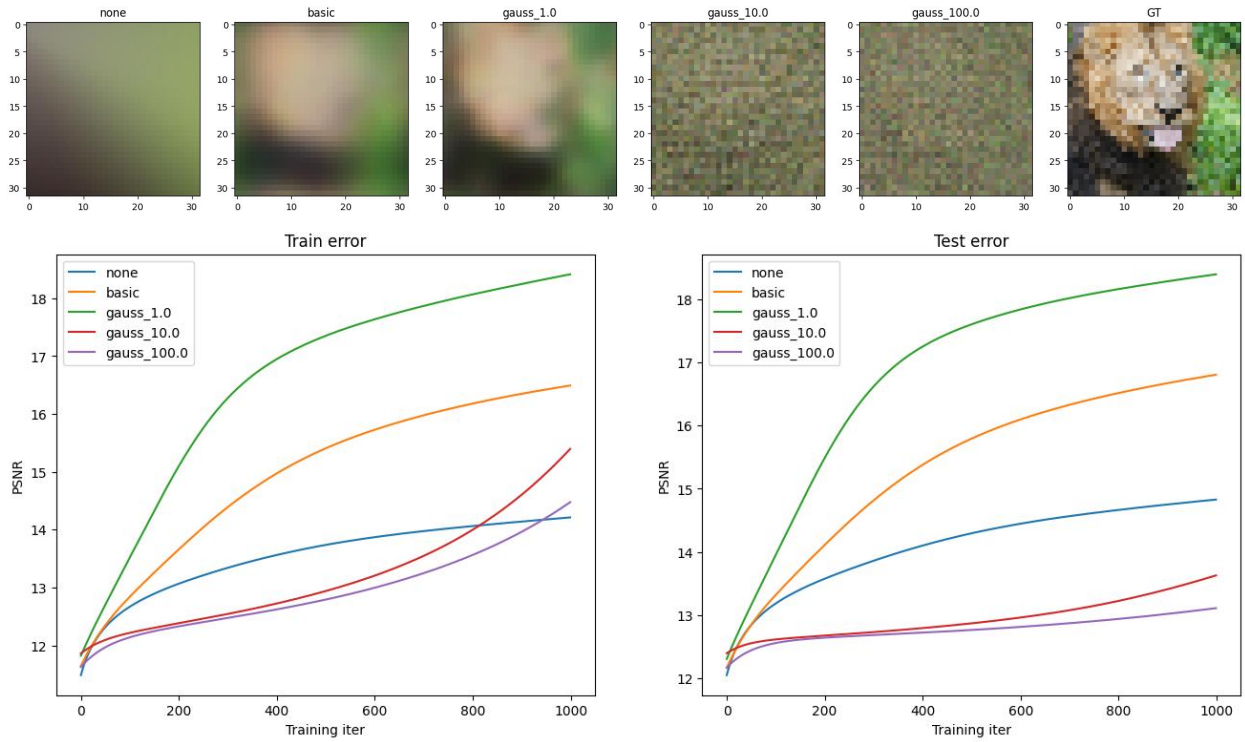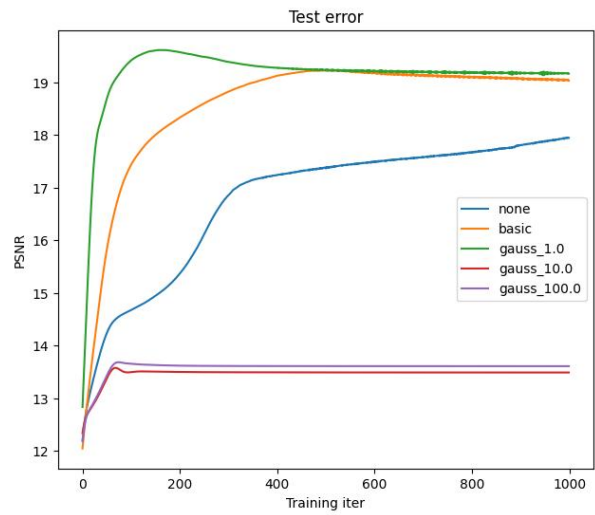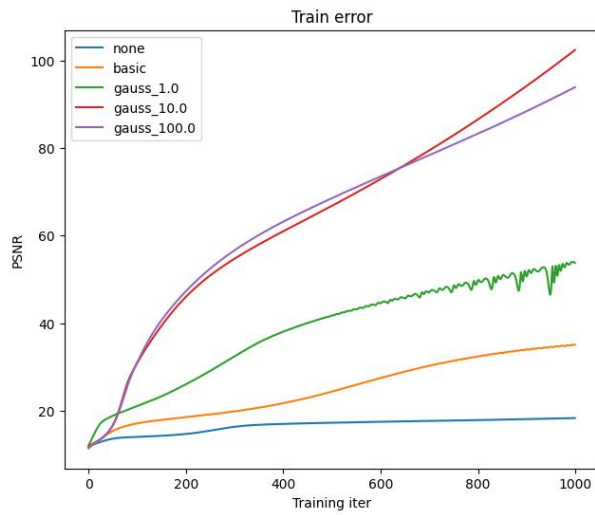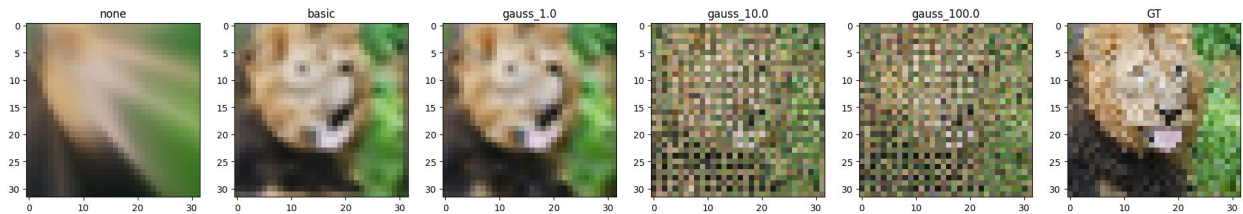# Assignment 2

