Asadullah Hill Galib
CSE 891-001: Deep Learning
Programming Assignment 2
October, 27

1 Q2 in 1.1 (Upsampling and Pooling)

After running the PoolUpSampleNet for 25 epochs, the results look quite good to me. Both the training and validation curve is decreasing consistently without any abnormal spikes. Also, the output images seem quite a bit reasonable.

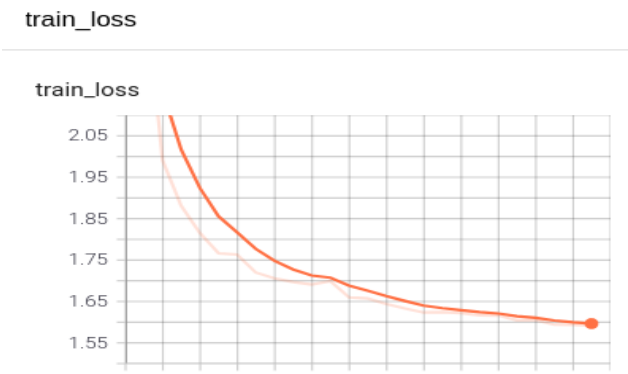

Figure 1: Training (25 Epochs)

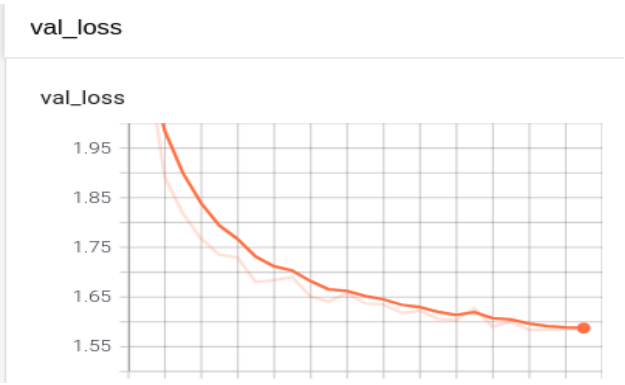

Figure 2: Validation (25 Epochs)

2 Q3 in 1.1 (Upsampling and Pooling)

Number of weights:

```
module1.0.weight torch.Size([32, 1, 3, 3])
module1.0.bias torch.Size([32])
module1.2.weight torch.Size([32])
module1.2.bias torch.Size([32])
module2.0.weight torch.Size([64, 32, 3, 3])
module2.0.bias torch.Size([64])
module2.2.weight torch.Size([64])
module2.2.bias torch.Size([64])
module3.0.weight torch.Size([32, 64, 3, 3])
module3.0.bias torch.Size([32])
module3.2.weight torch.Size([32])
module3.2.bias torch.Size([32])
module4.0.weight torch.Size([24, 32, 3, 3])
module4.0.bias torch.Size([24])
module4.2.weight torch.Size([24])
module4.2.bias torch.Size([24])
module_last.weight torch.Size([24, 24, 3, 3])
module_last.bias torch.Size([24])
```

Figure 3: Model Weights and biases

3 Q2 in 1.2 (Strided and Transposed Convolution)

After running the ConvTransposeNet for 25 epochs,

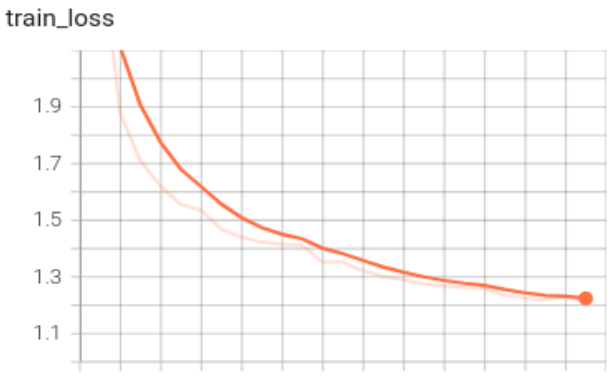

Figure 4: Training (25 Epochs)

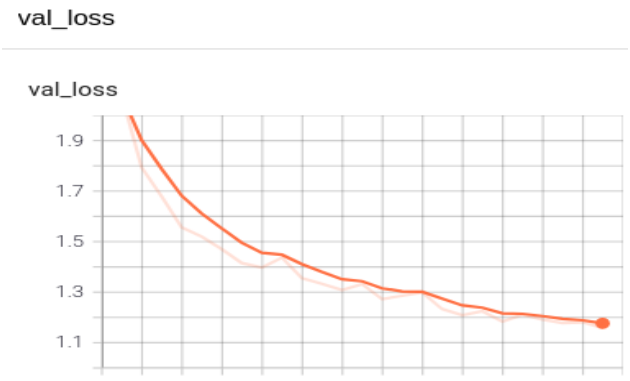

Figure 5: Validation (25 Epochs)

4 Q3 in 1.2 (Strided and Transposed Convolution)

The result is better than the PoolUpsampleNet. Yes, the ConvTransposeNet model results in lower validation loss (around 1.18) than PoolUpsampleNet (around 1.59). It works better as its deconvolution learns to decode the compressed representation rather than using any upsampling scheme. Its like an encoder-decoder scheme, which is more flexible and learnable.

