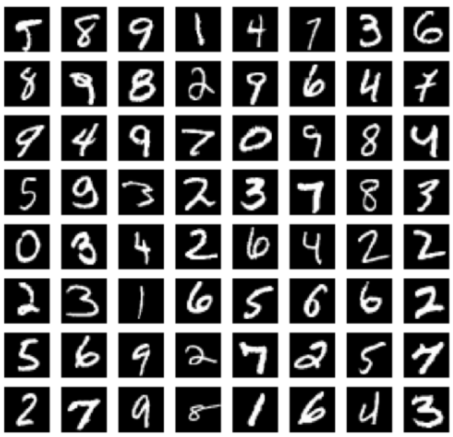
Denoising Autoencoder

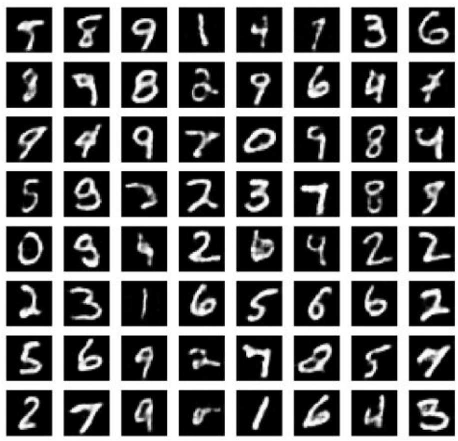
3 different types of noise with the inputs to the autoencoder. Epoch size is 10.

Visualization is as below

No Noise

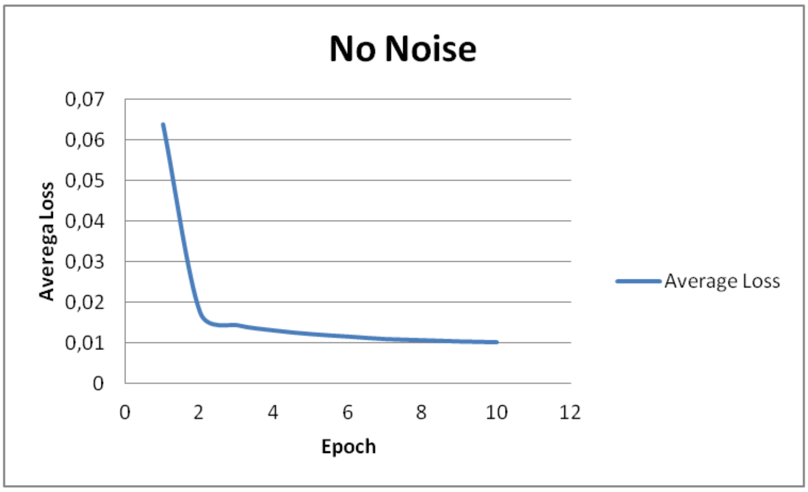
Original Inputs vs outputs from autoencoder





Visualization of losses with no noise on encoding:

No Noise	
Epoch	**Average Loss**
1	0,0638
2	0,0176
3	0,0145
4	0,0132
5	0,0123
6	0,0117
7	0,0111
8	0,0108
9	0,0105
10	0,0103

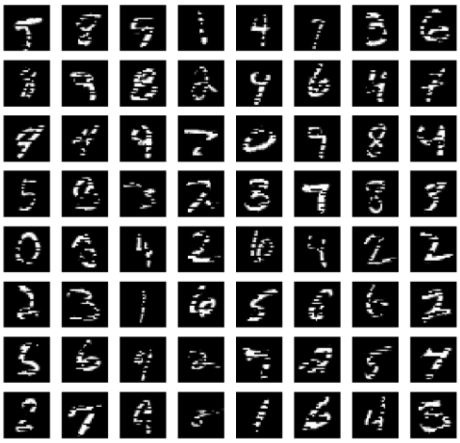


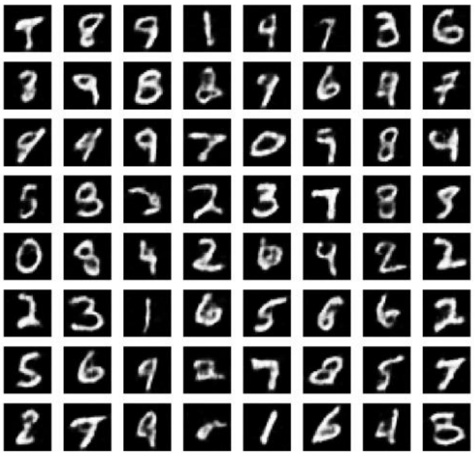
As expected, we achieved the lowest loss without noise.

