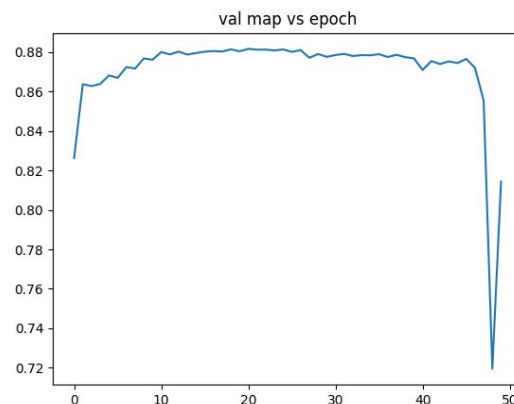
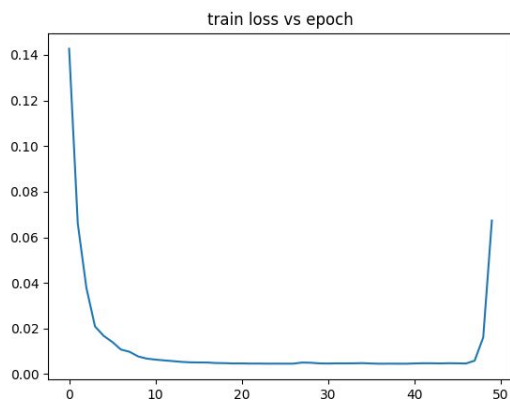
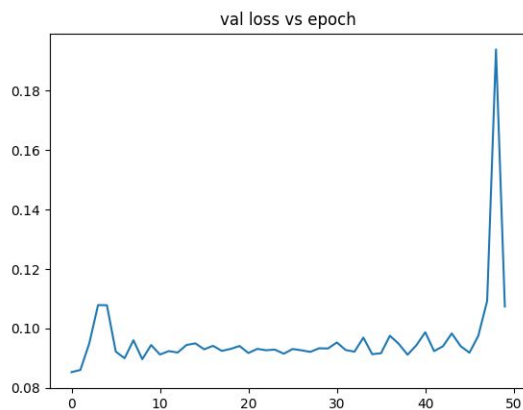


Woong Wen Tat (1002323)
Teo Kai Xiang (1002036)

Report

We achieved MAP of 0.84430 on the test set on the VOCPascal 2012 Object Classification Competition (comp2). We have published our model which is number 20 on the global leaderboard and hosted our GUI on wentat.ml. Our model is a Pytorch model Resnet 50 with transfer learning (pretrained on Imagenet data). Since this is a 20 class classification problem, we replaced the final fully connected (fc) layer with a 20 output fc layer. We implemented our own loss, a 20 class binary cross entropy loss(BCE) to fit this problem which is implemented using Pytorch BCEWithLogitsLoss. We do not have any learning rate schemes, however, we did quite a bit of feature augmentations in our model. During training, we enlarged the image to be 400x400, which is then followed by a five crop and normalization. During inference, we enlarged the image to 280x280, performed a five crop and normalization. Our 5 crop uses the max function instead of mean. The idea is that there can be objects at different corner of the image that the center crop would miss. To choose our best model parameters, we saved the model with the lowest validation loss over 50 epochs.





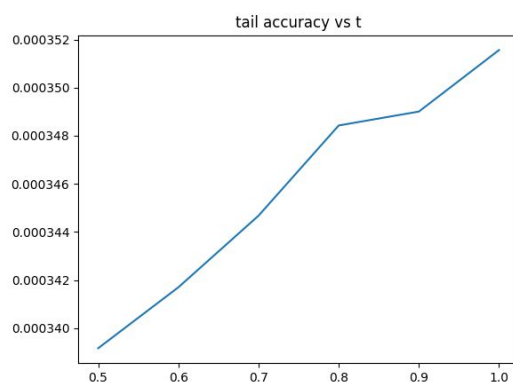
Hyperparameters used in evaluation

Learning rate = 0.1

Optimizer - Stochastic gradient descent only on layer 4 and fc layer, momentum = 0.9, weight decay = $1e-4$

Results

1. Accuracy is not a good measure for this problem because there are more pictures without the object than images with the object. Hence the accuracy can easily be made very high for each class, as an easy classifier could just don't predict any labels for any datapoints and still have a high accuracy. However for average precision, it is true positive over predicted positive so the measure would be more representative of the performance of the neural network.
2. The tail accuracy graph is as below:



Classwise Test mAP

Score

| Competition | Category | Score |
|-------------------------------|-------------|---------|
| Object Classification (comp2) | aeroplane | 0.97830 |
| Object Classification (comp2) | bicycle | 0.83960 |
| Object Classification (comp2) | bird | 0.93090 |
| Object Classification (comp2) | boat | 0.88120 |
| Object Classification (comp2) | bottle | 0.63030 |
| Object Classification (comp2) | bus | 0.88700 |
| Object Classification (comp2) | car | 0.80830 |
| Object Classification (comp2) | cat | 0.95780 |
| Object Classification (comp2) | chair | 0.72440 |
| Object Classification (comp2) | cow | 0.87180 |
| Object Classification (comp2) | diningtable | 0.77050 |
| Object Classification (comp2) | dog | 0.94440 |
| Object Classification (comp2) | horse | 0.92950 |
| Object Classification (comp2) | motorbike | 0.91020 |
| Object Classification (comp2) | person | 0.95370 |
| Object Classification (comp2) | pottedplant | 0.54640 |
| Object Classification (comp2) | sheep | 0.87840 |
| Object Classification (comp2) | sofa | 0.69340 |
| Object Classification (comp2) | train | 0.94260 |
| Object Classification (comp2) | tvmonitor | 0.80810 |
| Object Classification (comp2) | | 0.84430 |

Our submitted file includes the code for training and evaluation as well as the code to start a flask server (need to put the VOCdevkit in the static folder)