Assignment 2

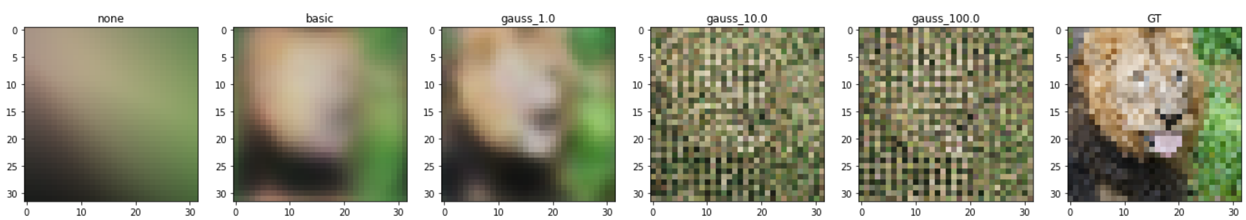
Name(s): Neha Jain, Pallaw Kumar
NetID(s): nehaj4, pallawk2

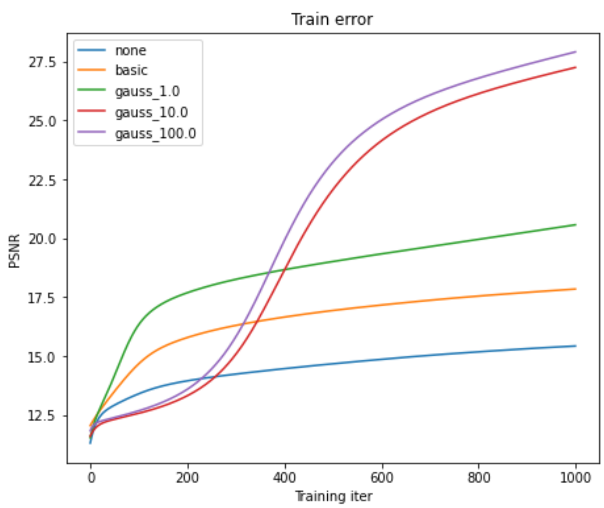
In each of the following parts, you should insert the following:

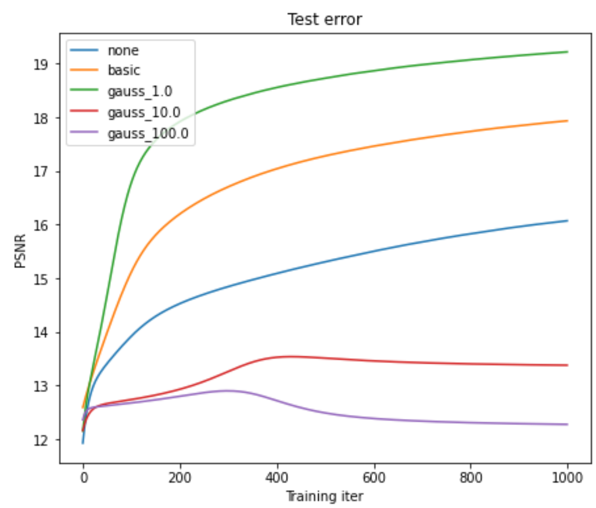
- Train/test loss plots
- Qualitative outputs for GT, No encoding, Basic Positional Encoding, and Fourier Feature Encoding at three different scales

Part 1: Low-resolution example - SGD

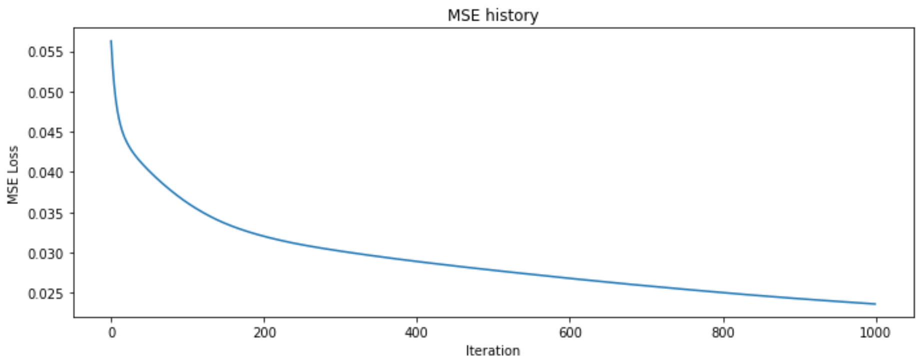
Number of layers	Hidden Size	Epochs	Learning Rate	Optimizer	Computation time
5	256	1000	1e-1	SGD	~ 6 mins for all mappings





