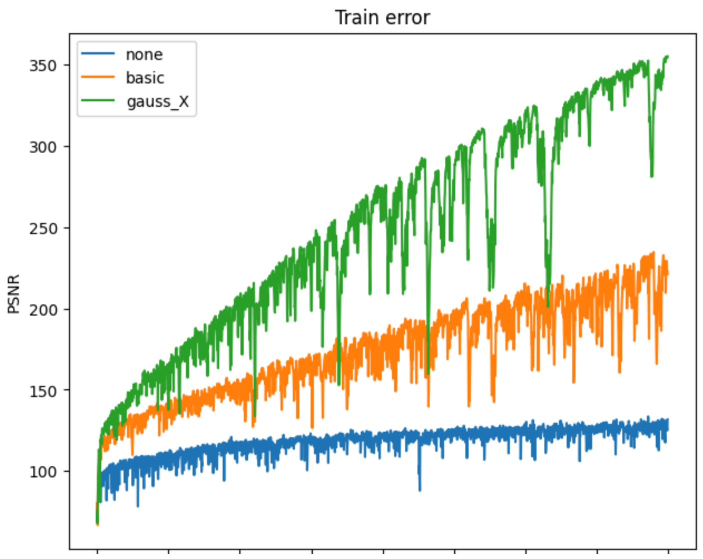
Assignment 2 Report

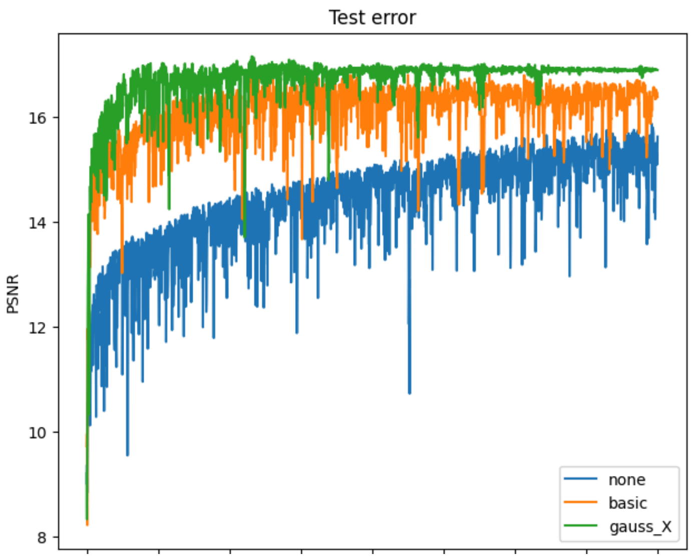
Name(s):Carlos Paredes
NetID(s):cfp3

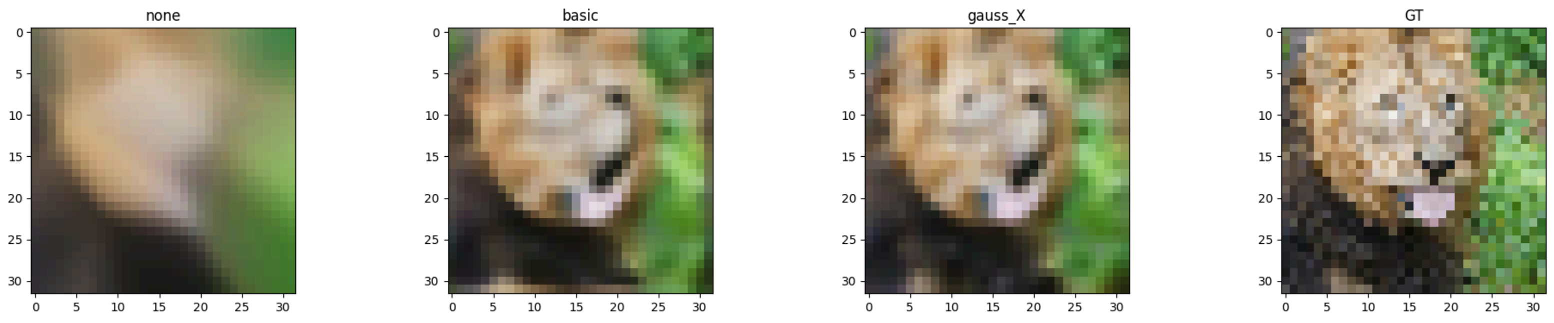
In each the following parts, you should insert the following:

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

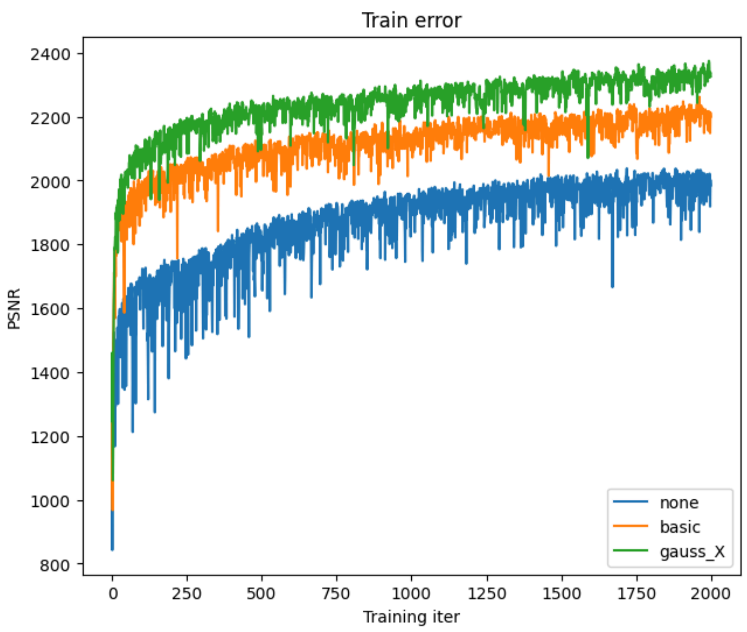
Part 1: Low resolution example

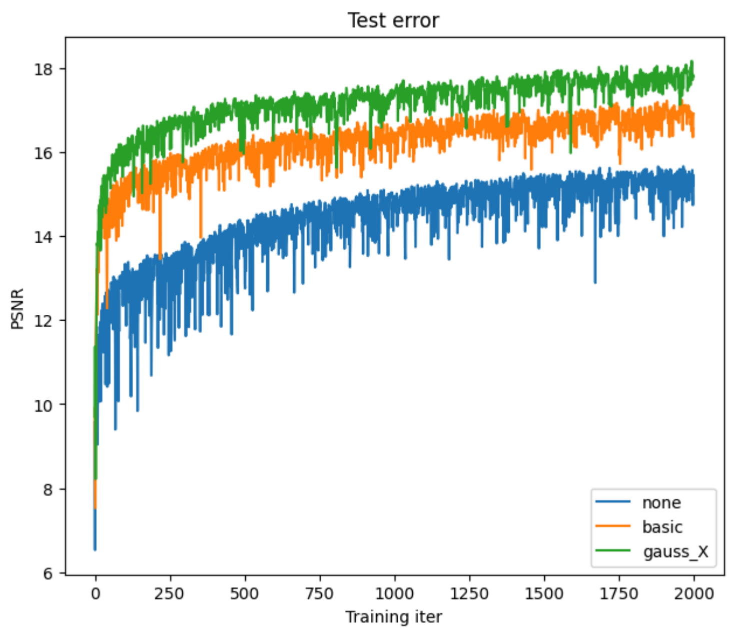


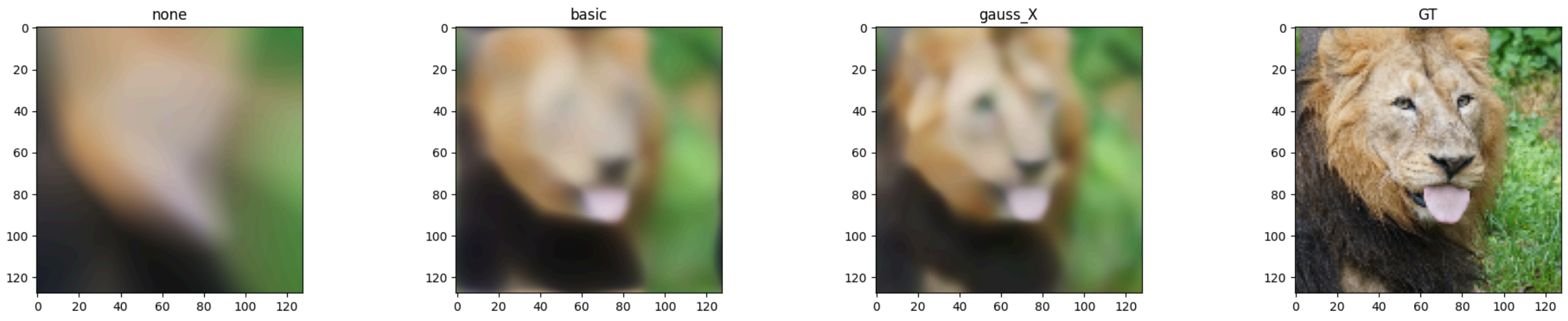




Part 2: High resolution example



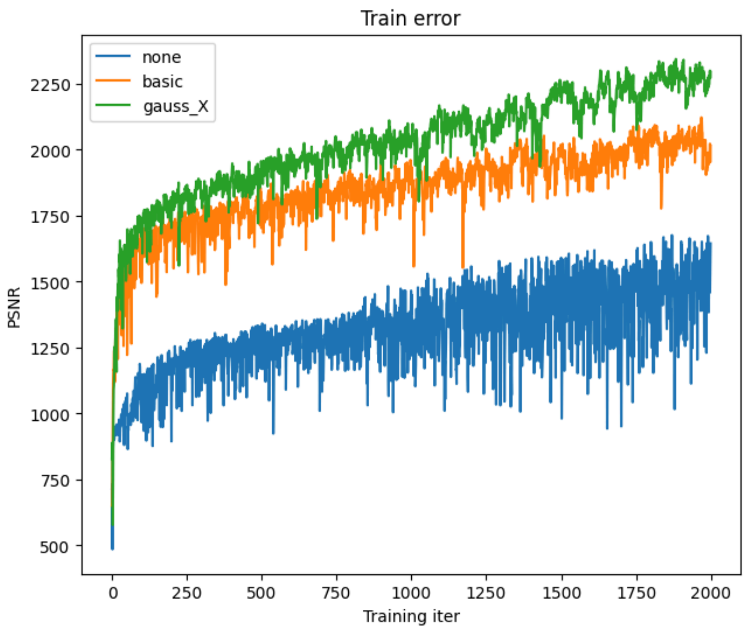


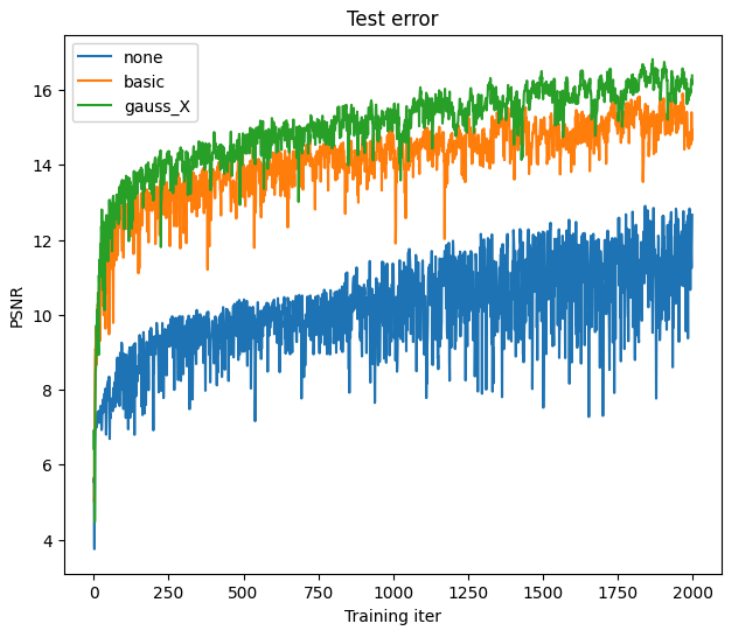


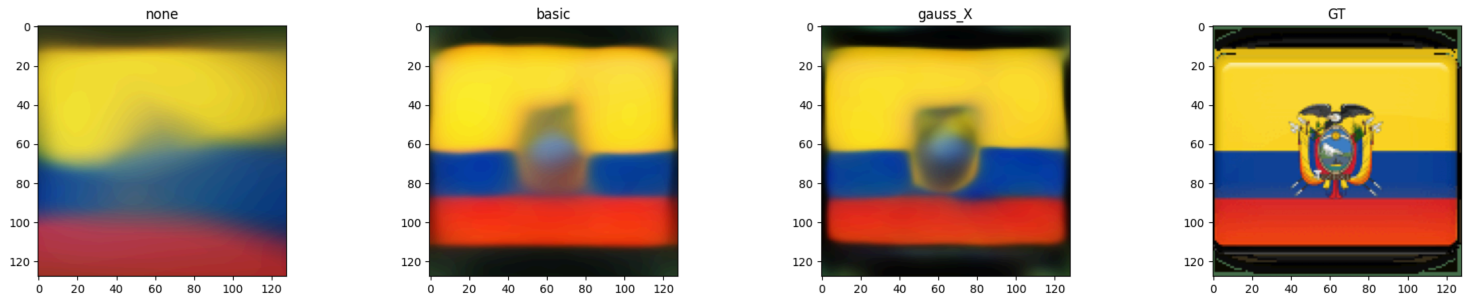
Part 3: High resolution (image of your choice)

(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 2. You do not need to show results for all of the mapping functions.)

Part 4: Discussion







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

So my hyperparameter assignments are: hidden_size = 256,epochs = 2000,learning_rate = 0.2 output_size = 3. Not much changed other than slightly increasing the learning rate, and doubling the epochs. The main reason was to let the high resolution examples train for longer and also gain more information each iteration. One thing that I did was implement batch training. While it didn't make much difference in accuracy, it did reduce the amount of resources the code used during training. One thing in the back propagation I experienced was that dividing gradients by y.shape[0]*y.shape[1] was not the correct value. I found that the best value was y.shape[0]/y.shape[1]. Reduced my relative error max from about 8.0e-1 down to about 1.758780e-08.

How did the different choices for coordinate mappings functions compare?

The most obvious difference is that the quality of the reconstruction worst to best is none, basic, Fourier Feature Encoding. This can be attributed to the input size increasing for each mapping. This allowed more information to be learned from each datapoint.

Do you make any interesting observations from the train and test plots?

In the plots I noticed that the variance of the plots decreases as the mappings improve. The variance is also affected by the resolution. The high resolution had much less variance than the low resolution, which makes sense since the high resolution image has more data points reducing the noise.

What insights did you gain from your own image example (Part 3)?

My own image has a lot of noise, this may be due to compression from the original link. Furthermore, the least accurate values are in the region with the most detail, meaning that those areas didn't learn as much so increasing the number of hidden layers, or increasing the learning rate may allow for more precise predictions