28) displays the second randomly sampled image and its corresponding heatmap at the ground truth index.

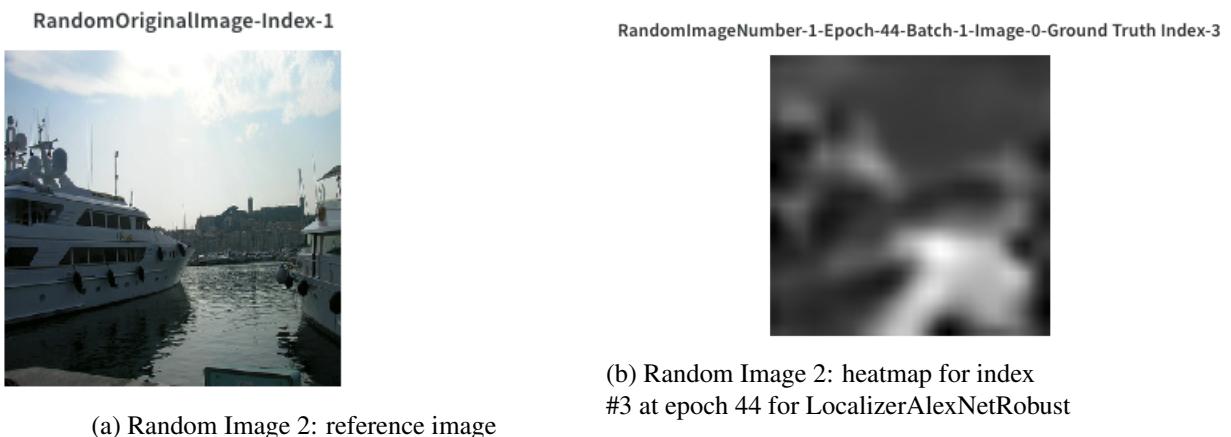


Figure 0.28: Random Image 2: reference image and heatmap for index #3 at last epoch

The figure shown below (see Figure 0.29) displays the third randomly sampled image and its first corresponding ground truth heatmap.



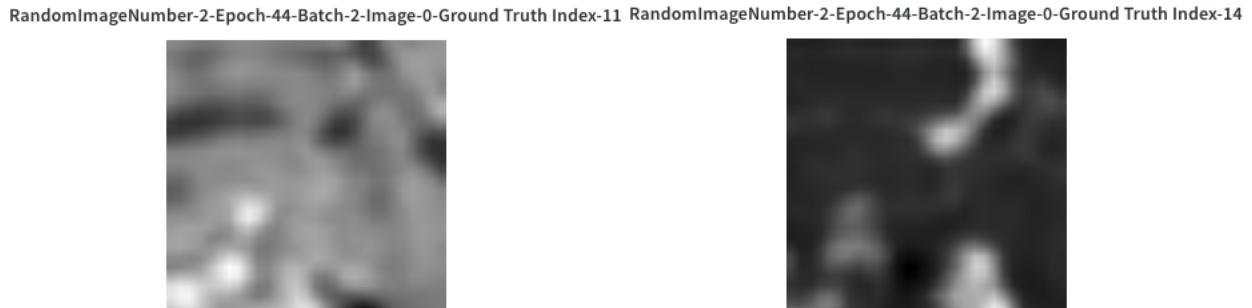
(a) Random Image 3: reference image



(b) Random Image 3: heatmap for index #7 at epoch 44

Figure 0.29: Random Image 3: reference image and heatmap for index #7 at last epoch

The figure shown below (see Figure 0.30) displays the last two corresponding heatmaps for the third randomly selected image.



(a) Random Image 3: heatmap for index #11 at epoch 44 for LocalizerAlexNetRobust

(b) Random Image 3: heatmap for index 14 at epoch 44 for LocalizerAlexNetRobust

Figure 0.30: Random Image 3: heatmap comparisons for various indices at last epoch

The figure shown below (see Figure 0.31) displays the training loss of the model over 45 epochs.