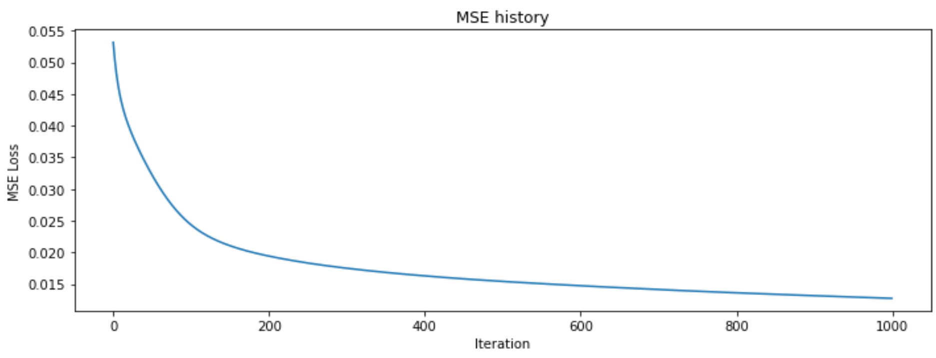


MSE plots:
None
MSE Test: 0.0227



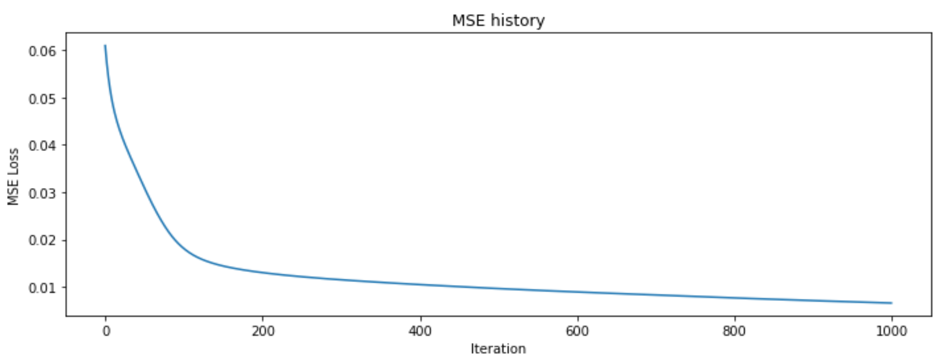
Basic

MSE Test : 0.0143



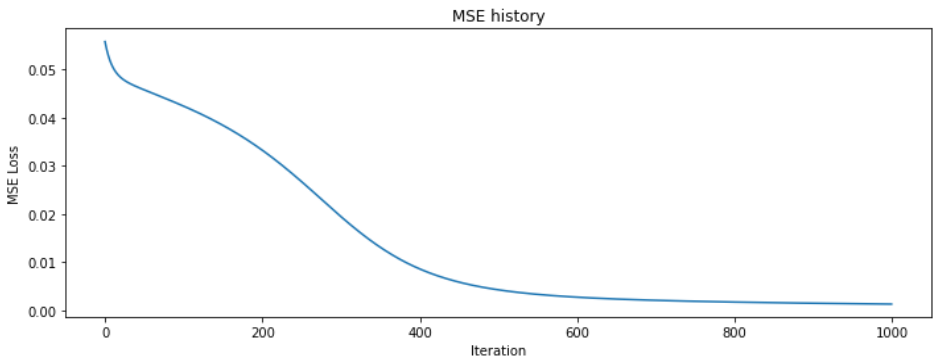
Gauss 1.0

MSE Test : 0.0104



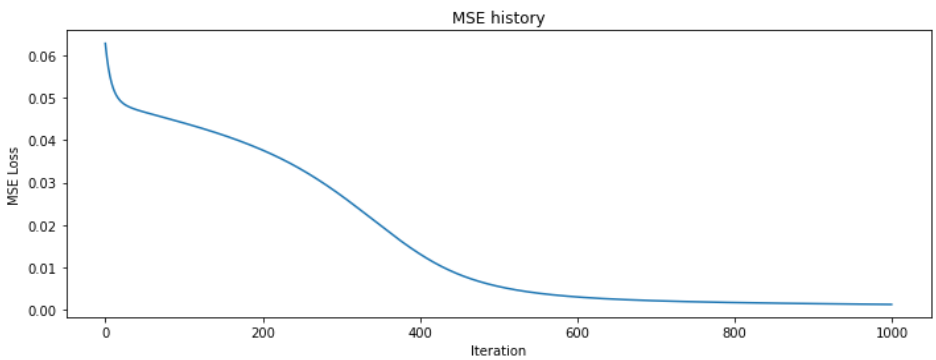
Gauss 10.0

MSE Test: 0.0416



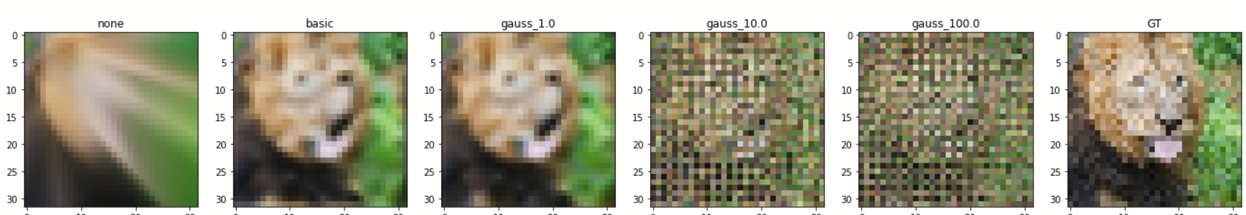
Gauss 100.0

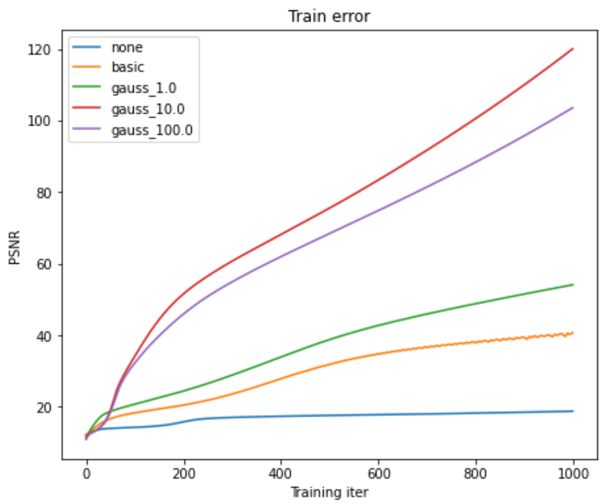
MSE Test: 0.0414

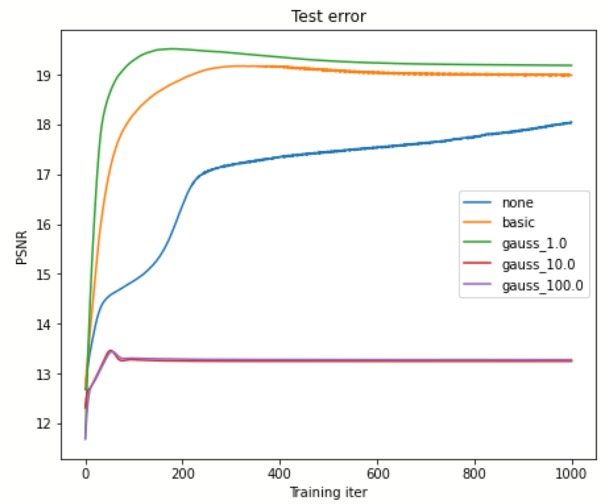


Part 2: Low-resolution example - Adam

Mapping	Number of layers	Hidden Size	Epochs	Learning Rate	Optimizer	Computation time
None	7	256	1000	1e-4	ADAM	~ 1 min
Basic	5	256	1000	1e-4	ADAM	~ 1 min
Gauss 1.0	4	256	1000	1e-4	ADAM	~ 1 min
Gauss 10.0	5	256	1000	1e-4	ADAM	~ 1 min
Gauss 100.0	5	256	1000	1e-4	ADAM	~ 1 min



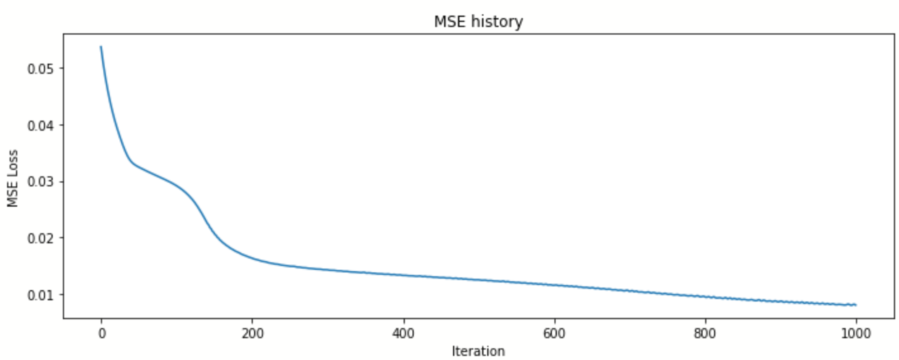




MSE Plots:

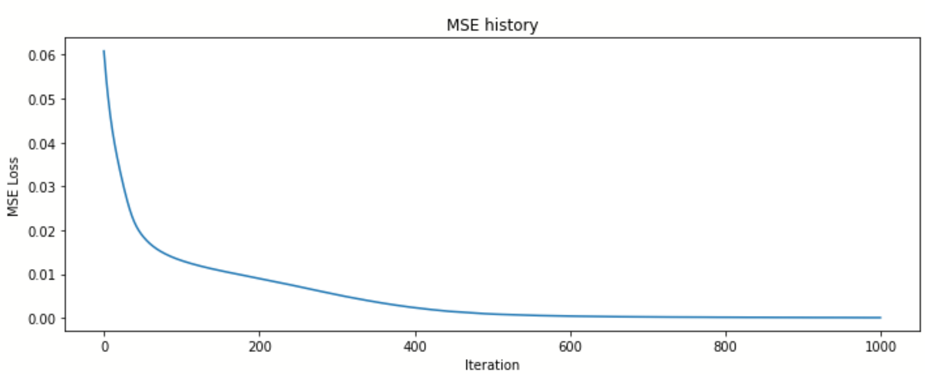
None

MSE Test: 0.0127



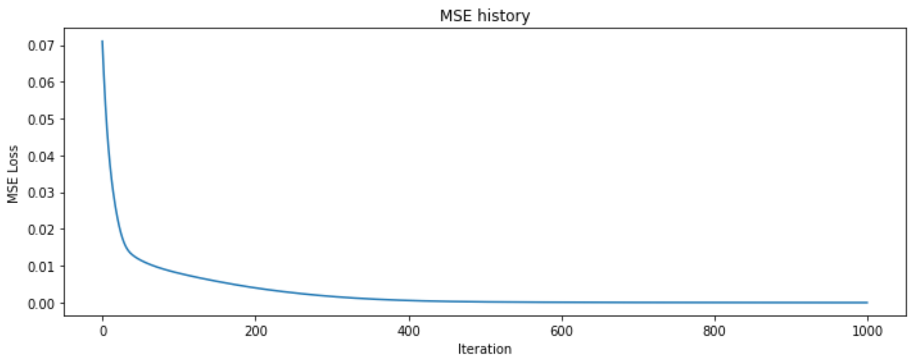
Basic

MSE Test : 0.011

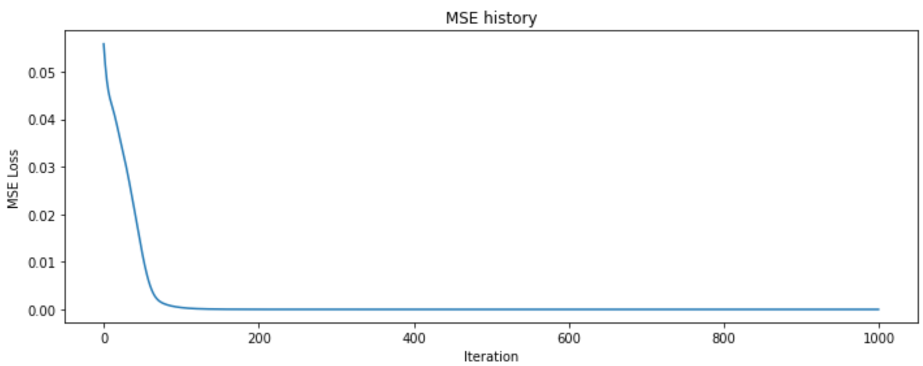


Gauss 1.0

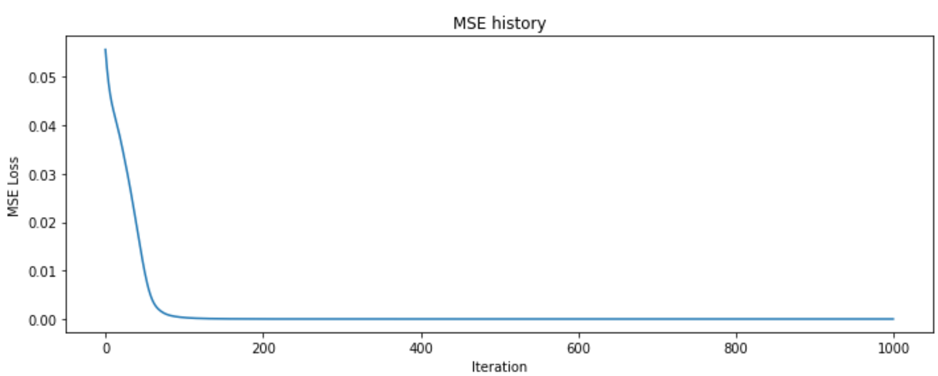
MSE Test: 0.0106



Gauss 10.0
MSE Test: 0.0427

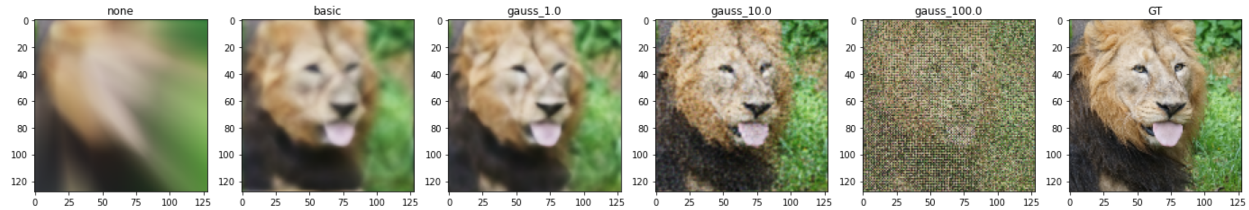


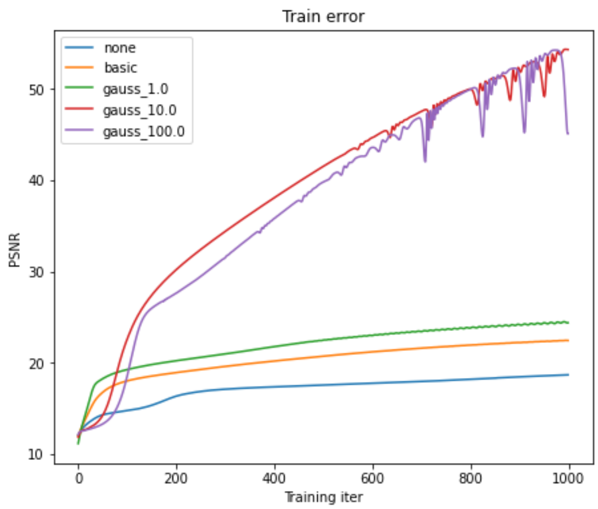
Gauss 100.0
MSE Test: 0.0491

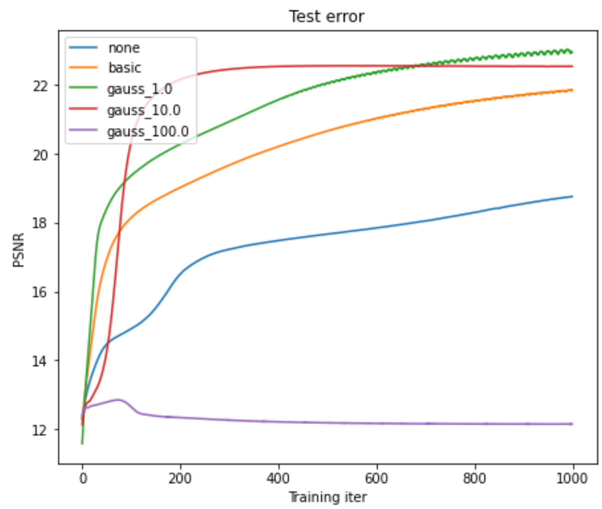


Part 3: High-resolution example

Number of layers	Hidden Size	Epochs	Learning Rate	Optimizer	Computation time
5	256	1000	1e-4	ADAM	~ 1 hr 20 mins for all mappings



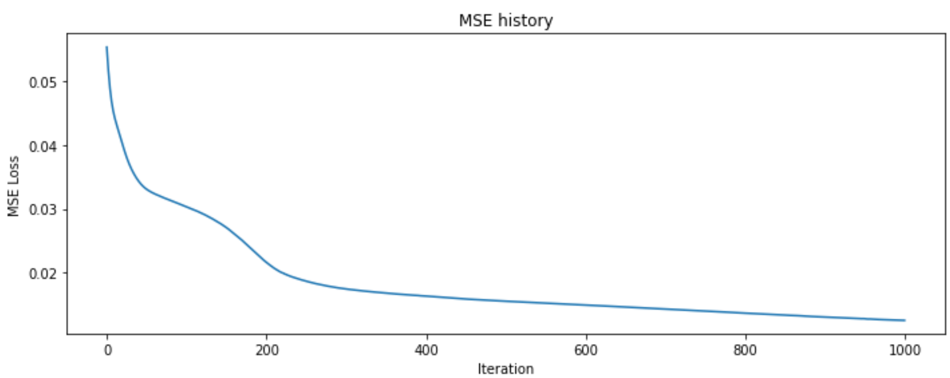




MSE Plots:

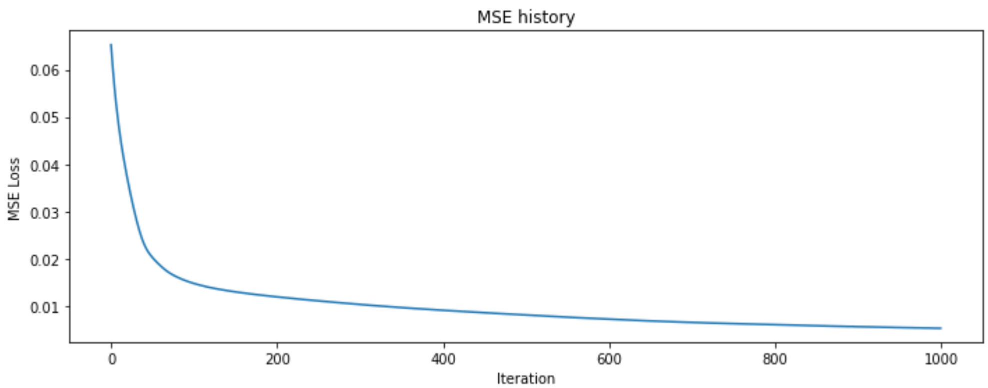
None

MSE Test: 0.0126



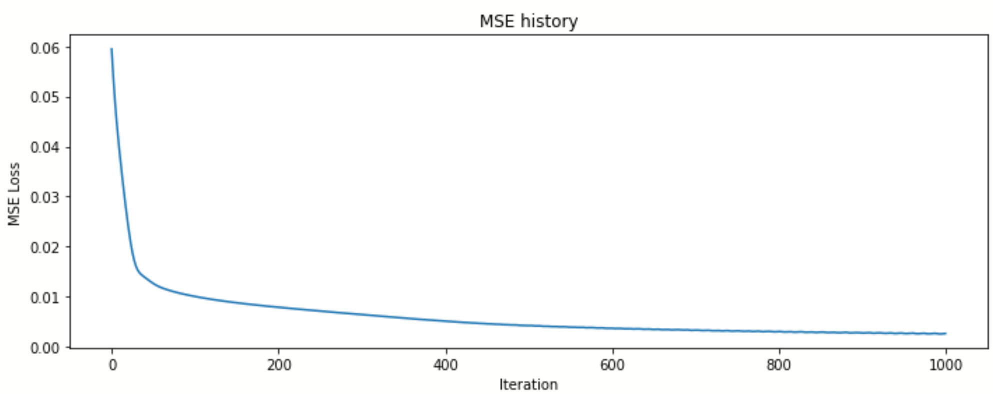
Basic

MSE Test: 0.0061



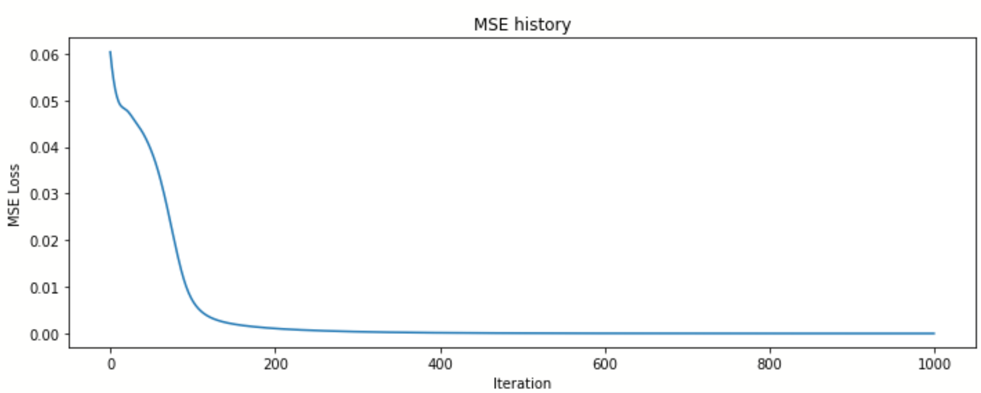
Gauss 1.0

MSE Test: 0.00434



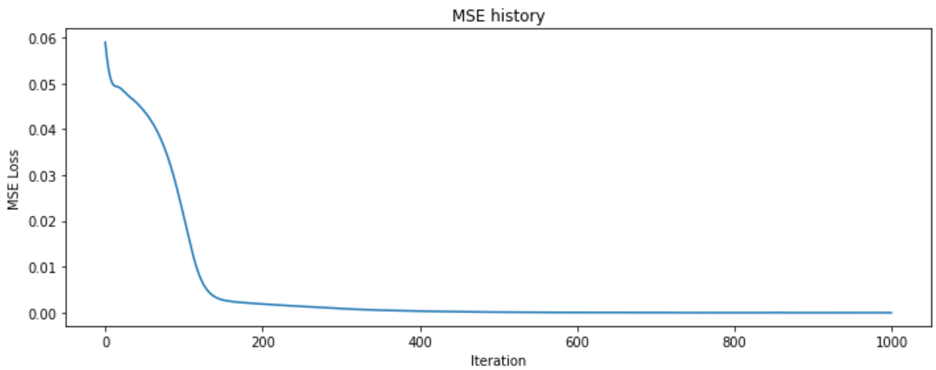
Gauss 10.0

MSE Test: 0.00495



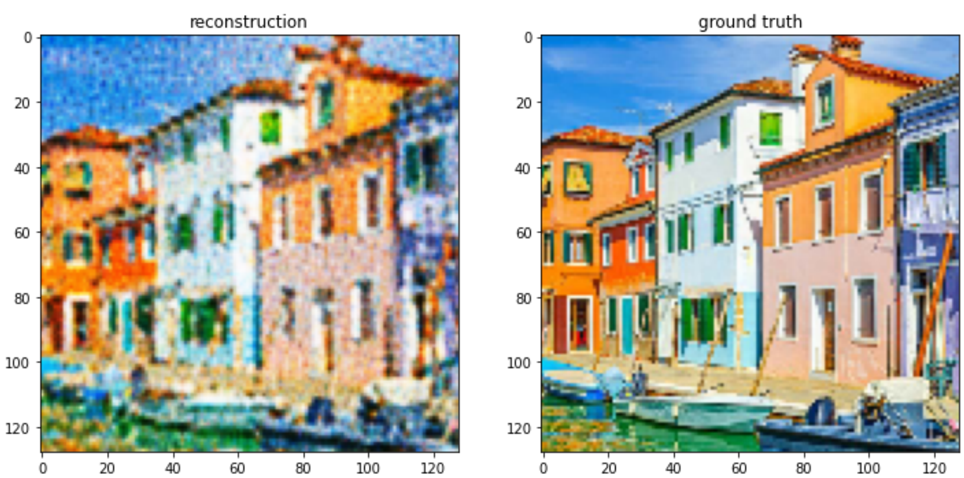
Gauss 100.0

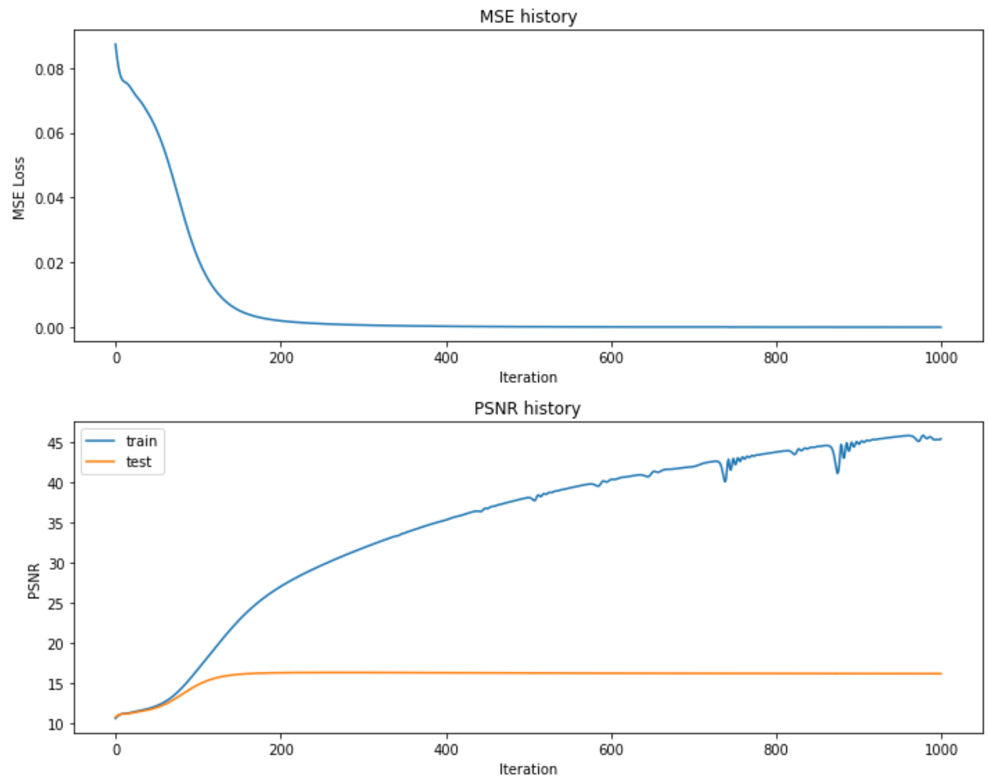
MSE Test: 0.0577



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.)

Number of layers	Hidden Size	Epochs	Learning Rate	Mapping Function	Optimizer
5	256	1000	1e-4	Gauss 10.0	ADAM





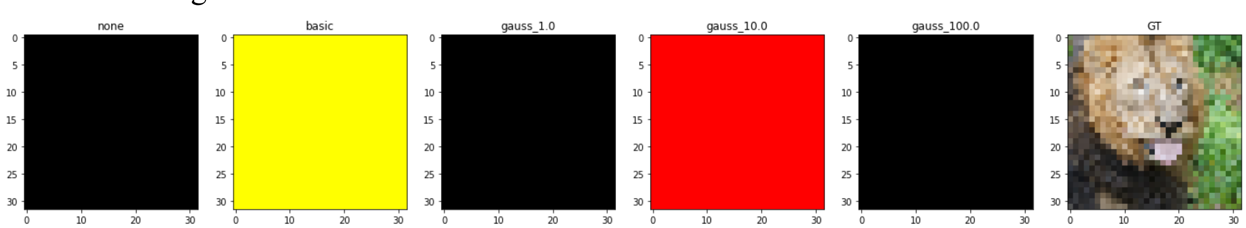
Test MSE 0.0248

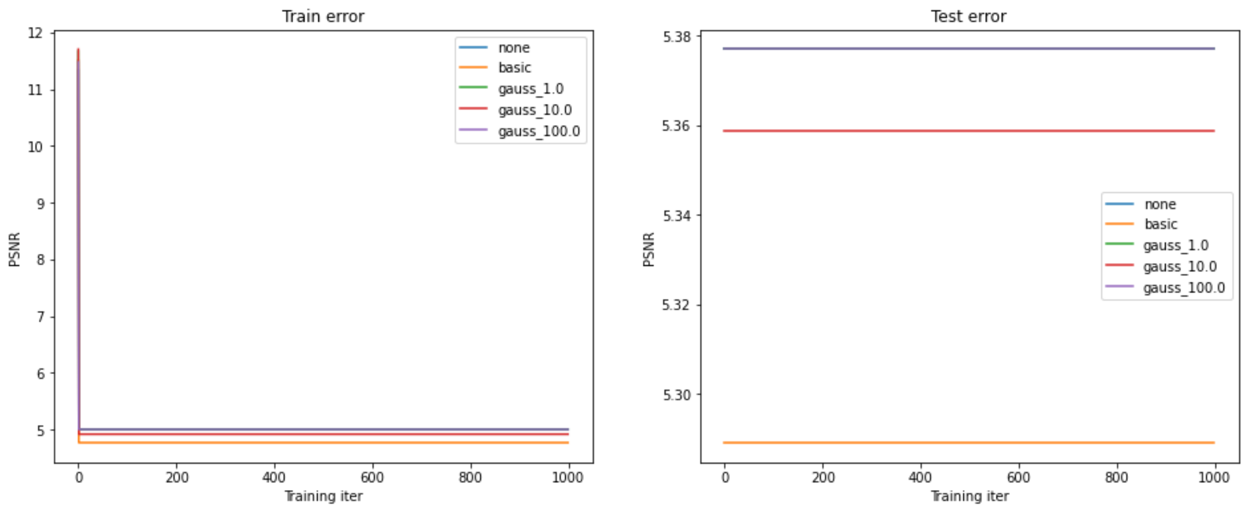
Part 5: Discussion

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

First, we tried decreasing the learning rate from 1e-1 to 1e-5 with a number of layers = 5, hidden_size = 256, epochs = 1000, and "Adam" Optimizer with default settings ($\beta 1 = 0.9$, $\beta 2 = 0.999$, $= 10{-}8$)

