CENG 483

Introduction to Computer Vision

Fall 2021-2022

Take Home Exam 3

Image Colorization

Student ID: 2375574

Please fill in the sections below only with the requested information. If you have additional things to mention, you can use the last section. Please note that all of the results in this report should be given for the **validation set** by default, unless otherwise specified. Also, when you are expected to comment on the effect of a parameter, please make sure to **fix** other parameters. You may support your comments with visuals (i.e. loss plot).

1 Baseline Architecture (30 pts)

Based on your qualitative results (do not forget to give them),

- Discuss effect of the number of conv layers:
- Discuss effect of the kernel size(except the last conv layer):
- Discuss effect of the number of kernels(except the last conv layer):
- Discuss effect of the learning rate by choosing three values: a very large one, a very small one and a value of your choice:

As number of convolutional layers increases, the network capacity increases. Hence, model can fit to data more strictly. Learning the model with 4-convolution requires more time, since capacity is much greater than 2-conv and 1-conv networks and we can observe this phenomenon in Figure-1 where red curve decreases slower than green and orange curves. One interesting observation is that why 4-conv has more train-loss than lower capacity networks, since theoretically it has more capacity to overfit/memorize the train-data, the reason is that probably we haven't gave him enough epochs to memorize the train-data since max-epochs is restricted to be 100. However, my observation is that after 20-40 epochs, rate of change of 4-conv is greater than 2-conv and 1-conv meaning that it keeps decreasing more even if it's slow. Another interesting observation is that why train-loss and valid-losses are almost the same at every epoch, the reason is that problem is not very difficult, we are trying to colorize the facial images and also train data and valid data are very similar and balanced. So it's expected to get similar curves for both valid-loss and train-loss.
Common Hyperparameter Baseline in Exp:
learning-rate = 0.001, batch-size = 16, epoch = 100, kernel-num=8, kernel-size=3

I've experimented with different kernel sizes of 3 and 5. 2-4 conv layers different kernel sizes 3 5 tried for both 2-4 conv 5 kernel has less loss why? pooling effect with 5 kernel ? img 80x80x3 reminder
Common Hyperparameter Baseline in Exp:
learning-rate = 0.001, batch-size = 16, epoch = 100, kernel-num=8

2-4 conv layers different number of kernels 2 4 8 tried
Common Hyperparameter Baseline in Exp:
learning-rate = 0.001, batch-size = 16, epoch = 100, kernel-size=5

conv 2-4 learning rates 1 0.1 0.01 0.001 0.0001
Common Hyperparameter Baseline in Exp:
learning-rate = 0.001, batch-size = 16, epoch = 100, kernel-num=8, kernel-size=5

