Course: Deep Learning for Computer Vision 5 November 2019
Name: Zeynep Cankara
Student Id: T08902113

Homework 02

Problem 1:

Part 1.1:

Which data augmentation techniques being used:

- Applied random horizontal flips with probability of 0.5.
- Applied random rotation in between -30° to 30° with probability of 0.5.

How data normalized:

- I used mean and standard deviation of the ImageNet to normalize the dataset.

Part 1.2.1 and Part 1.2.2:

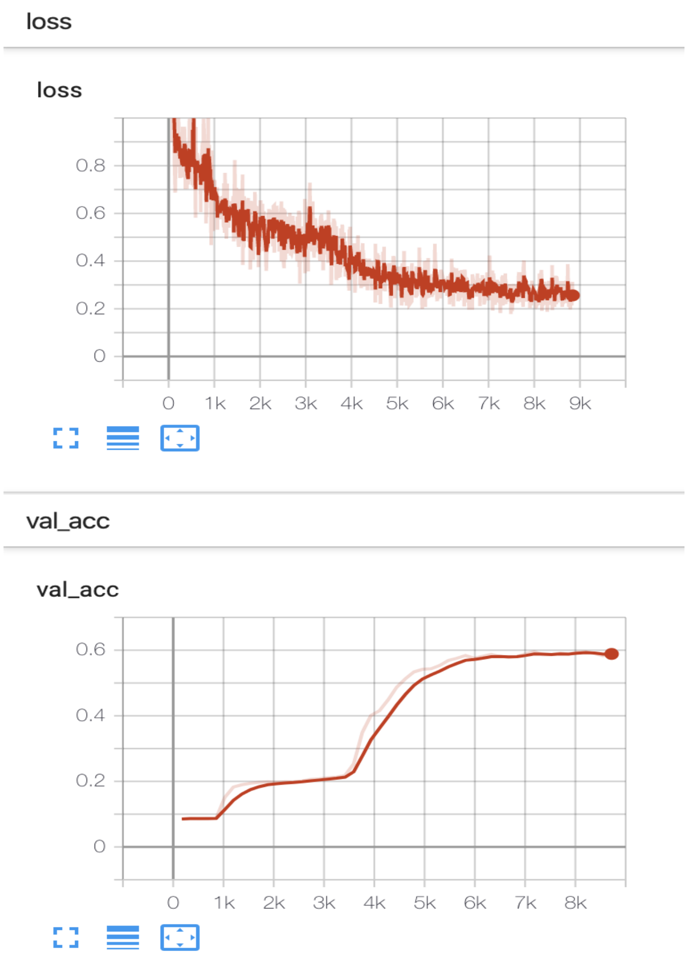


Figure 1: First training: ResNet18 parameters are freezed

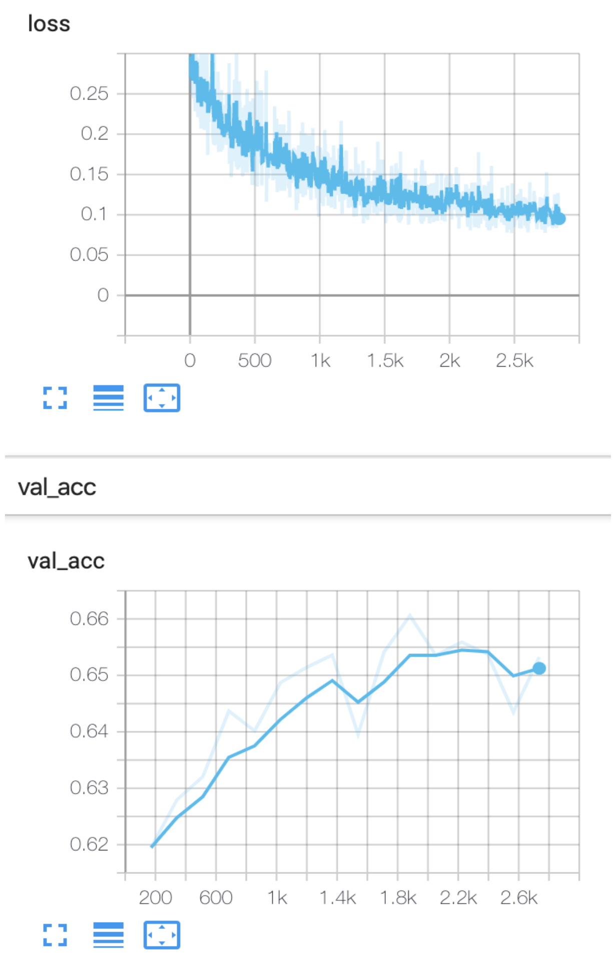


Figure 2: Second Training: Un-freezing the ResNet18 parameters and Fine Tune

Part 1.3:

