CENG 483

Introduction to Computer Vision

Spring 2018-2019

Take Home Exam 3

Image Colorization

Student Random ID: 72

This report introduces an analysis about image colorization by exploiting convolutional neural networks(CNNs) which are tuned with a variety of hyper-parameters with the aim of finding the configuration which provides the best results. The results in this report are obtained for the validation set provided. Some examples for the hyper-parameters that are tuned can be the learning rate, number of convolution layers, number of kernels in a convolution layer and kernel size. The related analysis of tuning these hyper-parameters are examined in details in the following sections.

1 Baseline Architecture (30 pts)

Before beginning the experiments, it should be noted that the given accuracy results are calculated with the provided evaluate.py script according to the 12-margin error. In addition, the optimal padding size for each layer is calculated with the following formula:

$(kernel\_size - 1)/2$

1.1 Effect of Changing Number of Convolutional Layers

Firstly, the effects of changing the number of conv layers in the model is examined below. Possible values for the number of conv layers are chosen as 1, 2 and 4. By fixing learning rate to 0.001, number of kernels to 8 and kernel size to $3x3$, the accuracy results obtained for different values of number of conv layers are given in the table below.

Table 1: Number of Conv Layers vs. Accuracy

Number of Conv Layers	Accuracy
1	0.68
2	0.69
4	0.67

From this table we can observe that increasing the number of conv layers does not mean increasing the accuracy. 2 layers have performed slightly better that 1 layer because adding more conv layers will result in extracting more features. However, this holds upto a limit. When the number of conv layers is increased to 4, the accuracy has decreased. This can be due to the fact that the CNN will tend to overfit because it will find the irregularities in the images when the number of conv layers is too high.

1.2 Effect of Changing Kernel Size

Secondly, in order to discuss the effect of changing the kernel size except the last conv layer, the values $3x3$ and $5x5$ are chosen as the possible values for the kernel size. Table-2 represents the accuracy results for these values of the kernel size for two different configurations by fixing the learning rate to 0.001.

Table 2: Kernel Size vs. Accuracy

Kernel Size	Accuracy	Number of Conv Layers	Number of Kernels
$3x3$	0.68	1	2
$5x5$	0.65	1	2
$3x3$	0.58	4	2
$5x5$	0.60	4	2

Kernel size determines the amount of how much fine details we would like to consider in the images. For example, with 3x3 kernel size we can capture fine details of the images whereas with 5x5 kernel size we can capture larger features. As it can be observed from the table above, when the number of layers is lower, having a smaller kernel size results in more accurate results. However, when the number of kernels is higher, having a larger kernel size results in more accurate results. Therefore, we can infer that kernel size is dependent on the other hyper-parameters such as the number of conv layers.

1.3 Effect of Changing Number of Kernels

Thirdly, we have experimented with different values for the number of kernels in the conv layers except the last conv layer. The table below shows the accuracy results related to the different values for the number of kernels by fixing learning rate to 0.001 and kernel size to $3x3$. When the number of conv layers is set to 1, there is only one conv layer in the model and it is required to have the number of kernels(number of output channels) parameter as 3. Because of this, we cannot observe the effect of changing the number of kernels by considering only one conv layer. So, for these experiments the number of conv layers is fixed to 2.

Table 3: Number of Kernels vs. Accuracy

Number of Kernels	Accuracy
2	0.64
4	0.67
8	0.69

By looking at Table 3, we can infer that when the number of kernels is increased, the accuracy also increases.

1.4 Effect of Changing Learning Rate

Lastly, the effect of changing the learning rate is discussed. The table below represents the accuracy results for three different values of the learning rate by holding other hyper-parameters constant. The number of conv layers is fixed to 4, the number of kernels is fixed to 2 and kernel size is fixed to $3x3$.

Table 4: Learning Rate vs. Accuracy

Learning Rate	Accuracy
0.001	0.58
0.01	0.71
0.1	0.50

From this table, we can observe that when the learning rate is too small or too large, the accuracy results are not so high. When it is increased from a very small value to a medium value, the accuracy increases significantly. However, after a threshold, increasing the learning rate will result in overfitting. Thus, it will not perform well on unseen data.

2 Further Experiments (20 pts)

After completing the experiments in the first section, we have experimented by adding a batch-normalization layer into each conv layer. The hyper-parameters are fixed as the following:

- Number of conv layers: 4
- Number of kernels: 2
- Kernel size: $3x3$
- Learning Rate: 0.01

Example accuracy results are provided below for the models with and without batch normalization layers.

Table 5: Batch-Norm Layer vs. Accuracy

Batch-Norm Layer	Accuracy
with Batch-Norm Layer	0.62
without Batch-Norm Layer	0.71

From the table above, we can see that adding a batch-normalization layer into the conv layers has decreased the accuracy results. Therefore, in this case it is not a good idea to proceed with the batch normalization layers. WHYYYYYYY???????????

Secondly, we have added a tanh activation function after the very last conv layer. Example accuracy results are provided below for the following configurations:

- Number of conv layers: 2
- Number of kernels: 8
- Kernel size: $3x3$
- Learning Rate: 0.01

Table 6: Tanh Layer vs. Accuracy

Tanh Layer	Accuracy
with Tanh Layer	0.72
without Tanh Layer	0.75

From the table above, we can infer that adding a Tanh activation function after the last conv layer is not a good idea since it decreases the accuracy results. WHYYYYYYY???????????

Lastly, after completing these experiments, we have tried setting the number of channels(kernels) parameter to 16 to observe its effects on the accuracy of the model. The table below is provided for the following values of the hyper-parameters:

- Number of conv layers: 2
- Number of kernels: 8
- Kernel size: $3x3$
- Learning Rate: 0.01

Table 7: Number of Kernels vs. Accuracy

Number of Kernels	Accuracy
8	0.75
16	0.74

Form the table above, we can infer that increasing the number of kernels does not always increase the accuracy. This is due to the fact that

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.

3 Your Best Configuration (20 pts)

The hyper-parameters tuned for the best configuration of the model is given as the following:

- Number of conv layers: 2
- Number of kernels: 8
- Kernel size: $3x3$
- Learning Rate: 0.01

- using batch normalization
- not using tanh after the last conv layer

The automatically chosen number of epochs is $VALUE$. The strategy that is followed to decide the optimal number of epochs is early stopping. This is a very beneficial technique for avoiding overfitting. While implementing early stopping strategy, the MSE loss for the validation set which is calculated every 5 epochs is considered. When the current loss is larger than the previous loss, training is stopped and the best model is chosen as the model corresponding to the previous loss calculation. By performing this calculation every 5 epochs instead of every epoch, the training time is decreased a bit and the strategy is not deceived by the little fluctuations in the loss.

The following graph shows the relationship between the training MSE-loss and the validation MSE-loss over the epochs.

$GRAPHPLOTPLOTPLOT$!!!!!!1

The following plot represents the 12-margin error over the epochs.

$GRAPHPLOTPLOTPLOT$!!!!!!1

The figure below shows the grayscale and colored versions of 9 different images in the validation dataset and the predicted images. The images in the first line are the grayscale versions and the images in the last line are the target colored versions while the images in the middle line are the predicted images for the best model obtained.

Figure 1: Image Predictions

Advantages/disadv of model........

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)

In order to run the best configuration, the code should be run as the way it is provided. It should be checked that in the *hps* dictionary, the hyper-parameters are set as the following:

- Number of conv layers: 2
- Number of kernels: 8
- Kernel size: $3x3$
- Learning Rate: 0.01

The results for the test set will be saved in a file named *estimations_test.npy*.