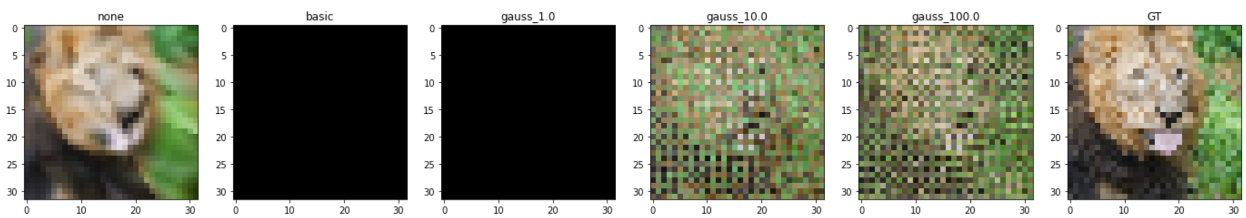
1. Learning rate = 1e-1

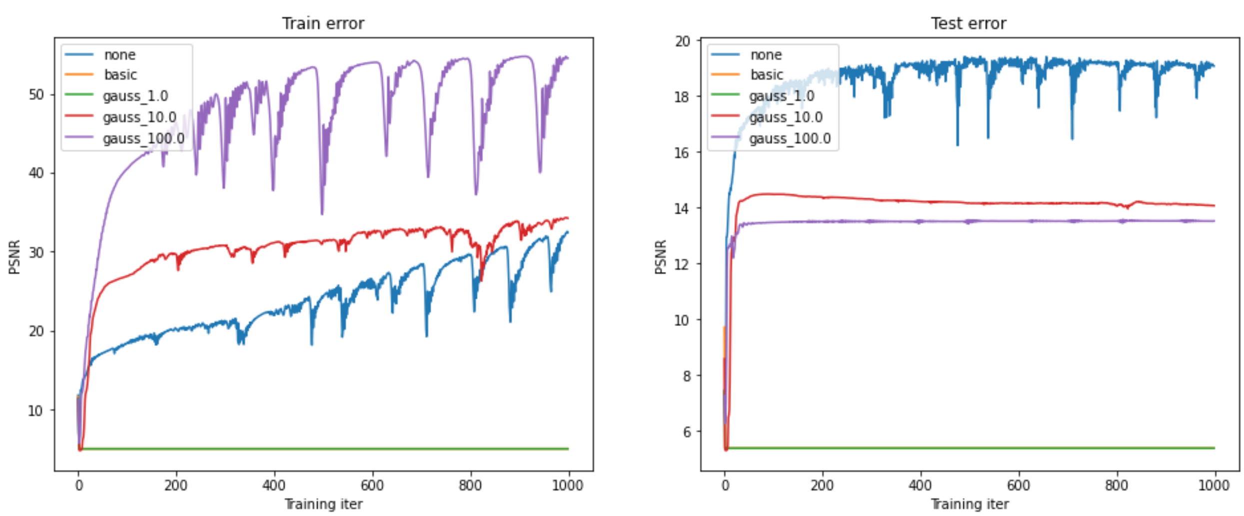




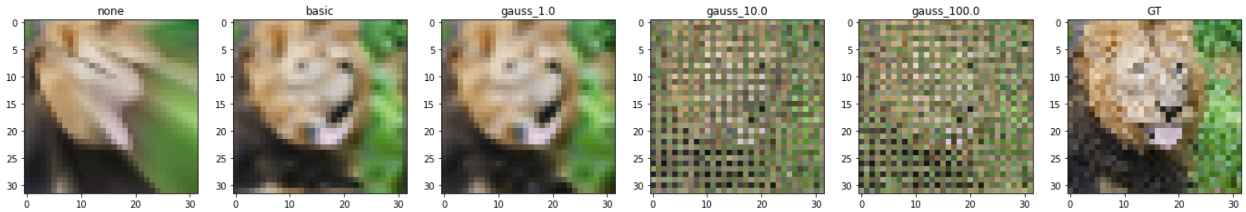
Observation: The learning rate is too high, the model is not learning anything

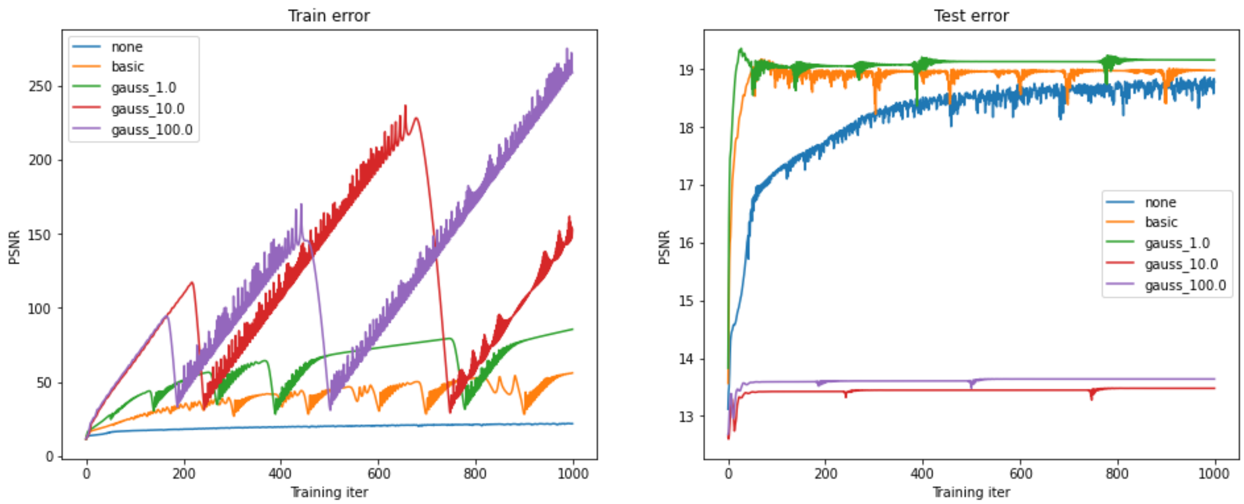
2. Learning rate = 1e-2





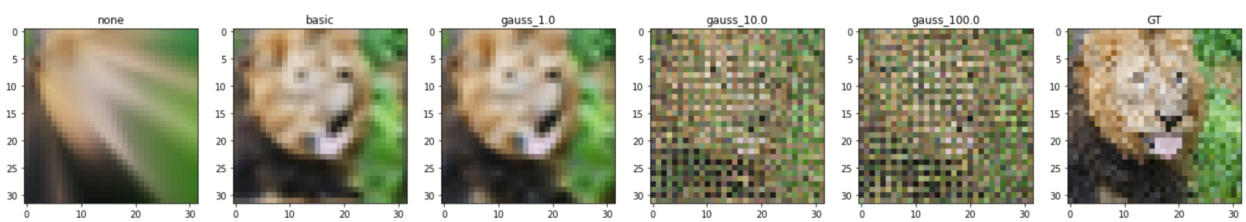
3. Learning rate = 1e-3

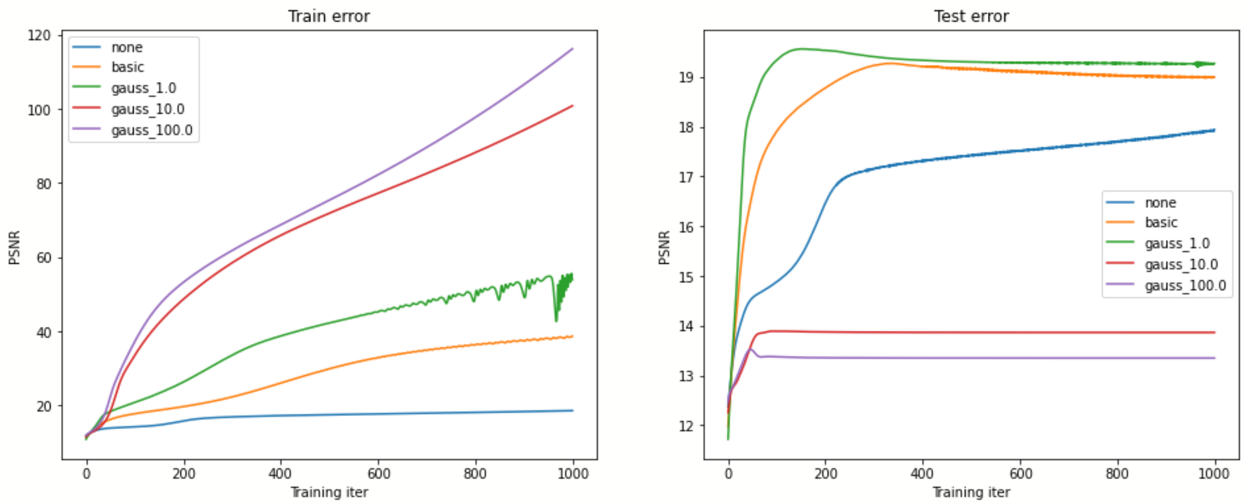




Observation: For learning rates, 1e-2 and 1e-3 learning is quite unstable, the optimizer is making large updates to the parameters that are leading to overshooting the optimal values, causing the loss function to oscillate and prevent convergence.

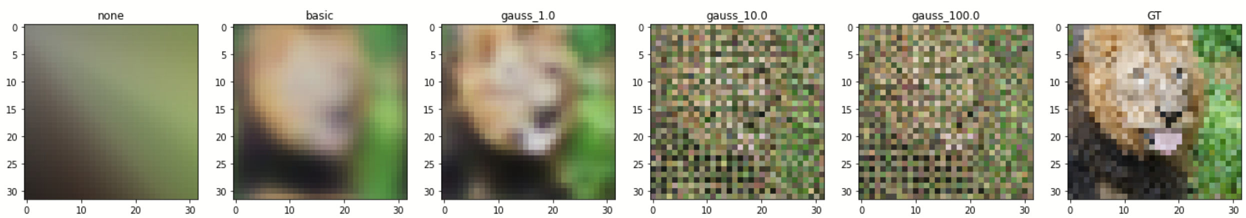
4. Learning Rate = 1e-4

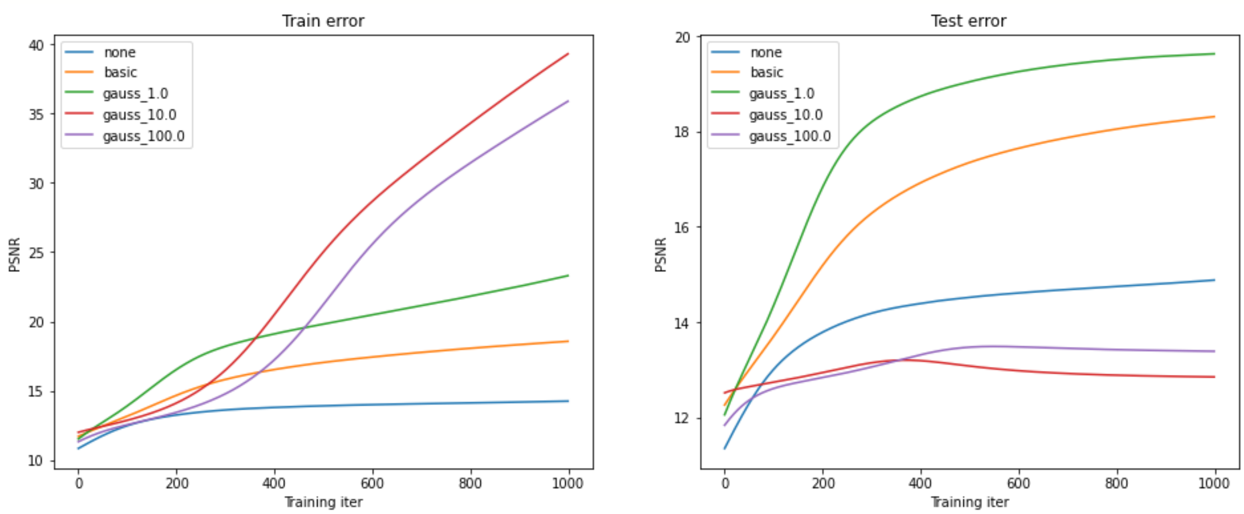




Observation: For learning rates, 1e-4 learning is quite stable and error is converging