## Name(s): Ziye Tang
## NetID(s): ziyet3

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
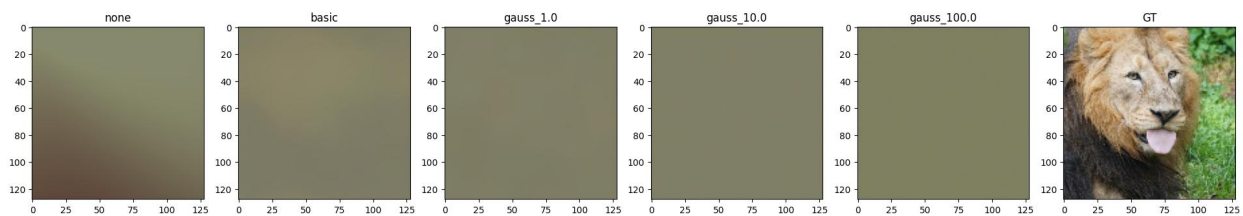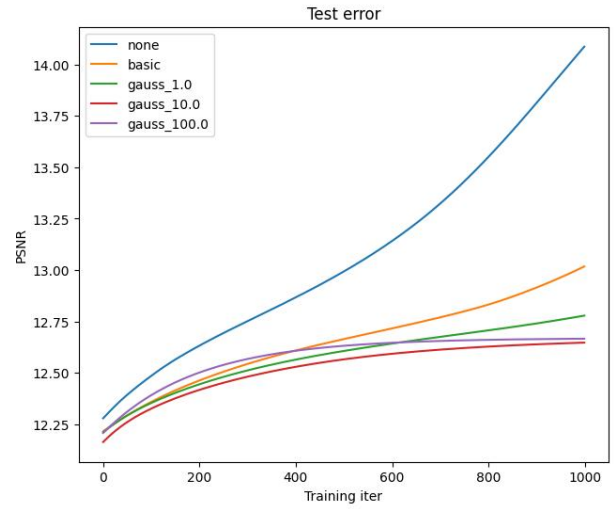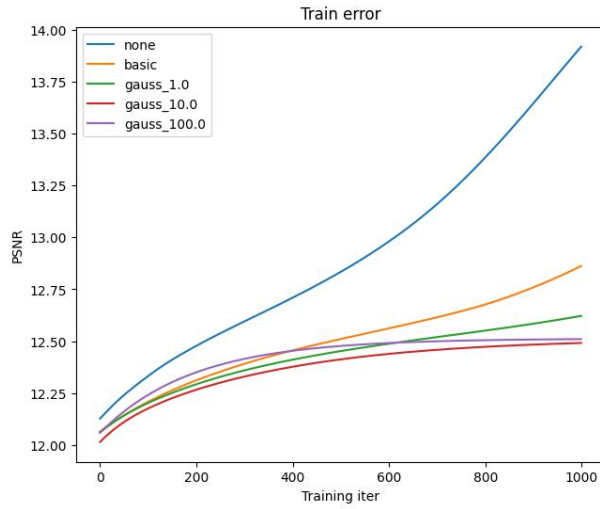
## Part 1: Low resolution example - SGD

