Assignment 2

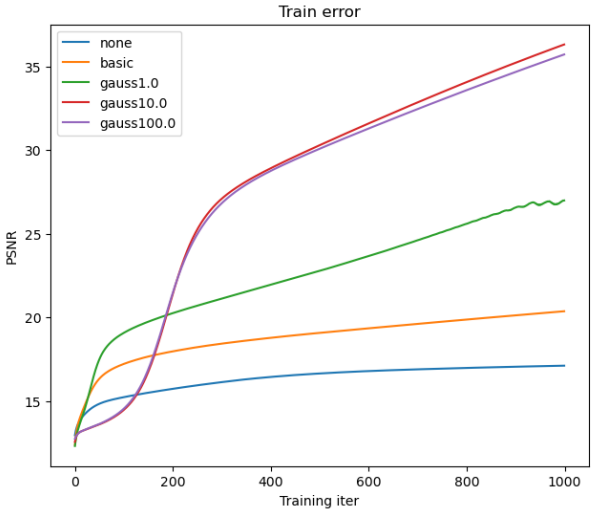
Name(s): Chengjun Lu
NetID(s): cl78

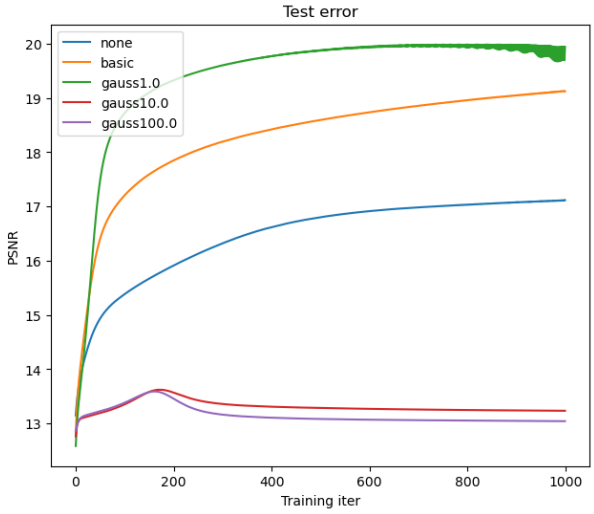
(I just realized I'm required to join a group (11) on canvas, but I'm working alone on this assignment).

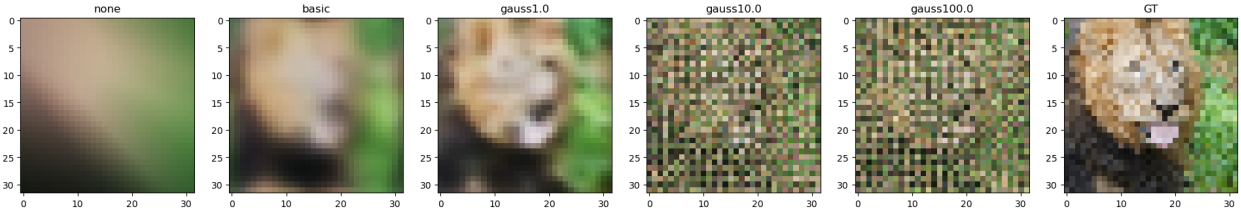
In each the following parts, you should insert the following:

- Train/test loss plots
- Qualitative outputs for GT, No encoding, Basic Positional Encoding, and Fourier Feature Encoding at three different scales