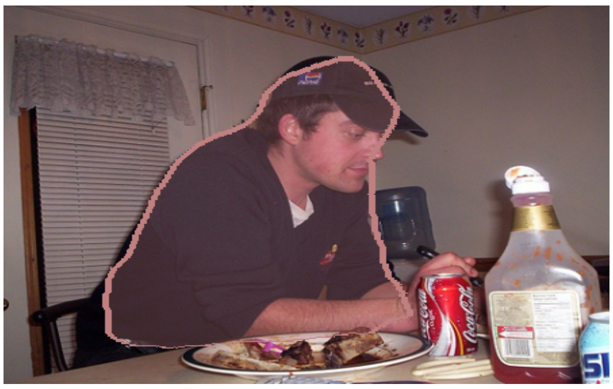


Figure 3: Class 1: Person

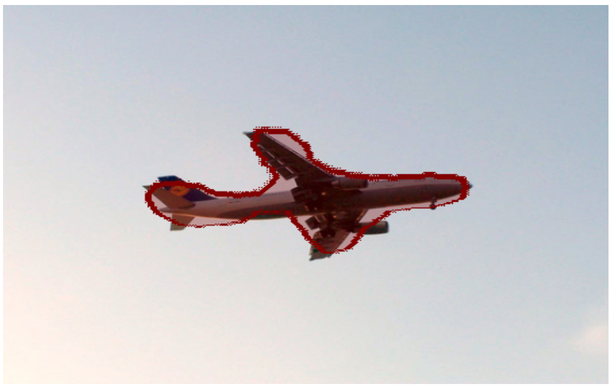


Figure 4: Class 2: Aeroplane

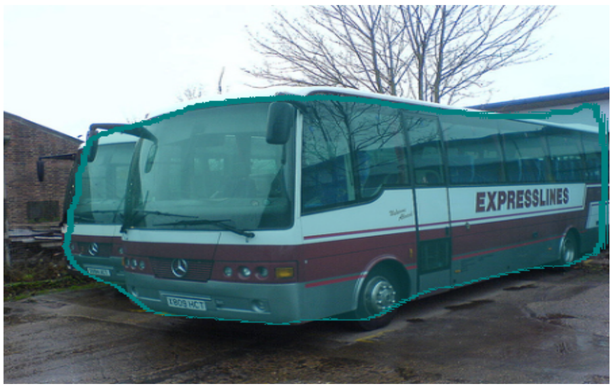


Figure 5: Class 3: Bus

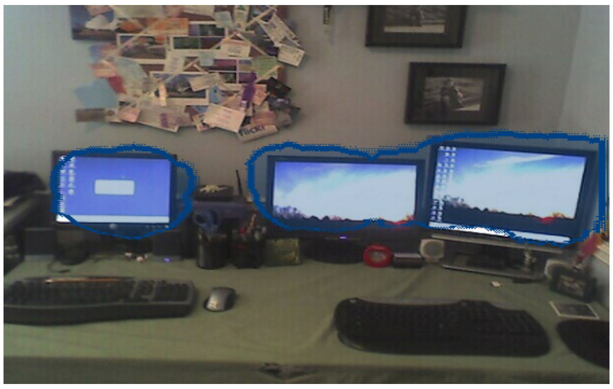


Figure 6: Class 4: TV/Monitor