Random Masking

Denoising autoencoder with Random Masking noise.

Inputs + random mask vs outputs from denoising.

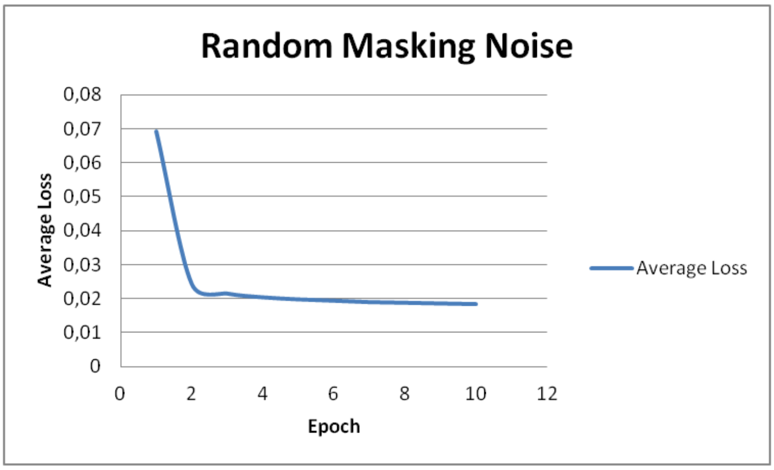




Visualizing losses:

Random Masking Noise

Epoch	Average Loss
1	0,0693
2	0,0243
3	0,0215
4	0,0204
5	0,0198
6	0,0194
7	0,0190
8	0,0188
9	0,0186
10	0,0184



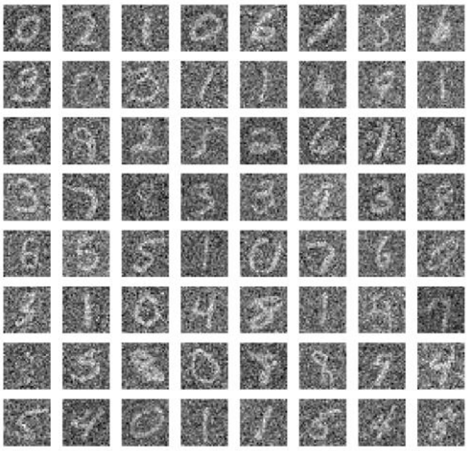
Losses are higher than without noise as expected. With the inputs to the autoencoder shifted by noise to the original images, the interpolation of the shifted images is thus expected to be shifted from the original images than without noise. Thus the perceived increase in loss.

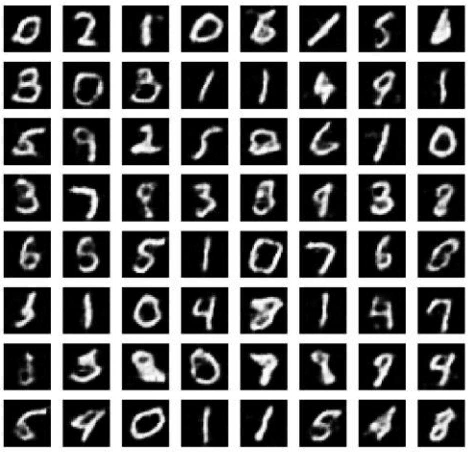
This type of noise, is the easiest to handle since only some pixels values are set to zero, leaving most of pixels values unchanged, these unchanged pixels(signals) give us the original context for handling the noise.

Gaussian Noise

Denoising autoencoder with Gaussian noise.

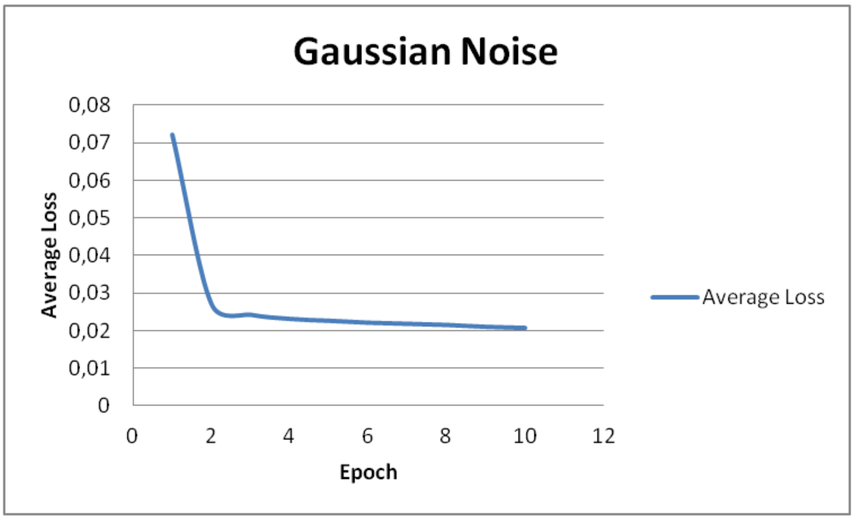
Inputs + Gaussian noise vs outputs





Visualizing losses:

