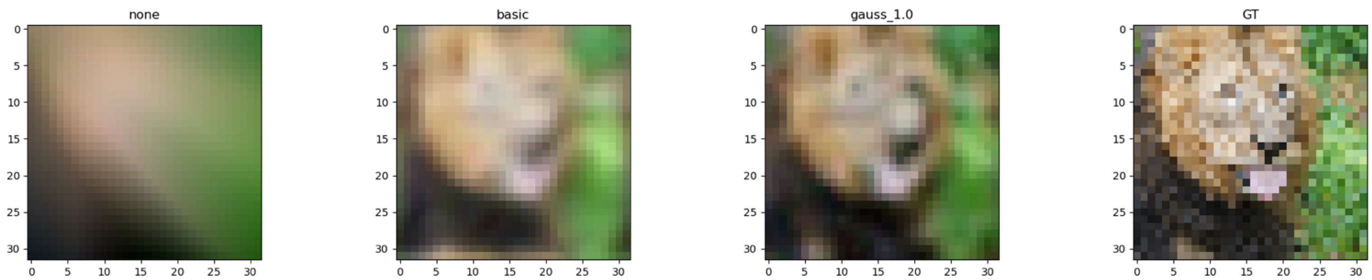
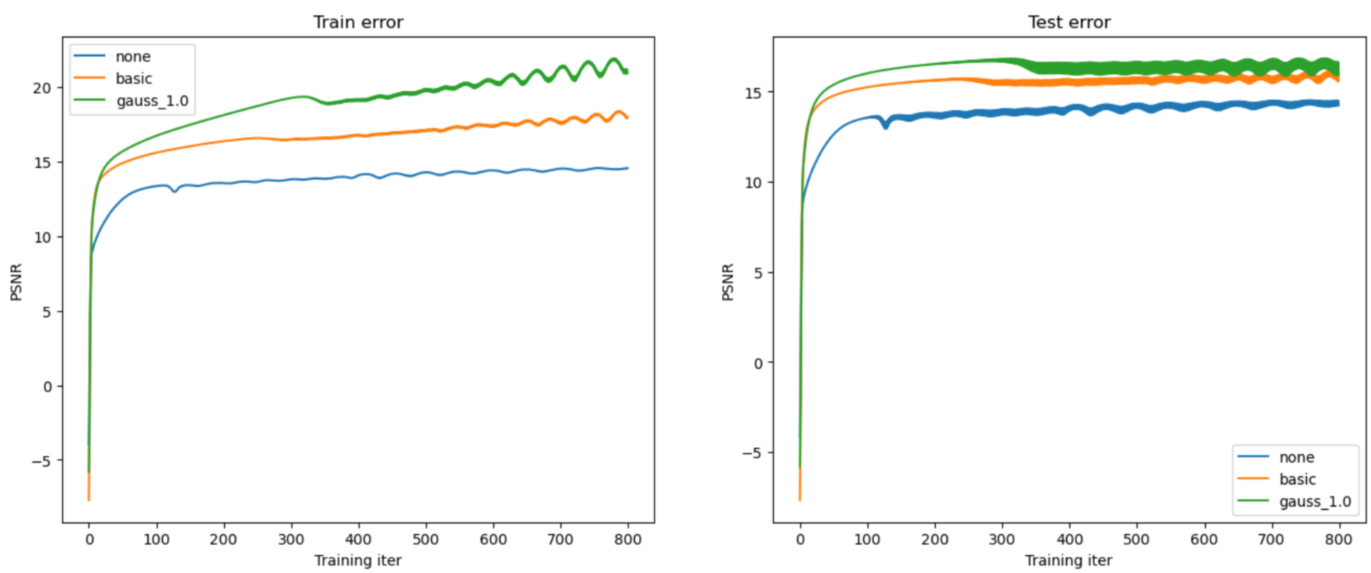
Assignment 2 Report