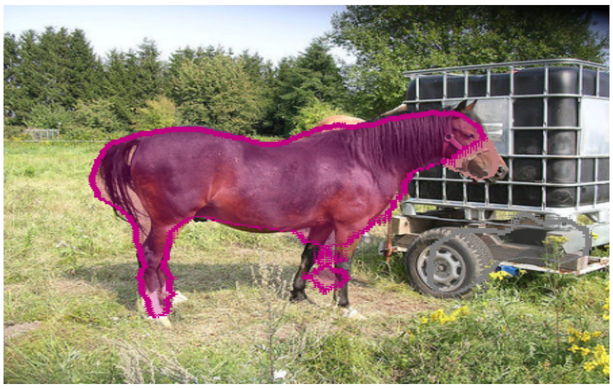


Figure 7: Class 5: Horse

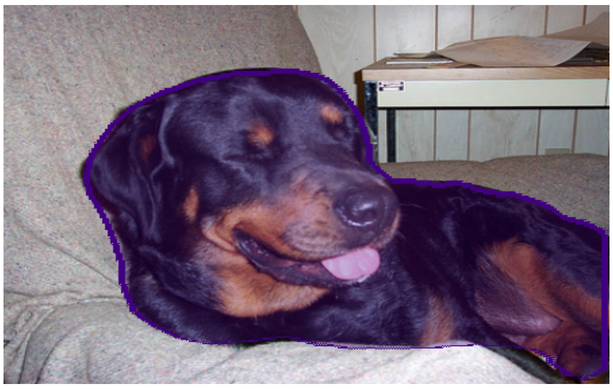


Figure 8: Class 6: Dog

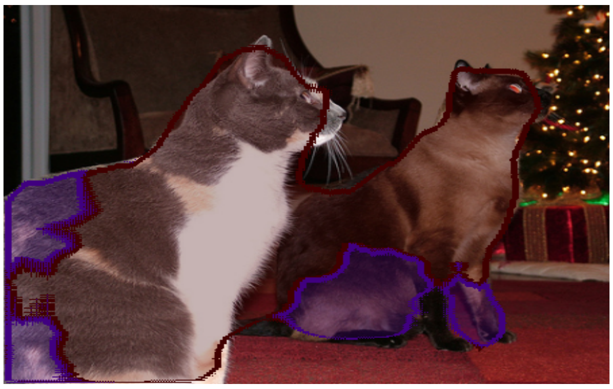


Figure 9: Class 7: Cat

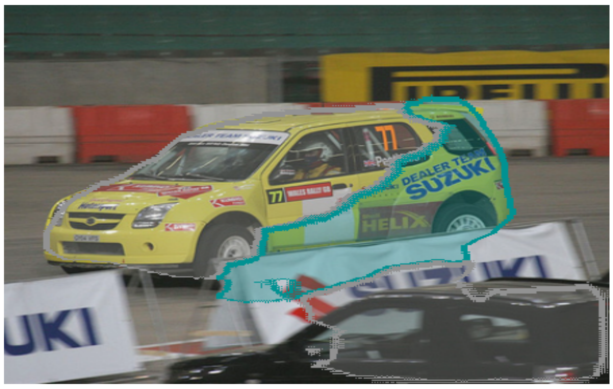
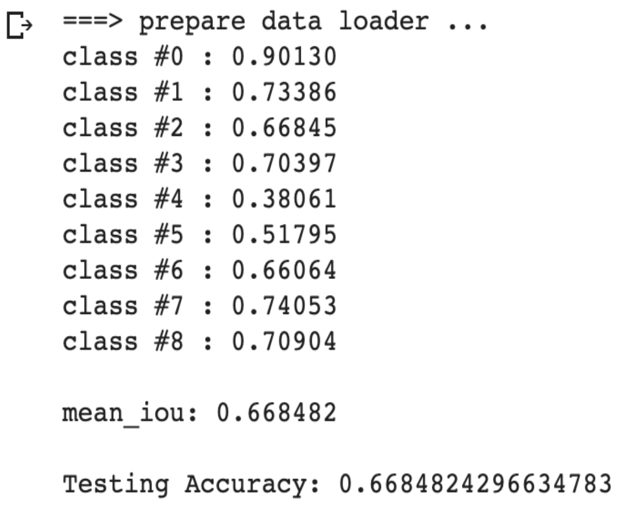