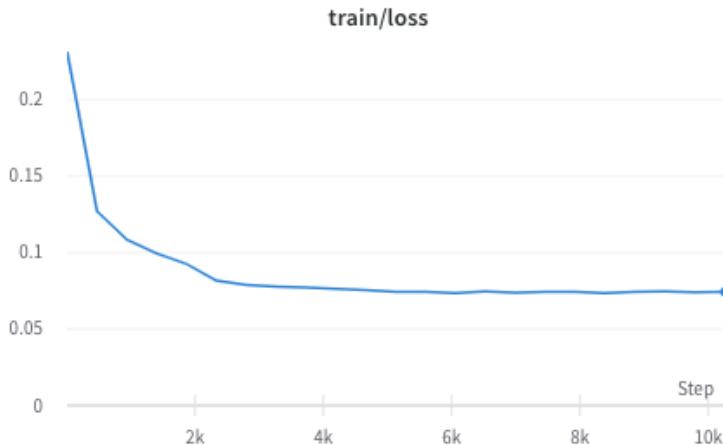


Figure 0.31: Training Loss over 45 epochs

The figure shown below (see at Figure 0.32) displays the *metric1* and *metric2* curves over 45 epochs.

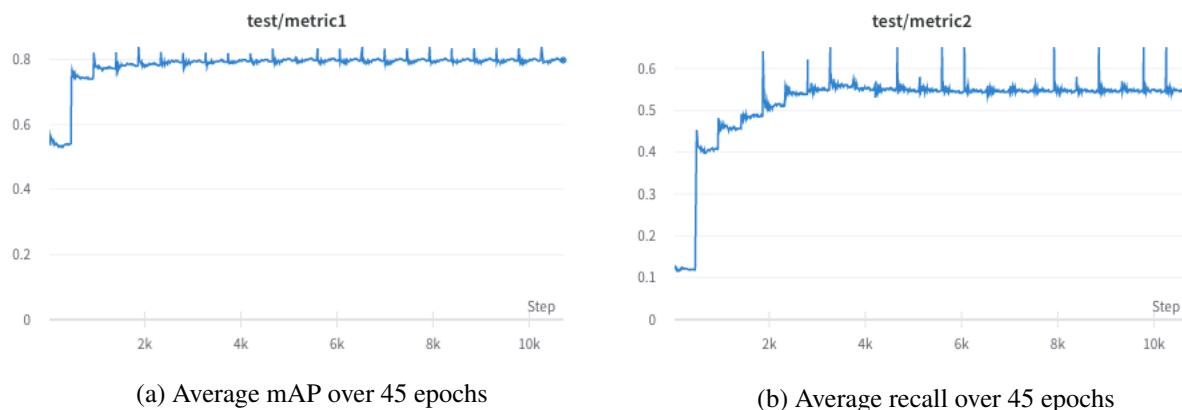


Figure 0.32: Average mAP and recall over 45 epochs

The final reported statistics of the best training session are the following:

- learning rate: 0.1
- batch size: 32
- epochs: 45
- train/loss: 0.07483
- train/metric1: 0.8663
- train/metric2: 0.59955
- validation/loss: 0.09892
- validation/metric1: 0.79491
- validation/metric2: 0.55166

For reference, the reported statistics for a learning rate of 0.01 is shown below for comparison:

- learning rate: 0.01
- train/loss: 0.12574
- train/metric1: 0.69143
- train/metric2: 0.32081
- validation/loss: 0.12299
- validation/metric1: 0.69372
- validation/metric2: 0.34235

References

- N/A

Task 2

Q2.4

Prompt Complete the TODOs on *task_2.py* and train for 5-6 epochs.

Answer The figure shown below (see Figure 0.33) displays the training loss of the model over 6 epochs.

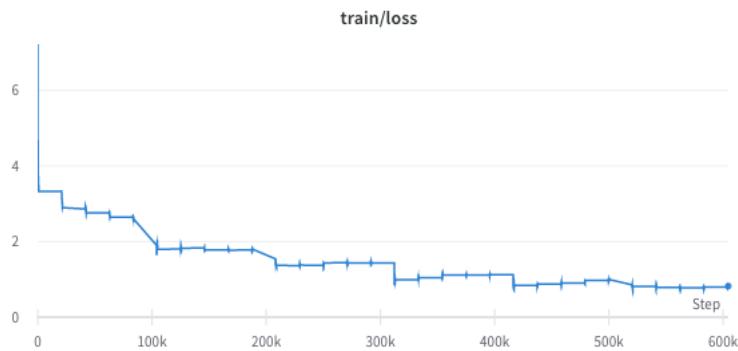


Figure 0.33: Training Loss over 6 epochs

The figure shown below (see Figure 0.34) displays the testing mAP of the model over 6 epochs.