# Part 2: Low resolution example - Adam





# Part 3: High resolution example

## SGD:

**Adam:**
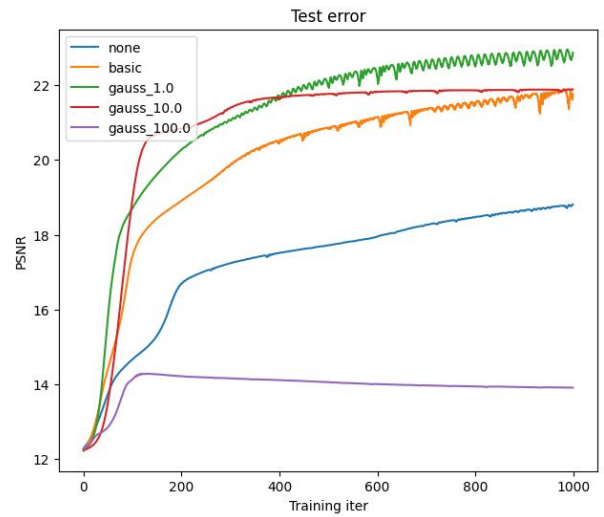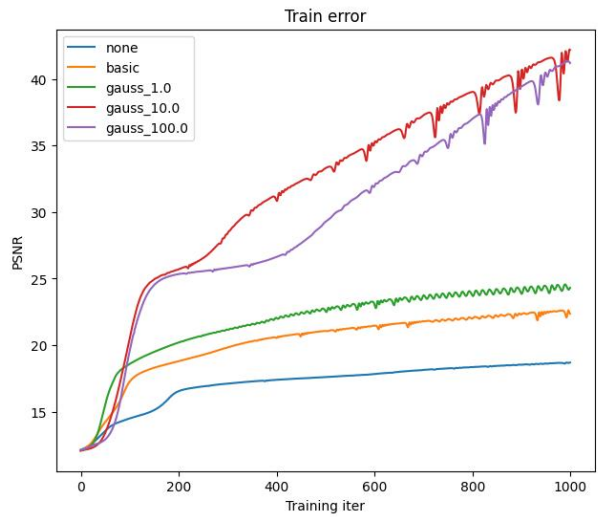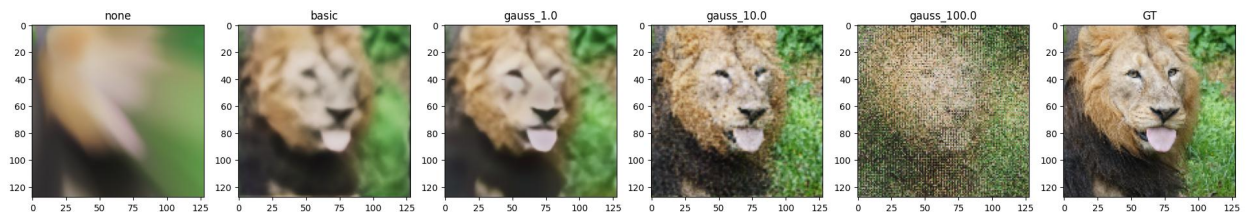
# Part 4: High resolution (image of your choice)