Figure 10: Class 8: Car

Part 1.4:



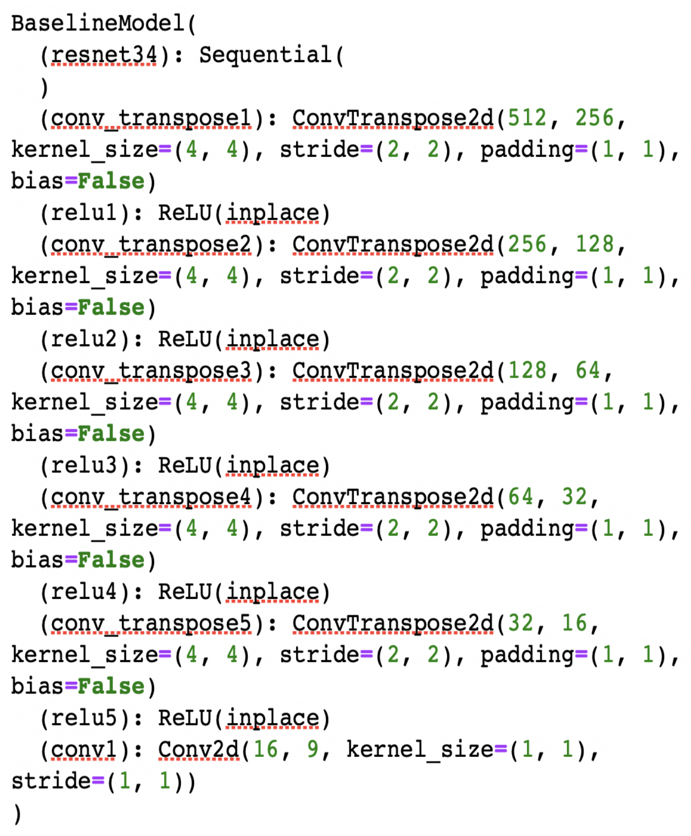
```
===> prepare data loader ...
class #0 : 0.90130
class #1 : 0.73386
class #2 : 0.66845
class #3 : 0.70397
class #4 : 0.38061
class #5 : 0.51795
class #6 : 0.66064
class #7 : 0.74053
class #8 : 0.70904

mean_iou: 0.668482

Testing Accuracy: 0.6684824296634783
```

Figure 11: mIoU scores of the Baseline Model

The class 0 which is the class belonging background has the highest IoU score. On the other hand, the class 4 which is the class belonging to the tv/monitor has the lowest IoU score.

The reason behind this result can be related with the content of the dataset given in the hw2 and the ImageNet dataset. The ResNet18 classifiers are pre-trained with the ImageNet dataset so they learnt detecting the objects which are most common in the ImageNet dataset. Additionally, the top feature extractors of the baseline model are trained with the dataset given in the hw2. Most part of the images in the dataset contains pixels belonging the background class and there are fewer pixels which labelled as tv/monitor class.

Part 2.1:



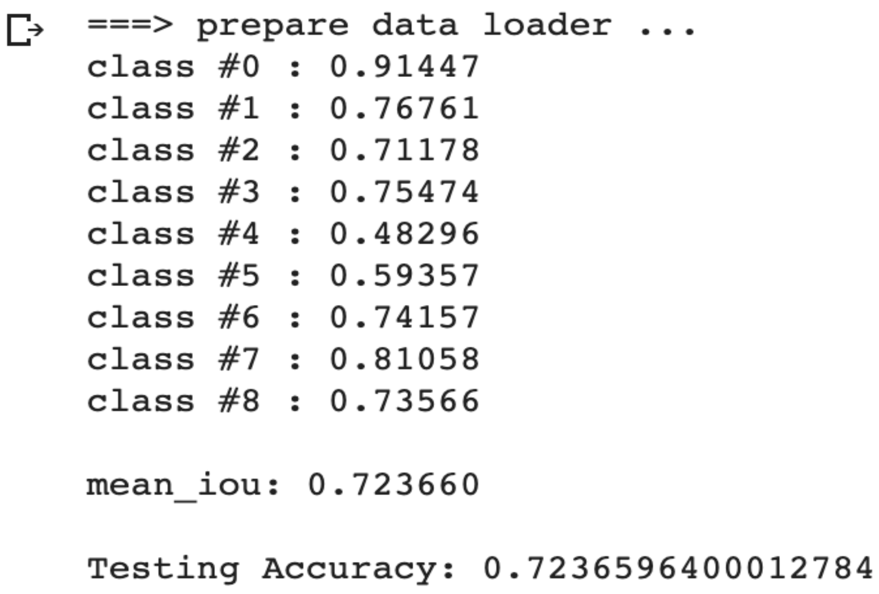
```
BaselineModel(
  (resnet34): Sequential(
  )
  (conv_transpose1): ConvTranspose2d(512, 256,
kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  (relu1): ReLU(inplace)
  (conv_transpose2): ConvTranspose2d(256, 128,
kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  (relu2): ReLU(inplace)
  (conv_transpose3): ConvTranspose2d(128, 64,
kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  (relu3): ReLU(inplace)
  (conv_transpose4): ConvTranspose2d(64, 32,
kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  (relu4): ReLU(inplace)
  (conv_transpose5): ConvTranspose2d(32, 16,
kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  (relu5): ReLU(inplace)
  (conv1): Conv2d(16, 9, kernel_size=(1, 1),
stride=(1, 1))
)
```

Figure 12: Improved Baseline Model Architecture

Part 2.2:

According to the training information belonging to the Baseline Model, the model converges after 7k epochs which is pushed forward by unfreezing and fine tuning the pretrained Resnet18 parameters. The fine tuned network also seems to converge after 2.5k epochs from start of the new training. The validation accuracy does not go down as I trained further the network which is a good indication of the model does not start to over-fit the data yet. So I decided to choose a deeper neural network architecture rather than using regularization techniques such as Dropout. I choose ResNet34 because it's demonstrated performance on benchmark datasets outperforms ResNet18. I keep architecture of the other layers same.

Part 2.3:



```
===> prepare data loader ...
class #0 : 0.91447
class #1 : 0.76761
class #2 : 0.71178
class #3 : 0.75474
class #4 : 0.48296
class #5 : 0.59357
class #6 : 0.74157
class #7 : 0.81058
class #8 : 0.73566

mean_iou: 0.723660

Testing Accuracy: 0.7236596400012784
```