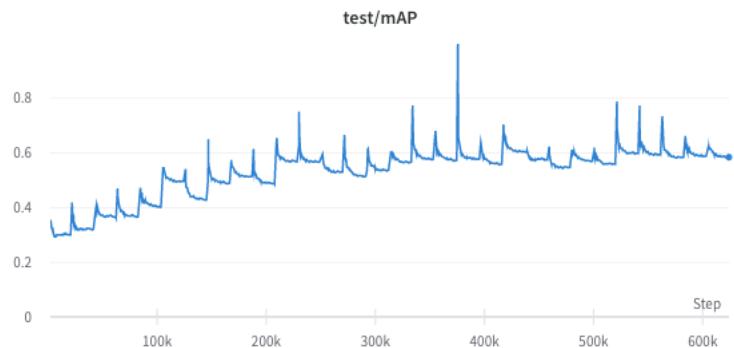


Figure 0.34: Testing mAP over 6 epochs

The figure shown below (see Figure 0.35) displays the testing AP values for index 7 of the model over 6 epochs.

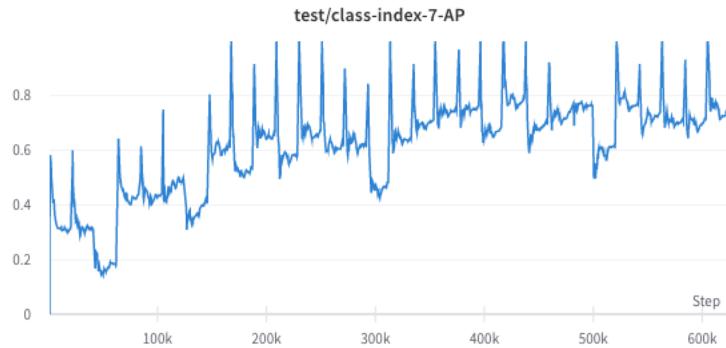


Figure 0.35: Testing AP for Index 7 over 6 epochs

The figure shown below (see Figure 0.36) displays the testing AP values for index 11 of the model over 6 epochs.

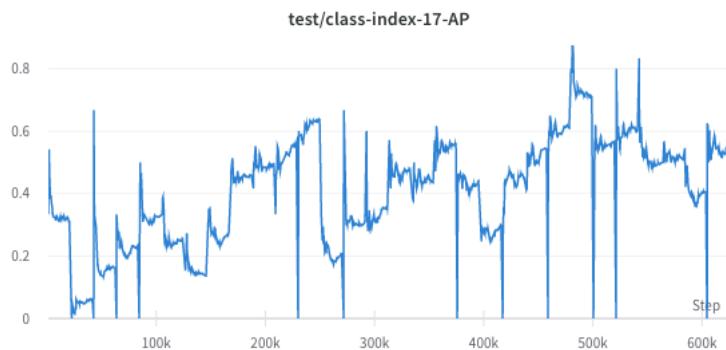


Figure 0.36: Testing AP for Index 11 over 6 epochs

The figure shown below (see Figure 0.37) displays the testing AP values for index 17 of the model over 6 epochs.



Figure 0.37: Testing AP for Index 17 over 6 epochs

The figure shown below (see Figure 0.38) displays a bounding box proposal for the first sampled image.

Epoch-0-RandomImage-12-ClassNumber-14



Figure 0.38: Sample Image 1: bounding box index 14

The figure shown below (see Figure 0.39) displays a bounding box proposal for the second sampled image.

Epoch-0-RandomImage-645-ClassNumber-6



Figure 0.39: Sample Image 2: bounding box index 17

The figure shown below (see Figure 0.40) displays a bounding box proposal for the third sampled image.