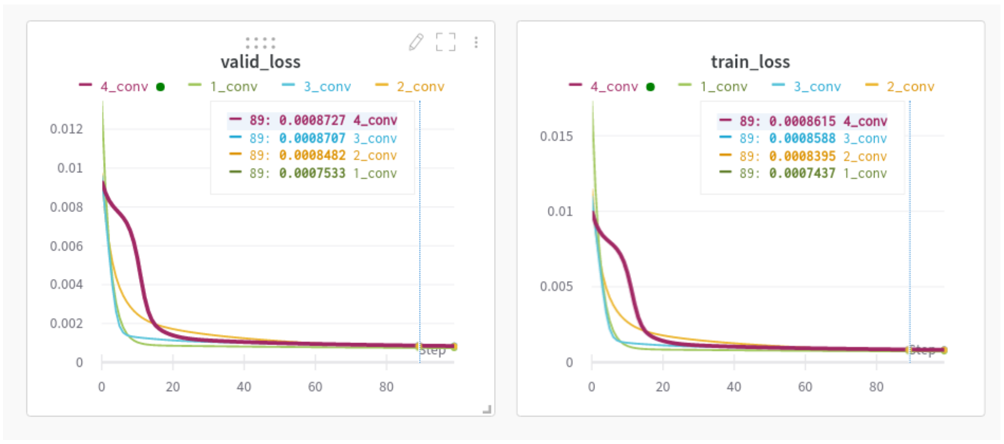


Figure 1: Effect of Different Conv Layers

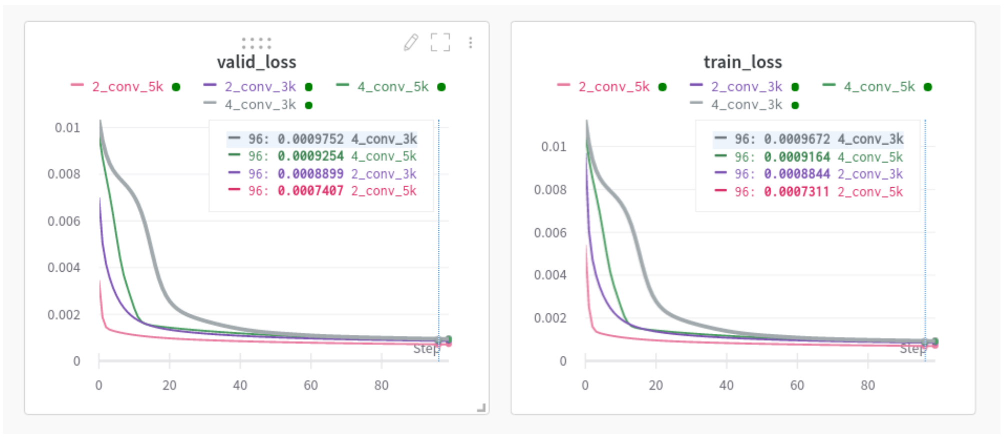


Figure 2: Effect of Different Kernel Sizes

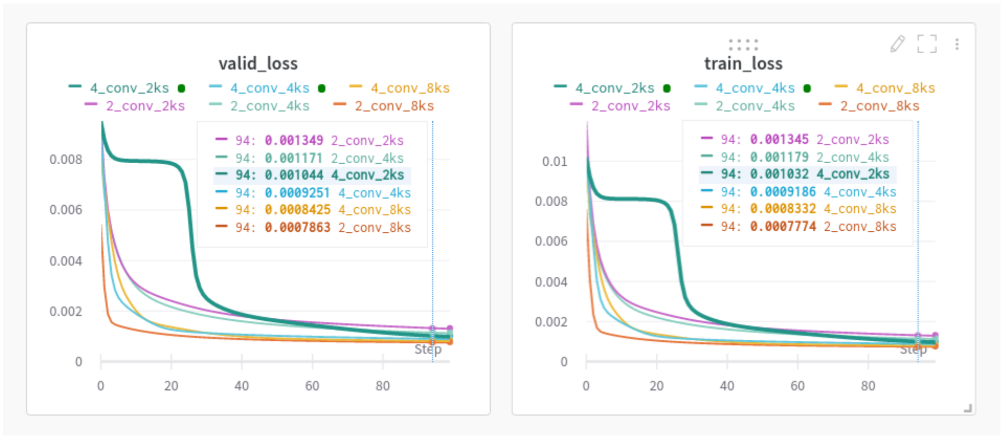


Figure 3: Effect of Different Number of Kernels

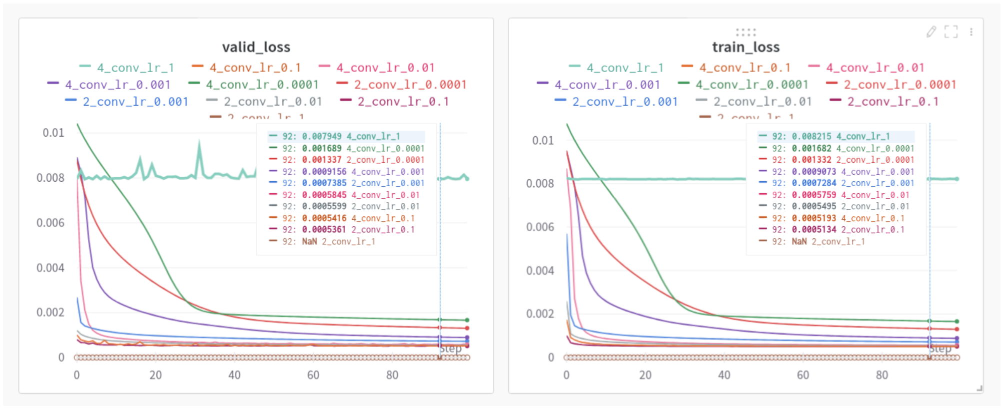


Figure 4: Effect of Different Learning Rates

2 Further Experiments (20 pts)

Based on your qualitative results (do not forget to give them),

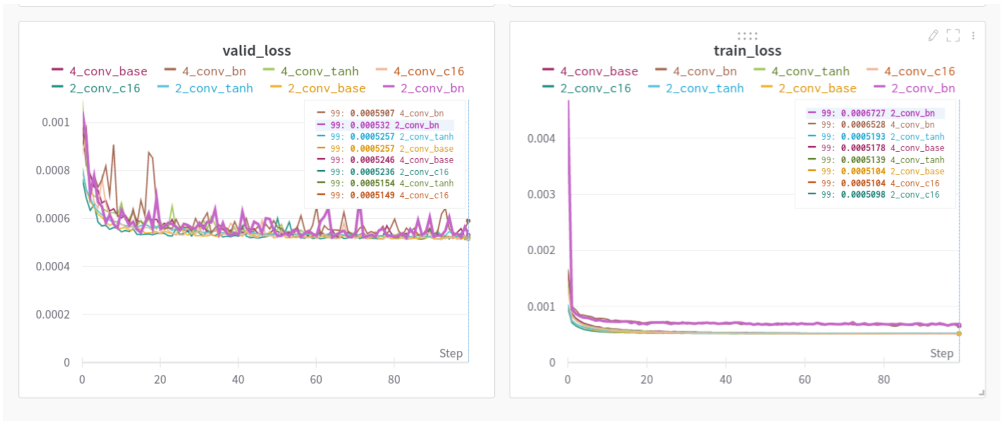
- Try adding a batch-norm layer (torch.nn.BatchNorm2d) into each convolutional layer. How does it affect the results, and, why? Keep it if it is beneficial.
- Try adding a tanh activation function after the very last convolutional layer. How does it affect the results, and, why? Keep it if it is beneficial.
- Try setting the number of channels parameter to 16. How does it affect the results, and, why? Keep it if it is beneficial.