5. Learning Rate = 1e-5

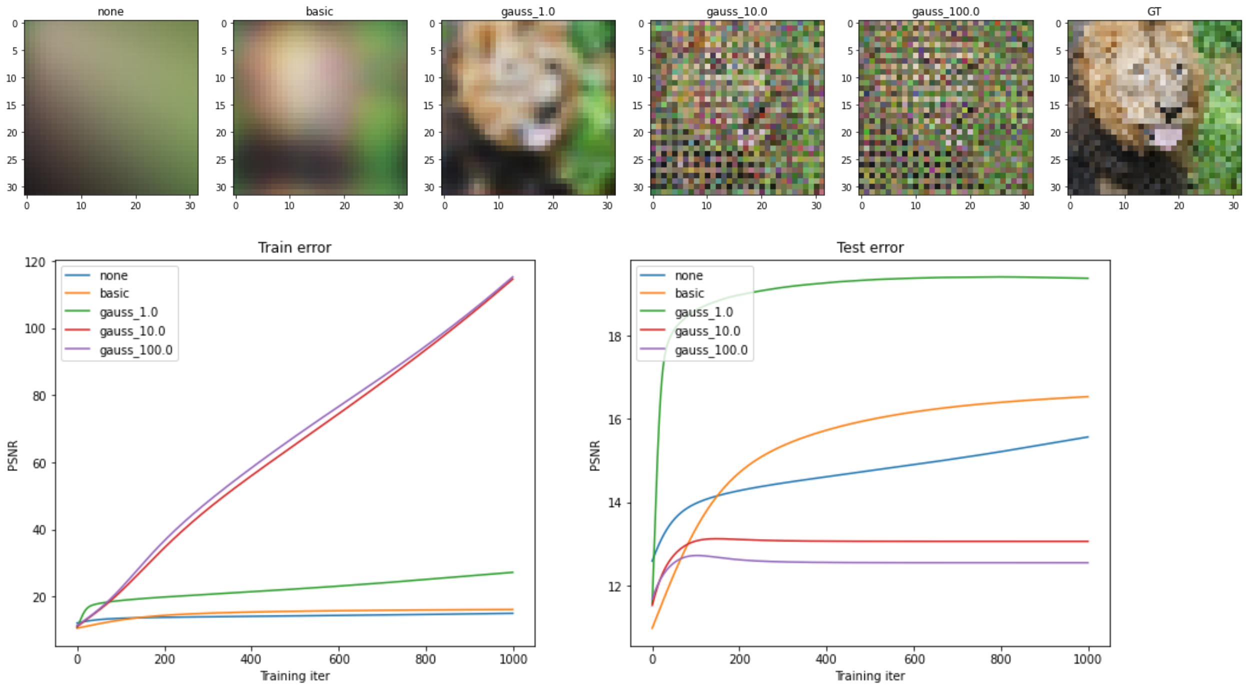




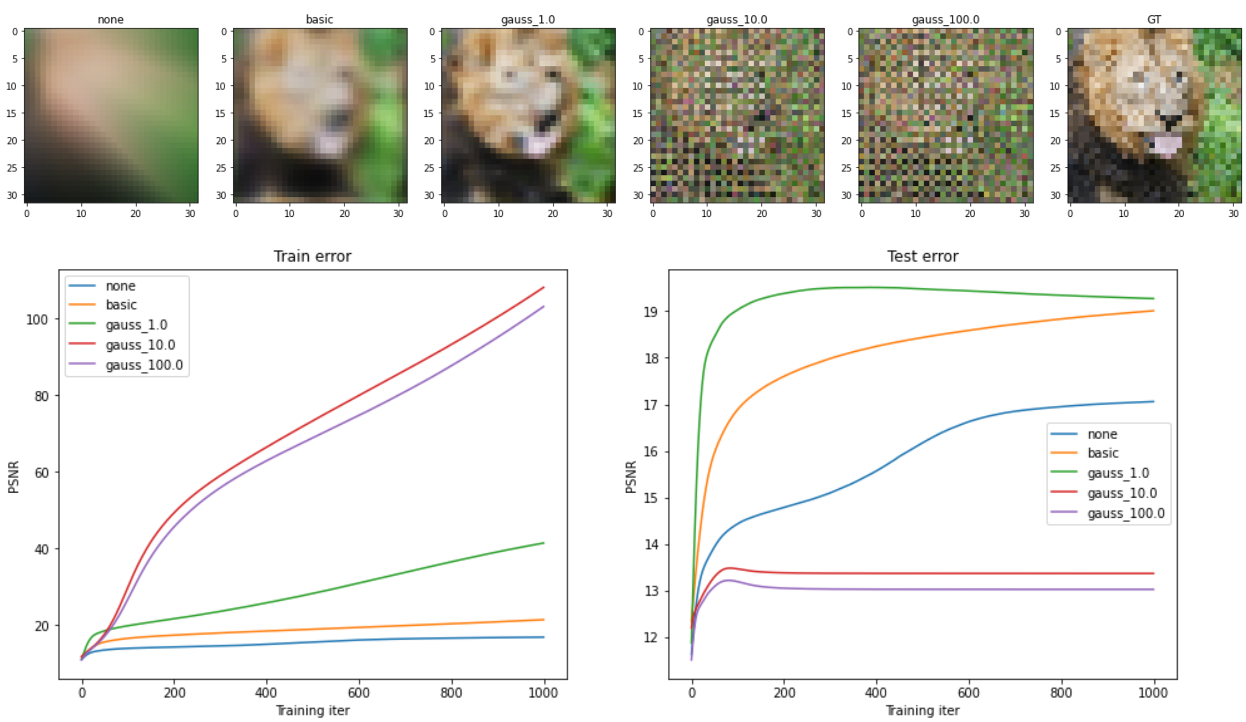
Observation: For learning rate 1e-5, the reconstructed output images for each mapping are less clear i.e. underfitting as result in over smoothed than the learning rate 1e-4.

Next, we picked up the best parameters from above i.e. learning rate = 1e-4, hidden_size = 256, epochs = 1000, and "Adam" Optimizer with default settings ($\beta 1 = 0.9$, $\beta 2 = 0.999$, $= 10{-}8$) and tried different number layers from 2 to 7.