Figure 13: Validation Set mIoU Score of the improved model

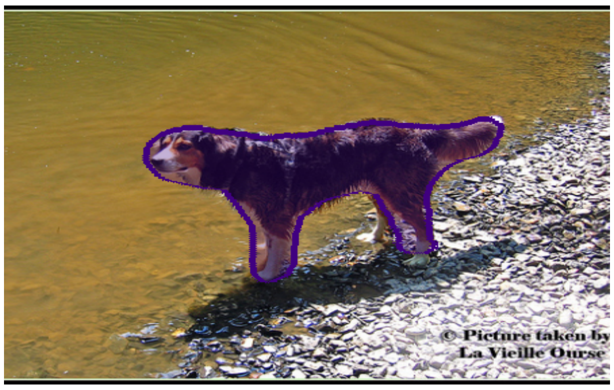


Figure 14: Improved Model Prediction: Class Dog

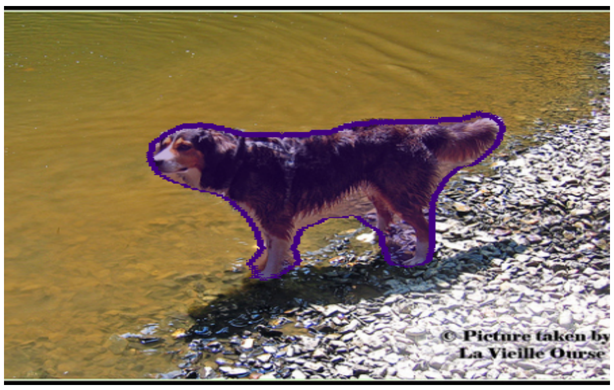


Figure 15: Baseline Model Prediction: Class Dog

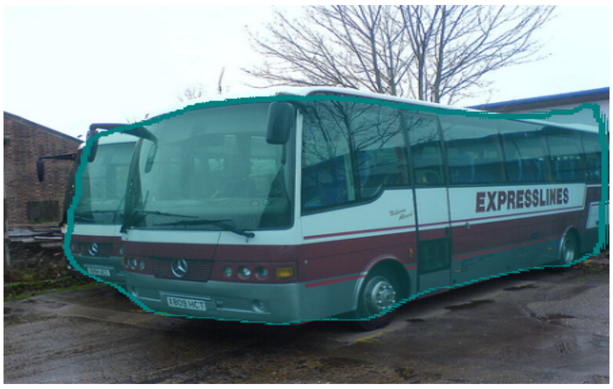


Figure 16: Improved Model Prediction: Class Bus

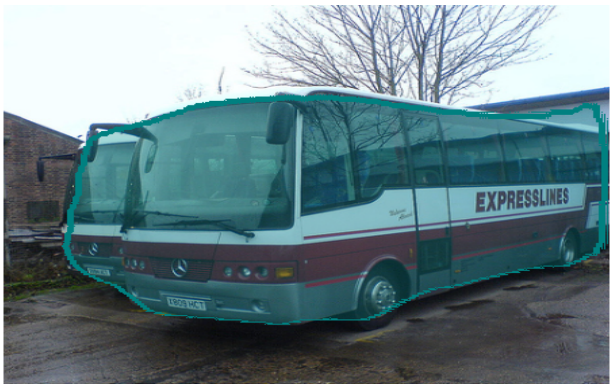


Figure 17: Baseline Model Prediction: Class Bus

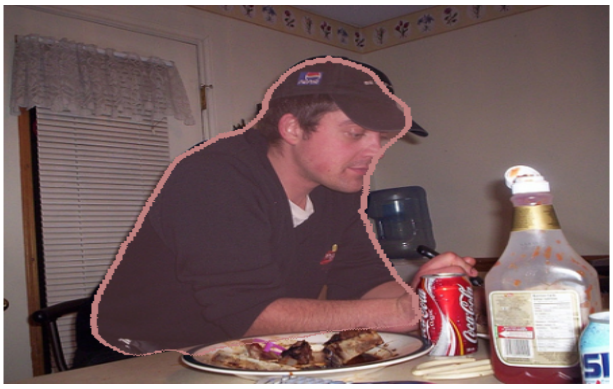


Figure 18: Improved Model Prediction: Class People

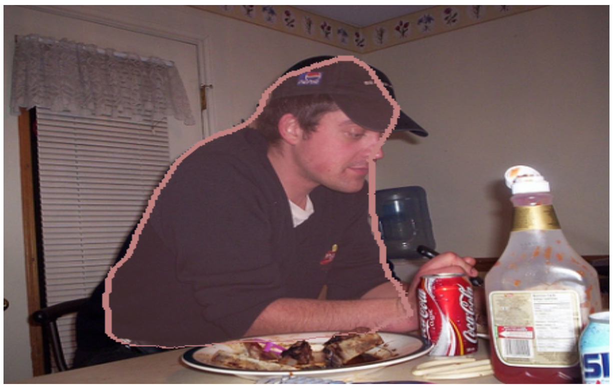


Figure 19: Baseline Model Prediction: Class People

References

I used Google Colab's GPU: Tesla K80

1. https://pytorch.org/docs/stable/torchvision/