Gaussian Noise	
Epoch	Average Loss
1	0,0721
2	0,0271
3	0,0243
4	0,0232
5	0,0227
6	0,0222
7	0,0219
8	0,0216
9	0,0211
10	0,0208



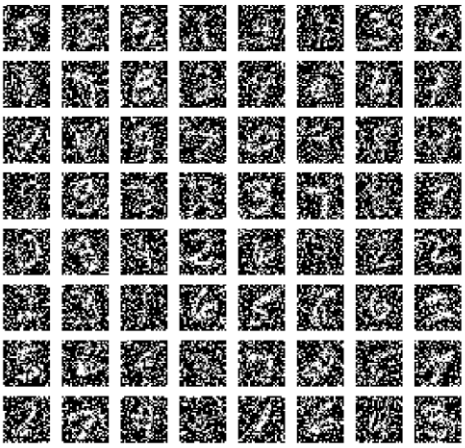
Observation : Losses > losses with random Masking noise > no noise losses

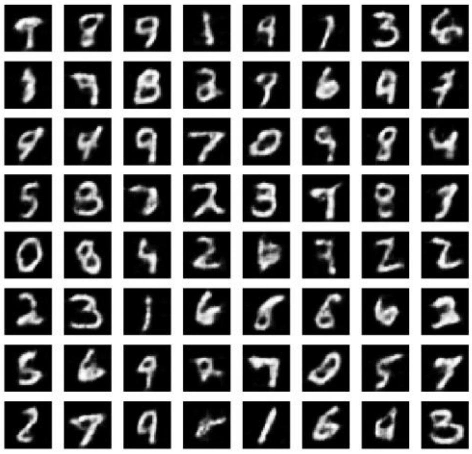
Guassina noise to inputs, shifts inputs wider from original images than random masking. Almost all pixel values are variated by some small margin.

Salt&Pepper Noise

Denoising autoencoder with Salt&Pepper noise.

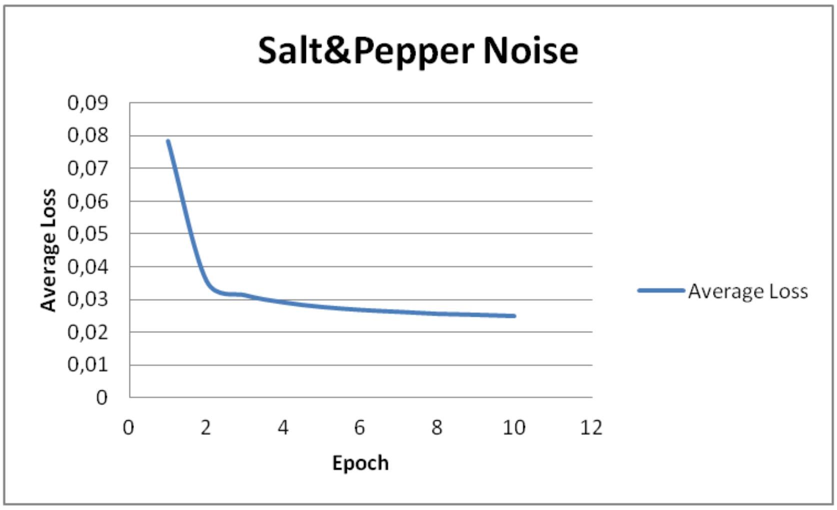
Inputs + salt and pepper noise vs outputs





Visualizing losses:

Salt&Pepper Noise	
Epoch	Average Loss
1	0,0785
2	0,0356
3	0,0312
4	0,0290
5	0,0276
6	0,0267
7	0,0261
8	0,0255
9	0,0252
10	0,0248



Produces the highest losses as noise shifts original images the most by changing pixel values by wider margins and removing some pixel values. Thus, the handling of both transformation is the toughest.