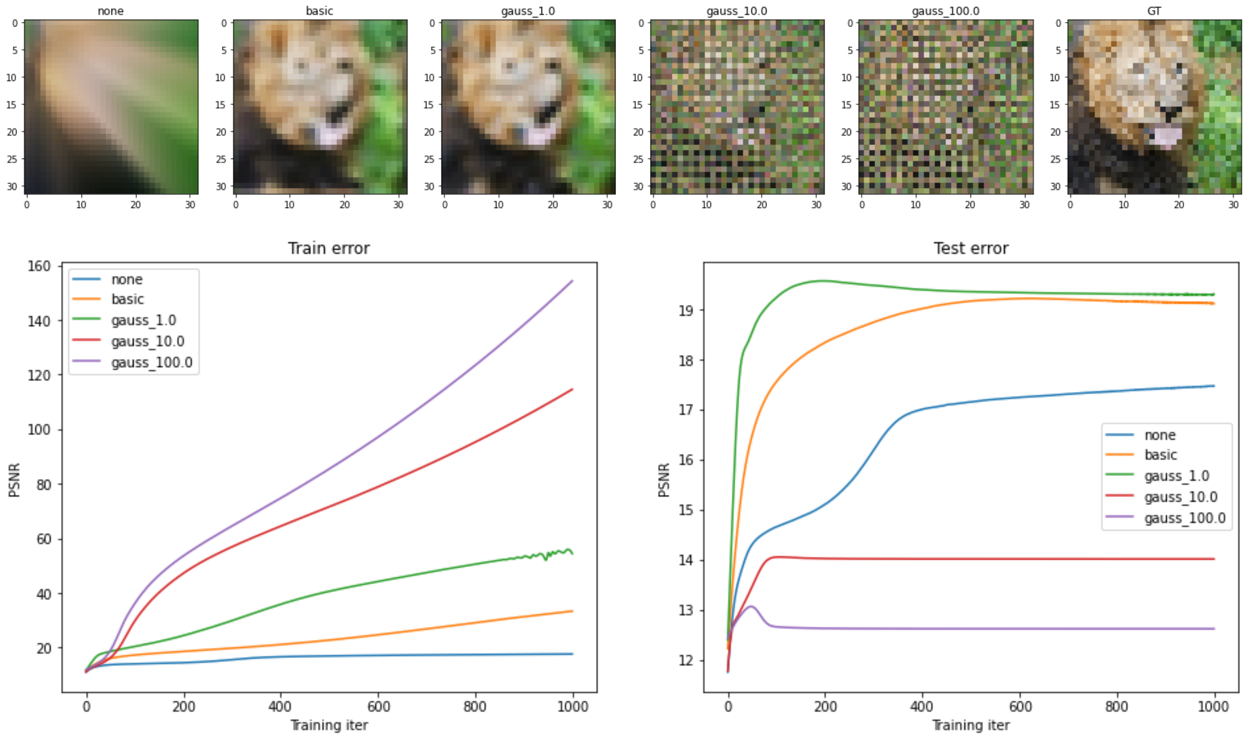
1. Number of layers: 2



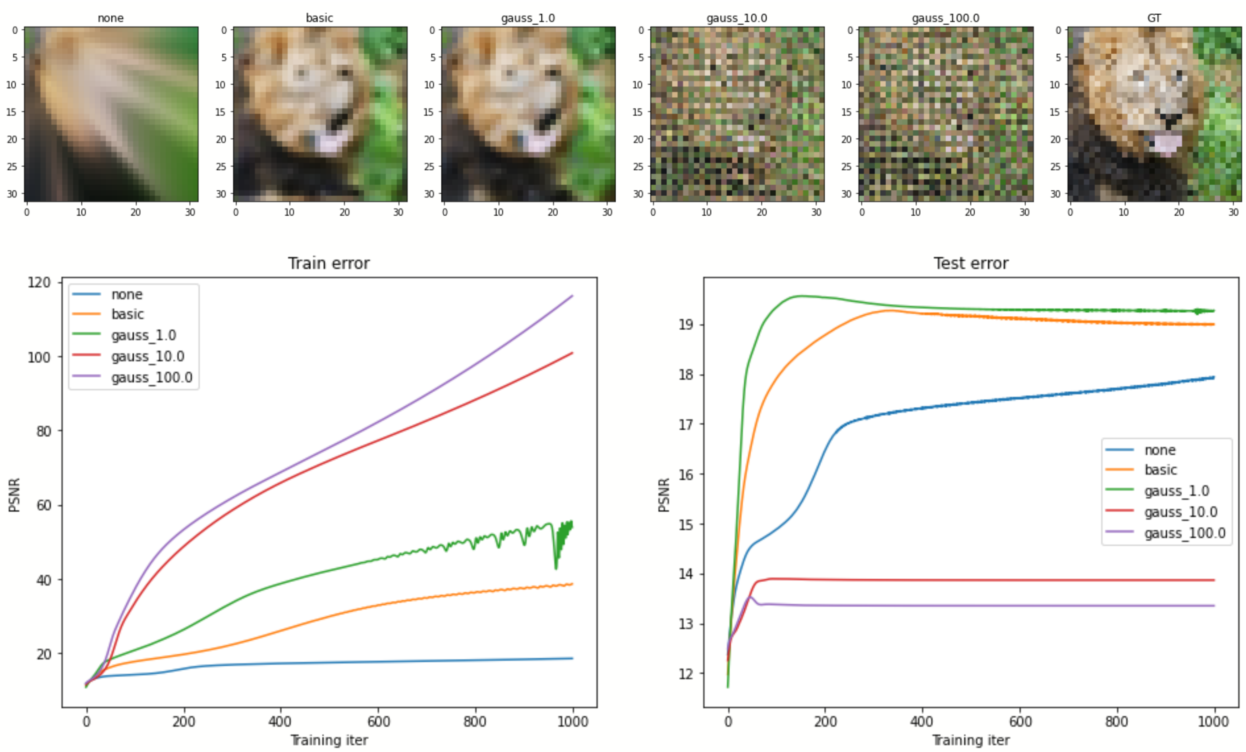
2. Number of layers : 3



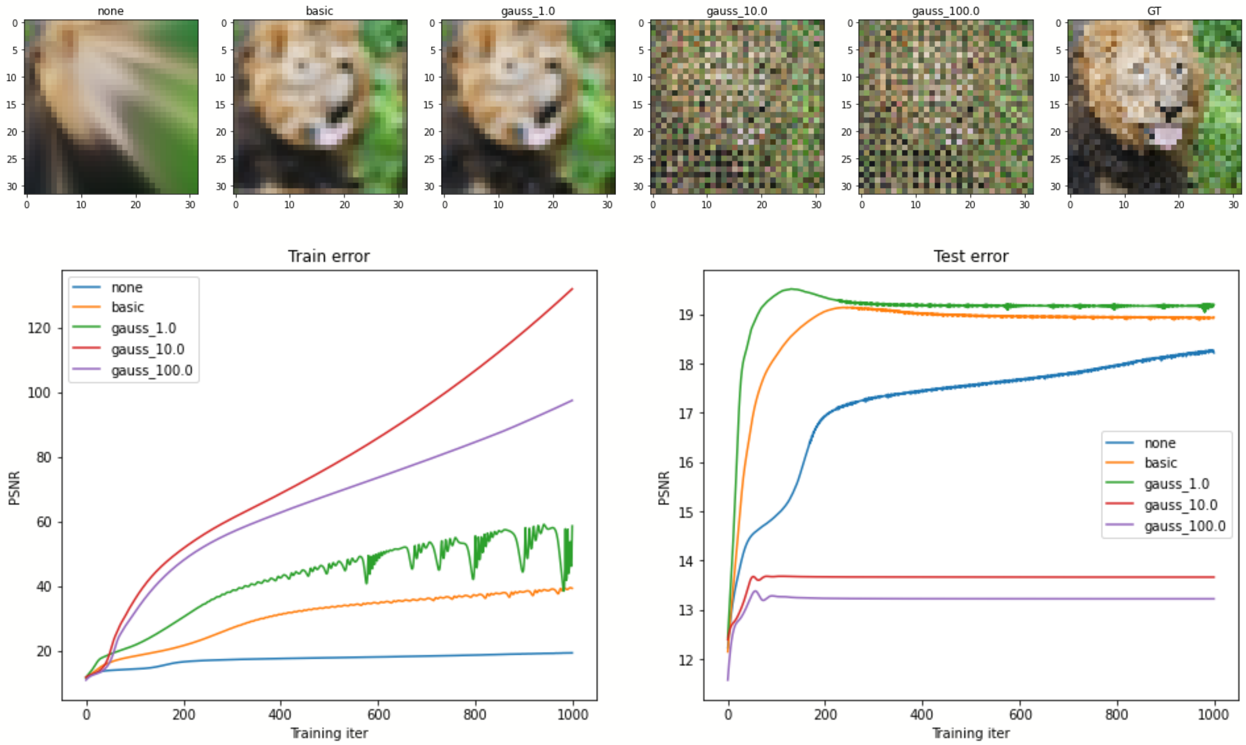
3. Number of layers : 4



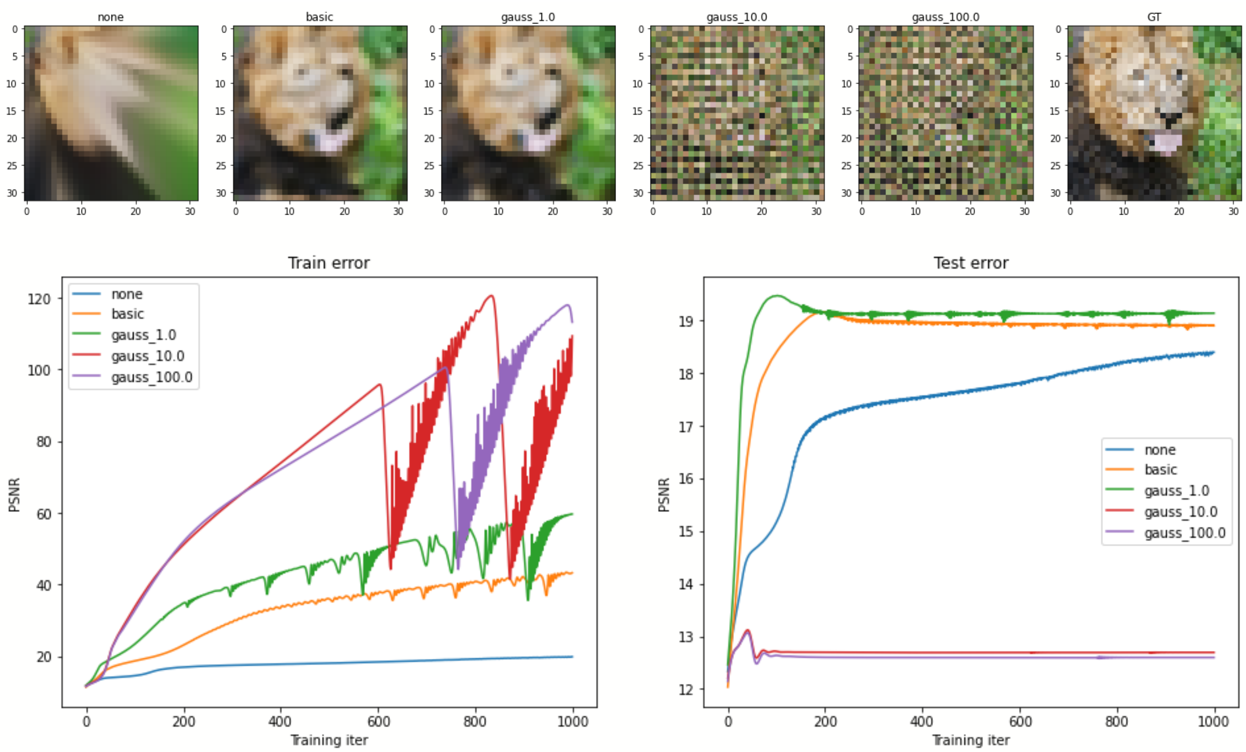
4. Number of layers: 5



5. Number of layers : 6



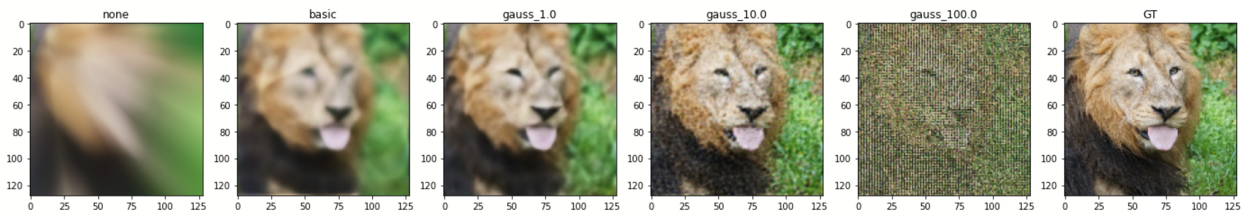
6. Number of layers: 7

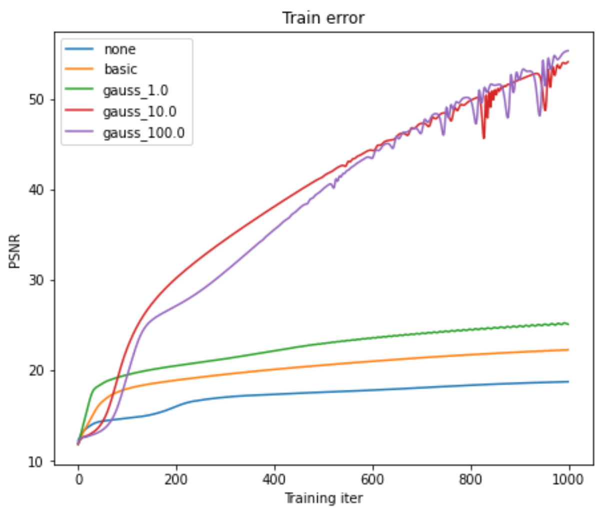


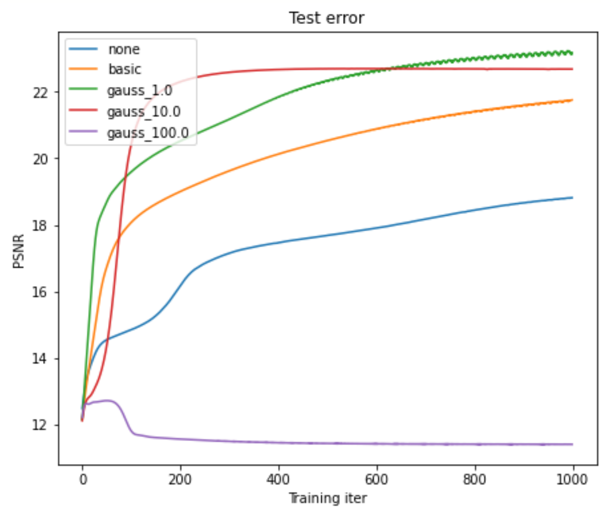
Observation: We identified the best parameter of the number of layers for each mapping type and used it in our final output.

For high resolution, we tried decreasing the learning rate from 1e-4 to 1e-5 with a number of layers = 5, hidden_size = 256, epochs = 1000, and "Adam" Optimizer with default settings ($\beta1 =$ 0.9, $\beta2 = 0.999$, $= 10-8$)

1. Learning rate = 1e-4

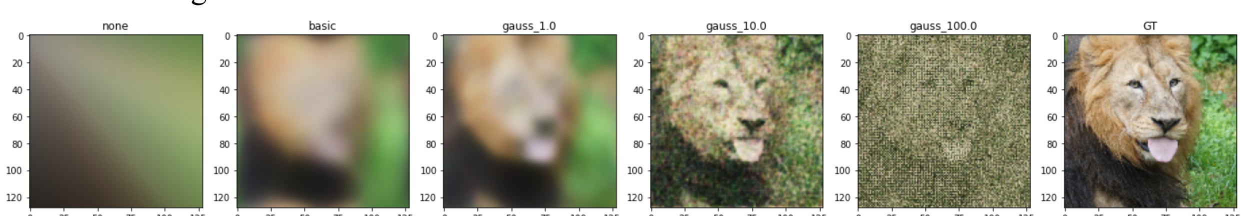


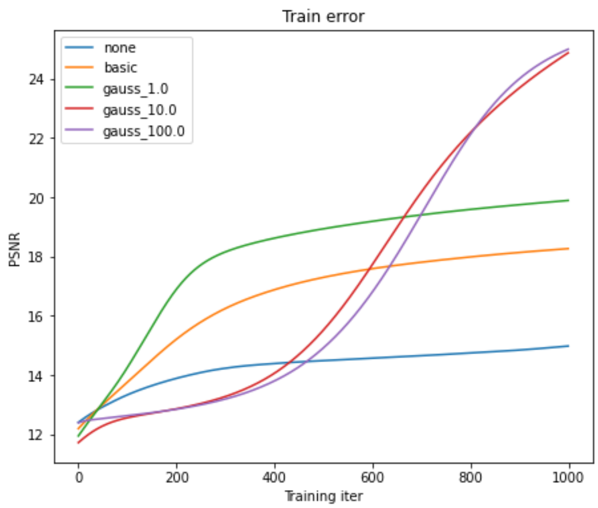


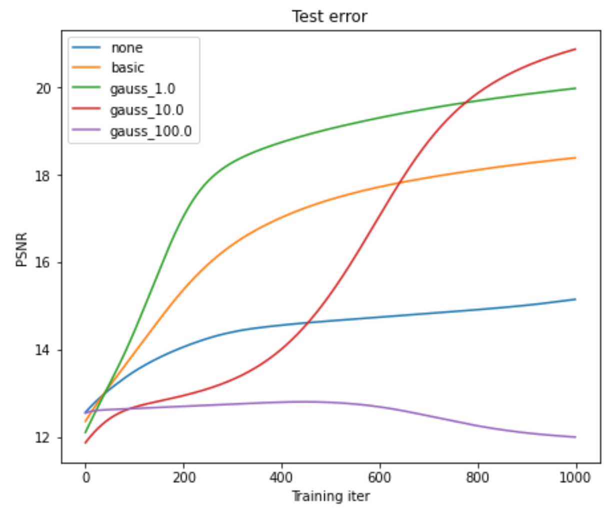


Observation: For learning rate 1e-4, we observed that output of none, basic, and gauss 1.0 are smoothened, and gauss 100.0 is noisier. Thus we selected learning rate 1e-4 for higher resolution with gauss 10.0.

2. Learning rate = 1e-5







Observation: We observed that output of none, basic, and gauss 1.0 are over smoothened for learning rate 1e-5 and for gauss 10 and gauss 100 are noisier for learning rate 1e-5. Thus we selected learning rate 1e-4 for higher resolution.

How did the performance of SGD and Adam compare?

The Adam method consistently improved picture reconstruction, despite the PSNR and error loss among the optimizers being very similar. This makes intuitive sense because the Adam optimizer utilizes moments and velocities to "guide" the gradient's direction; as a result, the model converges with a significantly less learning rate than the SGD optimizer would require for the same model parameters. In all of the experiments run, the Adam optimizer consistently outperforms the SGD optimizer.