*(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.)*



reconstruction      ground truth



iter 0   iter 125   iter 250   iter 375   iter 500   iter 625   iter 750   GT



MSE history



PSNR history
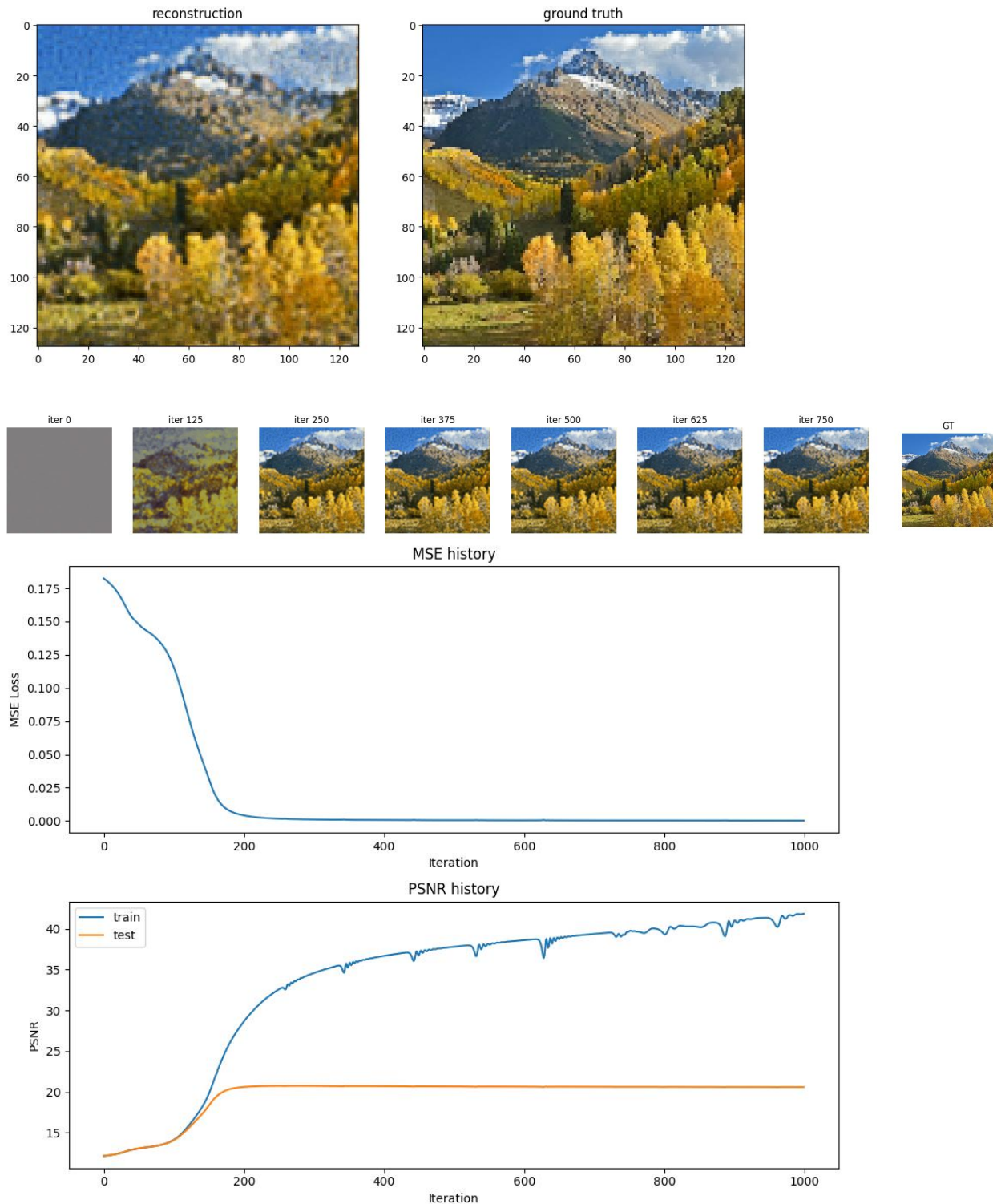
**Part 5: Discussion**

*Briefly describe the hyperparameter settings you tried and any interesting implementation choices you made.*
I mostly used defualt setting but set learning rate to be 1e-2 for SGD because I found it converges too slowly. I also tried to set the number of layers to be 10 and the hidden layer size to be 512, but it didn't seem to work very well for both Adam and SGD. For Adam, I think the defualt setting is already good enough.

*How did the performance of SGD and Adam compare?*
Adam is obviously better than SGD since it reconstruct most features of the original images. Images reconstructed by SGD is dim and not recognizable.

*How did the different choices for coordinate mappings functions compare?*
When reconstructing low-resolution images, gauss_1.0 might be the best choice. But when it comes to higher resolution, gauss_10.0 is better. And the image reconstructed by using gauss_100.0 is also much more clear when trainning with high-resolution image. I can infer that if we use images whose resolution is even higher than 128*128, then gauss_100.0 might be the best choice.

*What insights did you gain from your own image example (Part 4)?*
I found that my neural network converges very quickly when trainning my image. The reason might be that similar colors stay in the same area, and different colors separate clearly from each other. Also, my image doesn't have too much detail, like the facial features of lions. So, it would be easier for my neural network to reconstruct a very good image.