Epoch-0-RandomImage-104-ClassNumber-14

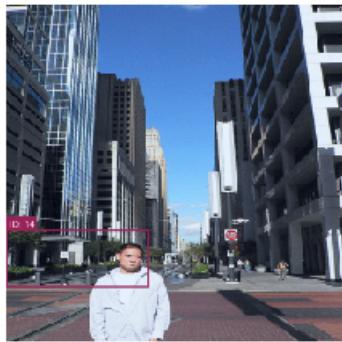


Figure 0.40: Sample Image 3: bounding box index 14

The figure shown below (see Figure 0.41) displays a bounding box proposal for the fourth sampled image.

Epoch-0-RandomImage-211-ClassNumber-6



Figure 0.41: Sample Image 4: bounding box index 6

The figure shown below (see Figure 0.42) displays a bounding box proposal for the fifth sampled image.

Epoch-0-RandomImage-994-ClassNumber-14



Figure 0.42: Sample Image 5: bounding box index 14

The figure shown below (see Figure 0.43) displays a bounding box proposal for the sixth sampled image.

Epoch-0-RandomImage-1086-ClassNumber-6



Figure 0.43: Sample Image 6: bounding box index 6

The figure shown below (see Figure 0.44) displays bounding box proposals for the seventh sampled image.

Epoch-0-RandomImage-3259-ClassNumber-13



(a) Sample Image 7: bounding box index 13

Epoch-0-RandomImage-3259-ClassNumber-14



(b) Sample Image 7: bounding box index 14

Figure 0.44: Sample Image 7: bounding boxes for various indices

The figure shown below (see Figure 0.45) displays a bounding box proposal for the eighth sampled image.

Epoch-0-RandomImage-2153-ClassNumber-14



Figure 0.45: Sample Image 8: bounding boxes index 14

The figure shown below (see Figure 0.46) displays a bounding box proposal for the ninth sampled image.

Epoch-0-RandomImage-1423-ClassNumber-14



(a) Sample Image 9: bounding box index 14

Epoch-0-RandomImage-1423-ClassNumber-4



(b) Sample Image 9: bounding box index 4

Figure 0.46: Sample Image 9: bounding box for various indices

The figure shown below (see Figure 0.47) displays a bounding box proposal for the tenth sampled image.

Epoch-0-RandomImage-555-ClassNumber-14

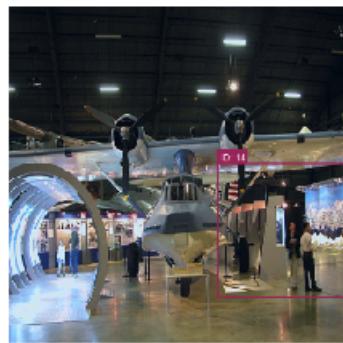


Figure 0.47: Sample Image 10: bounding box index 14

The figure shown below (see Figure 0.48) displays bounding box proposals for the first sampled image at the last epoch.

Epoch-5-RandomImage-12-ClassNumber-14



(a) Sample Image 1: bounding box index 14

Epoch-5-RandomImage-12-ClassNumber-12



(b) Sample Image 1: bounding box index 12

Figure 0.48: Sample Image 1: bounding box index 14 at last epoch

The figure shown below (see Figure 0.49) displays a bounding box proposal for the second sampled image at the last epoch.

Epoch-5-RandomImage-645-ClassNumber-6



Figure 0.49: Sample Image 2: bounding box index 6 at last epoch

The figure shown below (see Figure 0.50) displays a bounding box proposal for the third sampled image at the last epoch.

Epoch-5-RandomImage-104-ClassNumber-14

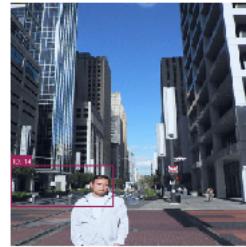


Figure 0.50: Sample Image 3: bounding box index 14 at last epoch

The figure shown below (see Figure 0.51) displays a bounding box proposal for the fourth sampled image at the last epoch.

Epoch-5-RandomImage-211-ClassNumber-6



Figure 0.51: Sample Image 4: bounding box index 6 at last epoch

The figure shown below (see Figure 0.52) displays a bounding box proposal for the fifth sampled image at the last epoch.

Epoch-5-RandomImage-994-ClassNumber-14



Figure 0.52: Sample Image 5: bounding box index 14 at last epoch

The figure shown below (see Figure 0.53) displays a bounding box proposal for the sixth sampled image at the last epoch.