How did the different choices for coordinate mappings functions compare?

For low-resolution images, none mapping and basic mapping under fits thus resulting in over-smoothed interpolation, and the reconstruction is extremely blurry. Whereas, large values of σ overfit (Gauss 10 and Gauss 100) result in noisy interpolation. We find that σ = 1 performs best for the low-resolution image.

Whereas for high-resolution images, no mapping, basic mapping, and low values of σ underfit (gauss 1.0) result in over-smoothed interpolation, and large values of σ overfit (Gauss 100.0) result in noisy interpolation. We find that gauss 10.0 performs best. This may imply that there is an ideal sigma value up to the point at which performance can be enhanced before declining. Thus, Gauss 10.0 is better than Gauss 100.0 for a high-resolution image.

Thus, we observed that if the image has a higher resolution, a larger sigma value (in our case gauss 10.0) may be appropriate to smooth out small details and noise, but a very high sigma value makes the image noisy (like for gauss 100.0). Conversely, if the image has a lower resolution, a smaller sigma value (in our case gauss 1.0) is more appropriate to preserve the edges and details in the image.

What insights did you gain from your own image example (Part 4)?

We choose a picture with more colorful pixels rather than an image with only a few color schemes. We tested with the Adam optimizer and the best parameters were obtained from hyper turning the high-resolution image. It appears that the reconstruction of the lion's image is better than our image. This could be because the ground truth image of the lion has fewer features than our image, which might make it simpler to reconstruct, or it could be because of the various details or composition of the image.