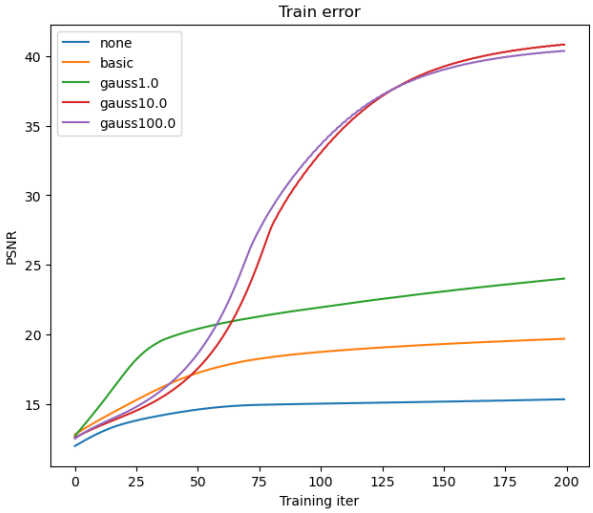
Part 1: Low resolution example - SGD

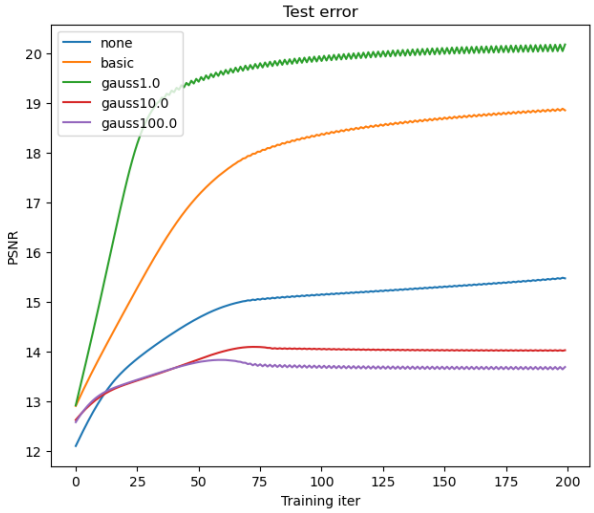


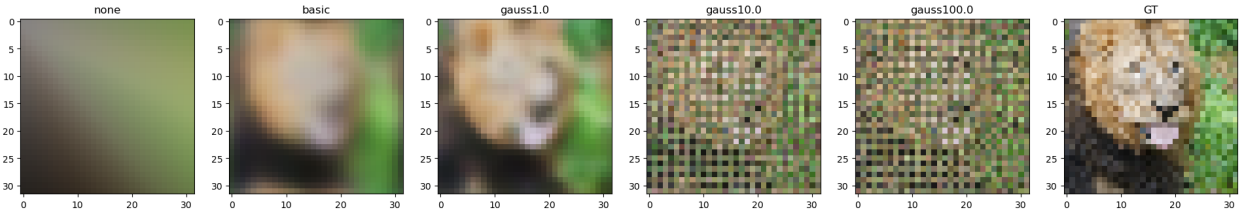




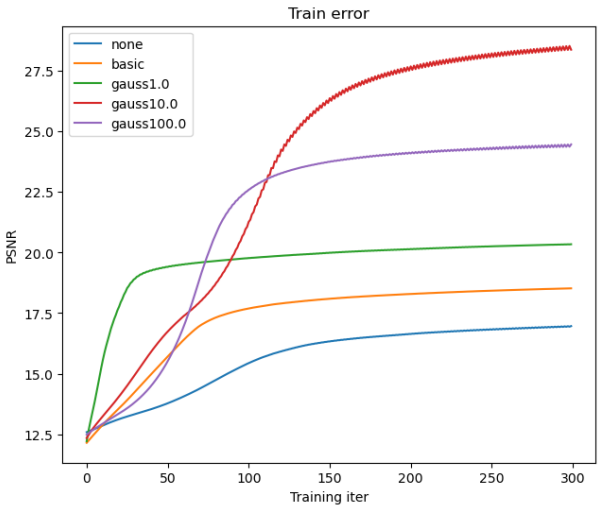
Part 2: Low resolution example - Adam

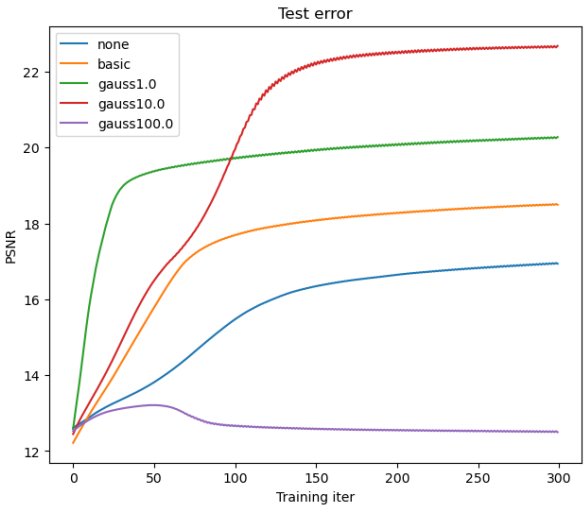


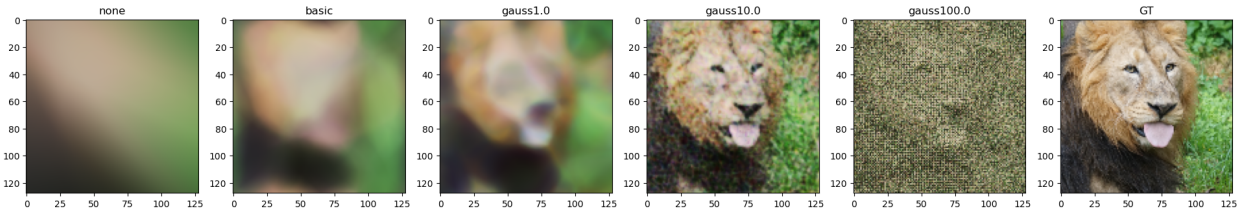




Part 3: High resolution example

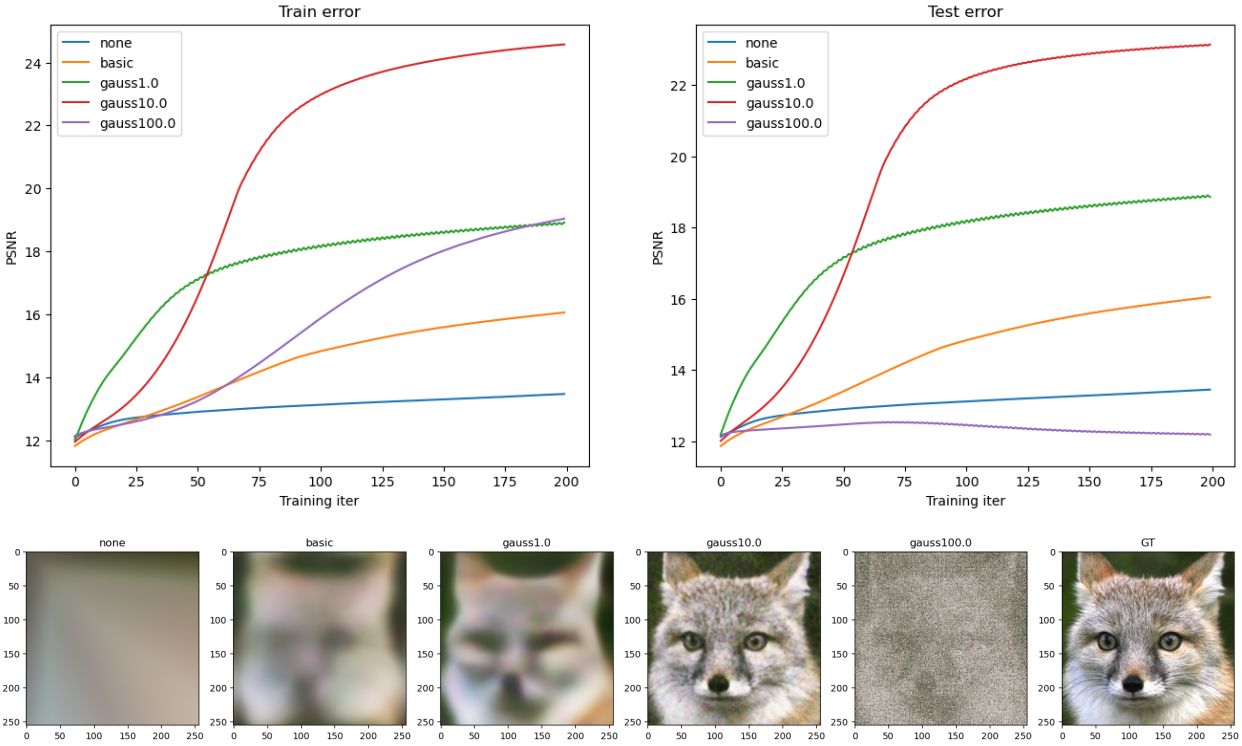






Part 4: High resolution (image of your choice)

(For this part, you can select an image of your choosing and show the performance of your model with the best hyperparameter settings and mapping functions from Part 3. You do not need to show results for all of the mapping functions.)



Part 5: Discussion

Briefly describe the hyperparameter settings you tried and any interesting implementation choices you made.

Firstly, for the model architecture, I started from the default parameters given in the notebook. Since the input image has a dimension of 256 * 256 * 3, the default parameters will take a lot of time to train. Therefore, I decided to modify the architecture to reduce the number of model parameters. I have tried different hidden layer sizes, including 256, 128, 64 and 32. A larger hidden layer size doesn't necessarily boost the performance, but it will increase the training time. I decided to use deeper networks with less hidden neurons per layer. The optimizer is Adam, and the learning rate is tuned, based on the PSNR vs epoch plot, such that the peak signal to noise ratio does not fluctuate from epoch to epoch.

The final parameters are:
Number of layers = 5
Hidden size = 64
Number of epochs = 200
Learning rate = 5e-4

How did the performance of SGD and Adam compare?
Given the same number of epochs and learning rate, it looks like Adam converges faster than SGD. So one way to improve SGD performance is either to increase the number of epochs, or to use a slightly larger learning rate. Therefore, I decided to use Adam optimizer for high resolution reconstruction tasks to save time.

How did the different choices for coordinate mappings functions compare?
On low resolution images, Fourier feature mapping with a std of 1.0 usually performs better than other mappings. On high resolution images, fourier feature mapping with a std of 10.0 usually performs better than other mappings.

What insights did you gain from your own image example (Part 4)?
If we use fourier feature mapping for a high resolution reconstruction task, it is better to use deeper networks than large and shallow networks. Fourier feature mapping with a std of 100 will generate a high variance distribution, and using a multi layer perceptron on this distribution will produce heavy noises in the resultant image.

References
https://towardsdatascience.com/derivative-of-the-softmax-function-and-the-categorical-cross-entropy-loss-ffceefc081d1

https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/

https://towardsdatascience.com/how-to-implement-an-adam-optimizer-from-scratch-76e7b217f1cc

https://bmild.github.io/fourfeat/