I compared my results with respect to two baselines for different number of convolutional layers, 4-conv and 2-conv. Overall, my observations are aligned for both 2-conv and 4-conv. All of the experiments in this part are summarized in Figure-5 with following abbreviations; bn: batch-norm experiment, tanh: tanh experiment, c16: 16 channel experiment. 2-conv and 4-conv stands for 2 and 4 conv layer NN respectively.
Common Hyperparameter Baseline in Exp:
learning-rate = 0.1, batch-size = 16, epoch = 100, kernel-size=5

Usually, batch-norm increases the performance of NN since scaling eases optimization during learning. However, BatchNorm2d increased mse loss for both 2-conv and 4-conv, its loss is the highest among all observations in Figure-5. The reason might be because, the input and ouput data are already scaled to the interval of [-1,1], so model doesn't have any benefit in this configuration. However, it has some regularization effect on the weights and as a result it requires more data for training and increases the loss. Therefore, I don't see much of a benefit for batch-norm and I'm not planning to keep it under this configuration.

I've expected tanh to improve the model's accuracy and decreasing the loss. I've observed in Figure-5 that for 4-conv tanh decreases the loss as expected, however for conv-2 the behaviour is opposite, the loss is higher than 2-conv-base. This might be because of the fact that, when we change the network, even 1 single additional activation function yields another different optimization problem, hence our baseline hyperparameters eg. learning-rate might not be the optimal for that configuration. In addition, these loss curves correspond to avg loss per pixel and also all of these individual pixel values are scaled to the interval of [-1,1] which is relatively small. Hence all of these models result in a very low mse loss value and loss differences between the models are marginal, so we need to make use of accuracy checking mechanisms like 12-margin and also manual image inspection. For example, I observe that applying tanh in both 2-conv and 4-conv has removed exploding pixel values eg. small blue segments in the image which is an improvement, but not directly visible in mse loss. Tanh works because we are squeezing our network's output to [-1,1] and our labels are also in this range. However, if we didn't apply tanh to conv output, it would be theoretically unbounded which could be more difficult to learn, in a way we are guiding the network to learn better since we have prior knowledge about our ground truth interval.

Number of channels or number of kernels that I've experimented with so far were 2, 4, and 8 as mentioned in the HW description. In this experiment, I've set channel-out=16 for all conv layers except for the last one which had to be 3 (r,g,b). I've expected a higher channel number to decrease the loss and this phenomenon is observable in Figure-5 where losses are the lowest among all of the models in this part for both 4-conv and 2-conv. Having higher number of channels/kernels gives our model more capacity to extract more different features from images and the model makes use of more of these features to fit the problem/data in a better way as a result decreasing the loss.



```
99: 0.0005907 4_conv_bn
99: 0.000532 2_conv_bn
99: 0.0005257 2_conv_tanh
99: 0.0005257 2_conv_base
99: 0.0005246 4_conv_base
99: 0.0005236 2_conv_c16
99: 0.0005154 4_conv_tanh
99: 0.0005149 4_conv_c16
```

```
99: 0.0006727 2_conv_bn
99: 0.0006528 4_conv_bn
99: 0.0005193 2_conv_tanh
99: 0.0005178 4_conv_base
99: 0.0005139 4_conv_tanh
99: 0.0005104 2_conv_base
99: 0.0005104 4_conv_c16
99: 0.0005098 2_conv_c16
```

Figure 5: Effect of batch-norm, tanh, channels

3 Your Best Configuration (20 pts)

Using the best model that you obtain, report the following:

- The automatically chosen number of epochs(what was your strategy?):
- The plot of the training mean-squared error loss over epochs:
- The plot of the validation 12-margin error over epochs (see the3 text for details):
- At least 5 qualitative results on the validation set, showing the prediction and the target colored image:
- Discuss the advantages and disadvantages of the model, based on your qualitative results, and, briefly discuss potential ways to improve the model:

4 Your Results on the Test Set(30 pts)

This part will be obtained by us using the estimations you will provide. Please tell us how should we run your code in case of a problem:

5 Additional Comments and References

(if there any)