Epoch-5-RandomImage-1086-ClassNumber-6



Figure 0.53: Sample Image 6: bounding box index 6 at last epoch

The figure shown below (see Figure 0.54) displays bounding box proposals for the seventh sampled image at the last epoch.

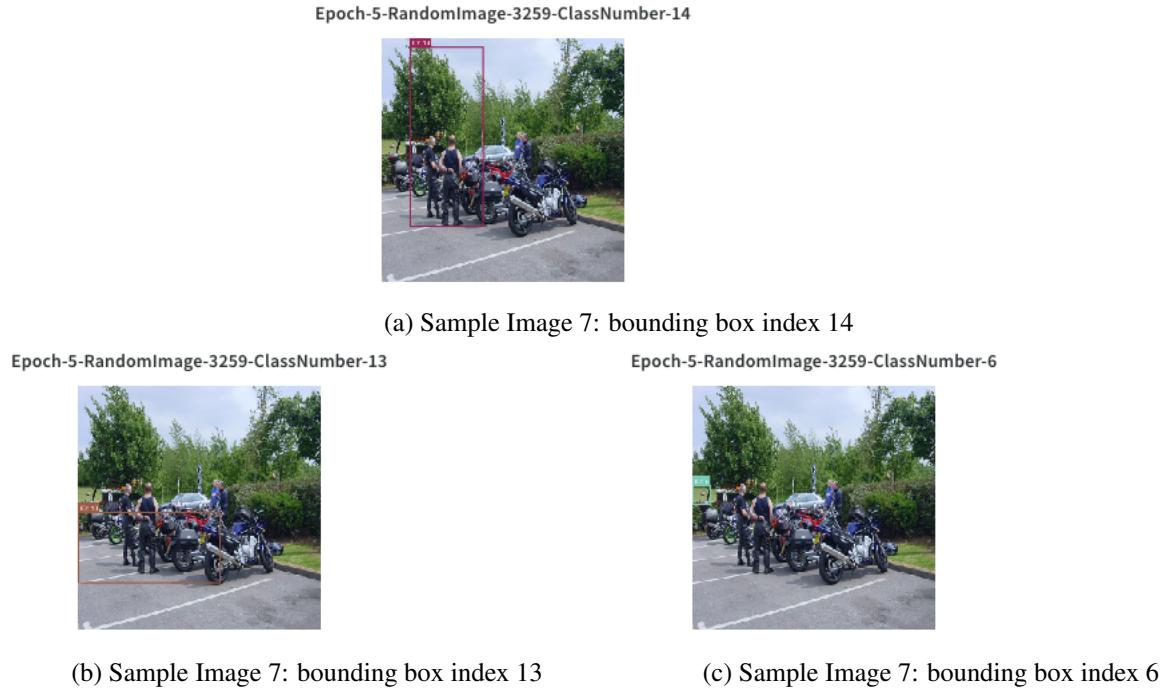


Figure 0.54: Sample Image 7: bounding boxes of various indices at last epoch

The figure shown below (see Figure 0.55) displays a bounding box proposal for the eighth sampled image at the last epoch.

Epoch-5-RandomImage-2153-ClassNumber-14



Figure 0.55: Sample Image 8: bounding box index 14

The figure shown below (see Figure 0.56) displays bounding box proposals for the ninth sampled image at the last epoch.

Epoch-5-RandomImage-1423-ClassNumber-4



(a) Sample Image 9: bounding box index 4

Epoch-5-RandomImage-1423-ClassNumber-14



(b) Sample Image 9: bounding box index 14

Figure 0.56: Sample Image 9: bounding boxes of various indices at last epoch

The figure shown below (see Figure 0.57) displays a bounding box proposal for the tenth sampled image at the last epoch.

Epoch-5-RandomImage-555-ClassNumber-14

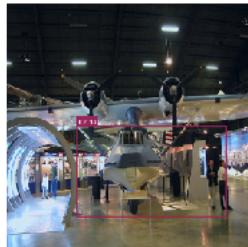


Figure 0.57: Sample Image 10: bounding box index 14

Final mAP and class-wise AP values (for the class indices 7, 11, and 17) are:

- mAP: 0.5846
- AP 7: 0.7387
- AP 11: 0.5421
- AP 17: 0.5858

References

- N/A