5 Q3, Q4, Q5 in 1.2 (Strided and Transposed Convolution)

5.1 Q3 in 1.2

The result is better than the PoolUpsampleNet. Yes, the ConvTransposeNet model results in lower validation loss (around 1.18) than PoolUpsampleNet (around 1.59). It works better as its deconvolution learns to decode the compressed representation rather than using any upsampling scheme. Its like an encoder-decoder scheme, which is more flexible and learnable.

5.2 Q4 in 1.2

For nn.Conv2d, the padding parameter should be 2. (as Floor[(input - kernel + 2*padding)/stride] + 1 = output). For nn.ConvTranspose2d, the padding parameter should be 2 and output padding should be 1. (as (input - 1)*stride + Kernel - 2*padding = output).

5.3 Q5 in 1.2

After retraining the model using different batch sizes and fixed number of epoch, the validation loss increases with the increase of batch sizes (validation loss: 1.11 for batch size 32, 1:16 for batch size 64, 1.24 for batch size 128, and so on). In terms of final image quality, with the increase of batch size, the images are having less noise (kind of), but color classification seem less appropriate.

6 Q2 in 1.3 (Skip Connections)

After running the UNet for 25 epochs,

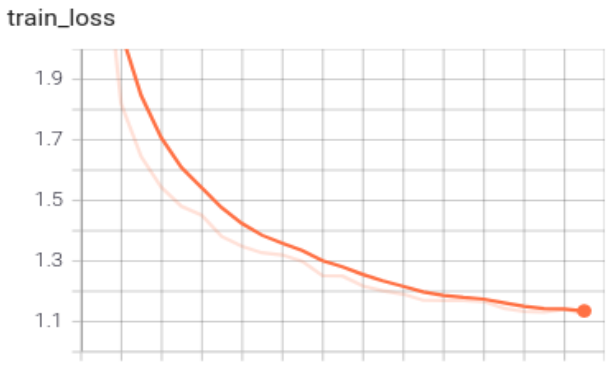

Figure 6: Training (25 Epochs)

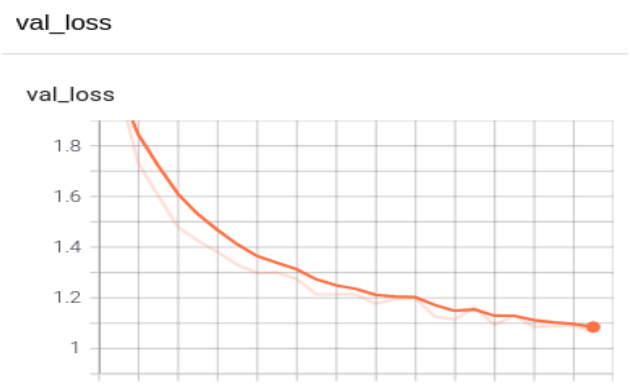

Figure 7: Validation (25 Epochs)

7 Q3 in 1.3 (Skip Connections)

The result is better than the PoolUpsampleNet and ConvTransposeNet models. It has a lower validation loss around 1.06.

According to those curves and final image, skip connections improve the validation loss and accuracy. Also, skip connection improves the output quality fairly.

It works better (firstly) as its creates additional path for backpropagation which might be useful to avoid vanishing gradient problem. (Secondly) It works better as the skip connection provides features to the later stages, which helps to recover spatial representation that might be lost while downsampling.

8 Q1 in 2.1

Training and validation plots using cross entropy loss:

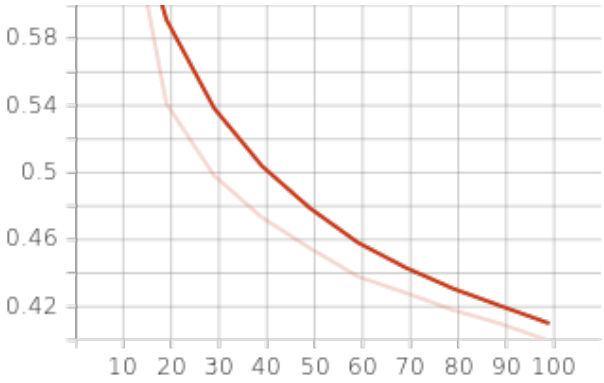

Figure 8: Training (10 Epochs)