Name: Ruichao Chen

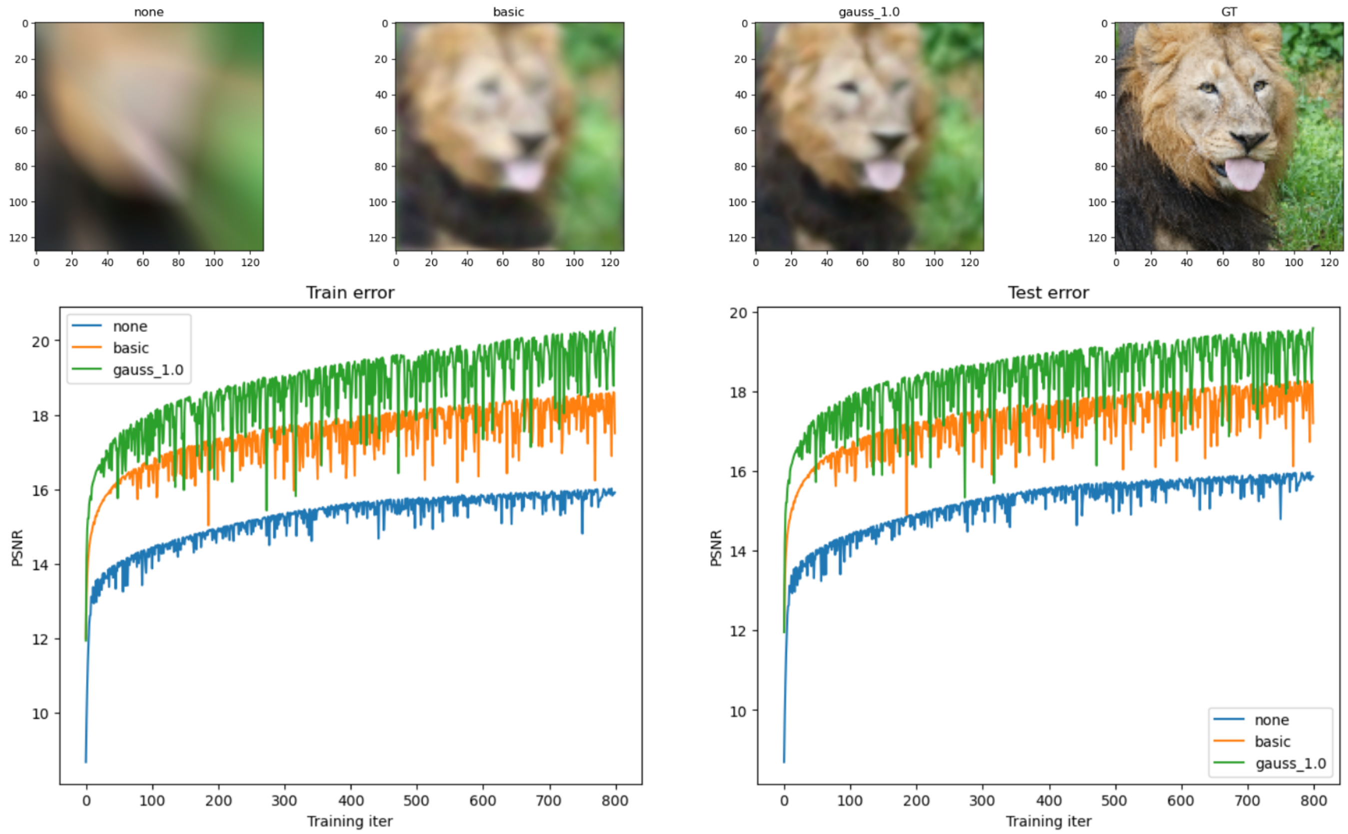
NetID: rc38

Part1: Low resolution example:

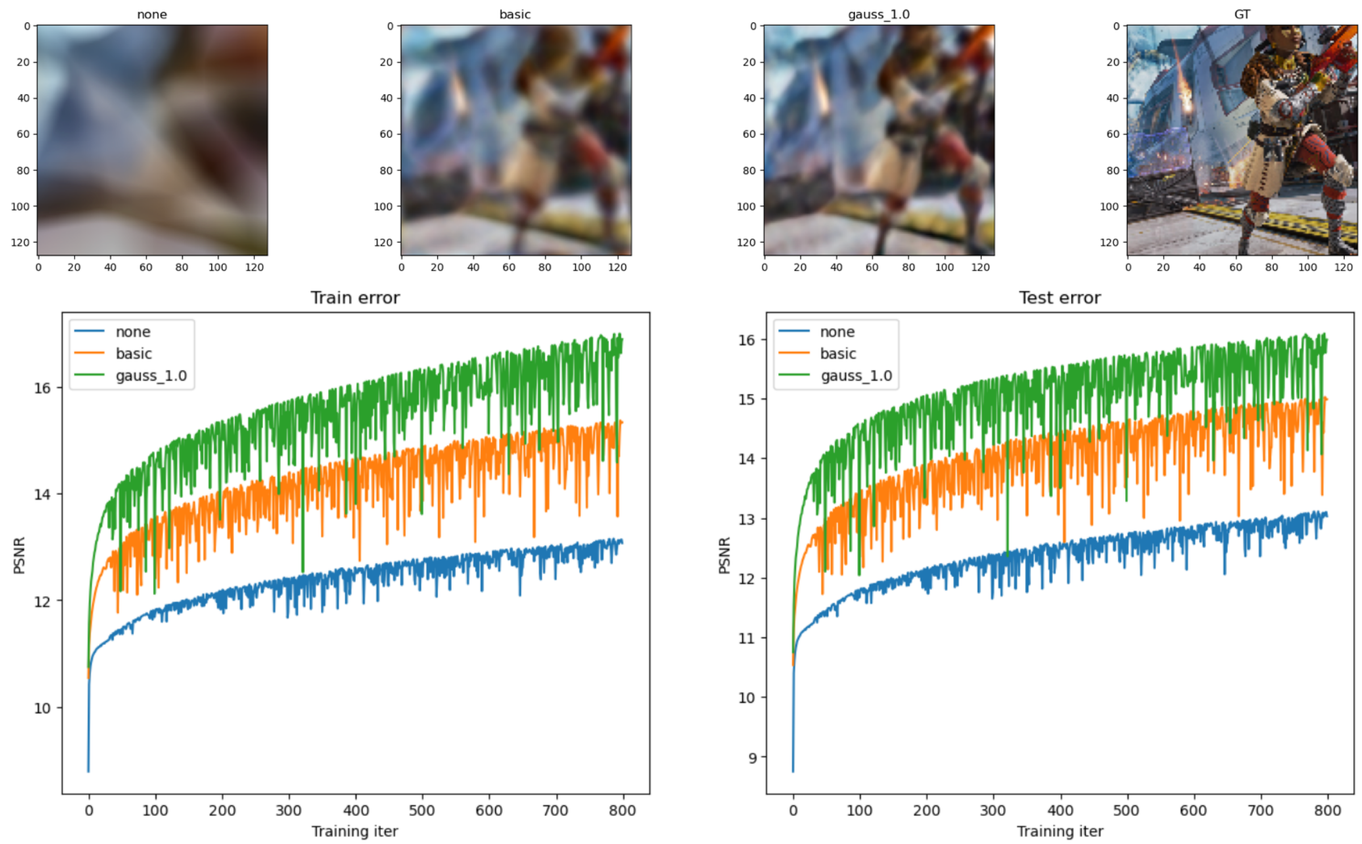




Part2: High resolution example:



Part3: High resolution (image of your choice):



Part4: Discussion:

1.During this experiment, I tried various hyperparameter combinations before settling on the final configuration hidden_size=256, epochs=800, learning_rate=0.1, and batch_size=512. The adjustment process included:

For the learning rate, I initially tried the suggested 0.1, but I encountered instability and gradient explosion issues with high-resolution images. I gradually reduced it to 0.06, which improved stability while maintaining reasonably fast convergence. And for the hidden layer size, I started with 128 and increased to 256. Considered 512 but found that computational costs increased significantly with limited performance gains. The size of 256 provided a good balance between performance and training time.

For the training epochs, the original recommendation was 1000 epochs, but I observed convergence stabilizing after 800 epochs, with minimal improvements from further training, so I chose 800 as a compromise. Although the plot had an upward trend when I train 800 epochs, the velocity was so slow that it had to take me over half an hour to get the final results if I set the epochs over 2000. So I finally gave up to increase the epochs due to the long process. I also added the mini batch, and I tested both 256 and 512, finding that the larger batch size provided better training efficiency on my hardware while maintaining good generalization capabilities.

2. For the none, it is the lowest curve with PSNR around 16, resulting in noticeably blurry reconstructions lacking details and sharp edges. The basic has medium performance with PSNR around 18, capable of recovering more details but still performing poorly in complex texture areas. The gauss1.0 has the best performance with PSNR exceeding 20, able to reconstruct sharper edges and finer details. Particularly impressive in reproducing the lion's facial features and fur textures. These differences highlight the critical impact of input feature representation on coordinate network performance, with Fourier feature mapping significantly enhancing the network's ability to represent high-frequency details through input space transformation.

3. I observed from the train and test plots that all methods converge most rapidly in the first 100 iterations, followed by a slower improvement phase. I also found the volatility of the curves, particularly the gauss_1.0

curve exhibits notable up-and-down fluctuations in later stages. This may be due to:

- Gradient noise from random batch sampling
- Relatively high learning rate (0.1) causing optimization to oscillate around local minima
- Natural instability in capturing high-frequency features

4. For my own image example, I chose an image with more complex color distributions and more lines. And that can be helpful for me to observe the differences. With a complex image, the performance differences between the three methods become more pronounced. The advantages of Fourier feature mapping in capturing high-frequency details (such as sharp edges and fine textures) become more prominent. The no-encoding method performs adequately in areas with gradual color changes but completely fails in areas with abrupt color changes (like sharp edges); the Fourier method can more accurately preserve these transitions. These observations further confirm the effectiveness of Fourier feature mapping for high-detail image reconstruction, emphasizing the importance of choosing the right input representation in neural implicit representations.