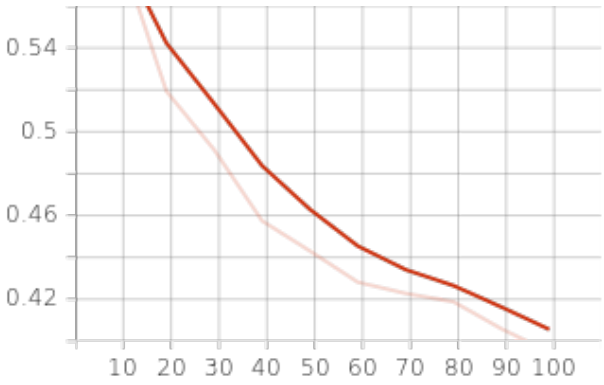

Figure 9: Validation (10 Epochs)

mIoU plots with respect to epochs:

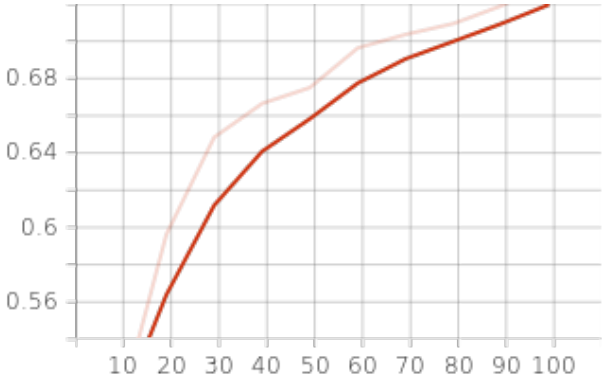

Figure 10: mIoU using cross-entropy loss(10 Epochs)

Using cross-entropy loss, the best mIoU is: 0.72.
Both training and validation losses are decreased consistently, and mIoU is increasing steadily.

9 Q2 in 2.1

Visualizations of input image, ground truth segmentation map and the predicted segmentation map for cross entropy loss:

Figure 11: Input

Figure 12: Ground Truth

Figure 13: Output

10 Q1 in 2.2

Training and validation plots using IoU loss:

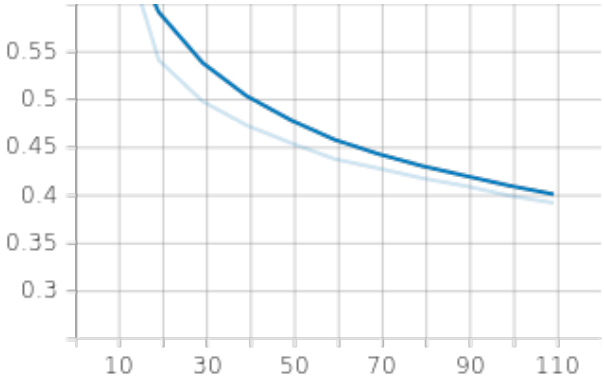

Figure 14: Training (10 Epochs)

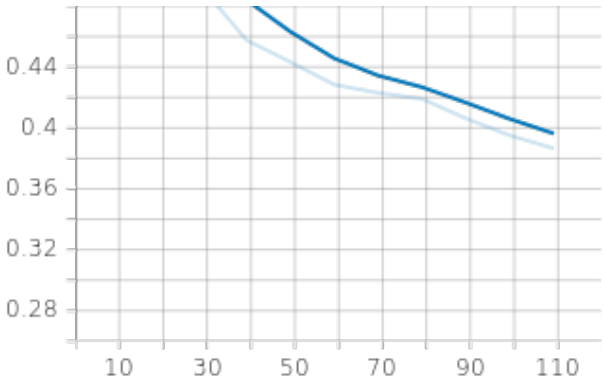

Figure 15: Validation (10 Epochs)

mIoU plots with respect to epochs:

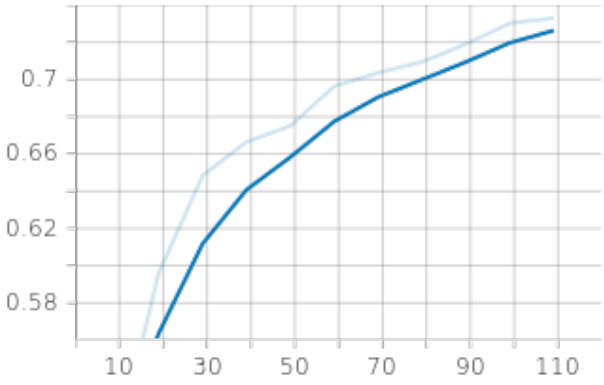

Figure 16: mIoU using IoU loss (10 Epochs)

Using cross-entropy loss, the best mIoU is: 0.74.
Both training and validation losses are decreased consistently, and mIoU is increasing steadily.
The performance (mIoU) is better than the previous one (with cross-entropy loss).

11 Q2 in 2.2

Visualizations of input image, ground truth segmentation map and the predicted segmentation map for IoU loss:

Figure 17: Input

Figure 18: Ground Truth

Figure 19: Output