Transfer Learning

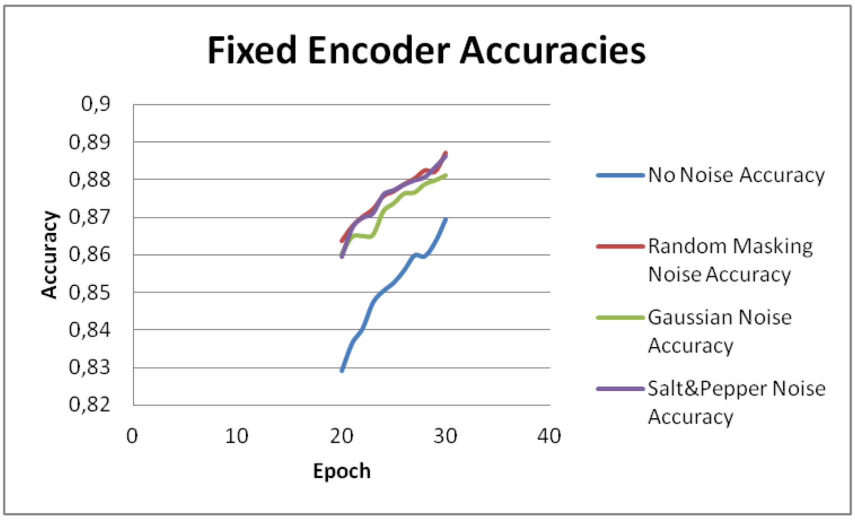
Using our trained encoders as feature extractors to classify images.

Epoch size = 30. Showing only last 10 epochs.

Fixed encoder feature extractor

Encoder weights are freezed. Linear classifier is optimized on top features.

Epoch	No Noise Accuracy	Random Masking Noise Accuracy	Gaussian Noise Accuracy	Salt&Pepper Noise Accuracy
20	0,8291	0,8636	0,8605	0,8596
21	0,8365	0,8675	0,8649	0,8672
22	0,8404	0,8701	0,8651	0,8699
23	0,8473	0,8721	0,8654	0,8711
24	0,8504	0,8756	0,8716	0,8760
25	0,8526	0,8769	0,8737	0,8772
26	0,8559	0,8788	0,8763	0,8787
27	0,8599	0,8803	0,8766	0,8798
28	0,8597	0,8825	0,8788	0,8807
29	0,8635	0,8822	0,8798	0,8833
30	0,8695	0,8872	0,8811	0,8862



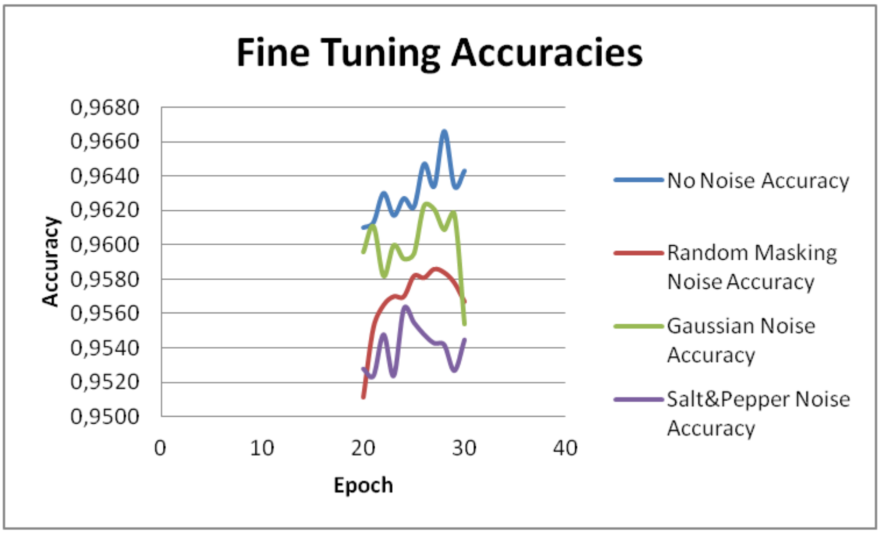
Graph of classifier validation accuracies on features extracted from different encoders

As fixed feature extractors, denoising encoders produce better-generalized feature since they learned to produce features that allow for some shift from the seen examples.

Finetuning

Optimizing both autoencoder and linear classifier. Epoch size = 30. Showing only last 10 epochs.

Epoch	No Noise Accuracy	Random Masking Noise Accuracy	Gaussian Noise Accuracy	Salt&Pepper Noise Accuracy
20	0,9610	0,9511	0,9596	0,9528
21	0,9613	0,9552	0,9611	0,9524
22	0,9630	0,9565	0,9582	0,9548
23	0,9617	0,9570	0,9600	0,9524
24	0,9627	0,9570	0,9592	0,9563
25	0,9622	0,9582	0,9595	0,9555
26	0,9647	0,9581	0,9623	0,9548
27	0,9634	0,9586	0,9621	0,9543
28	0,9666	0,9584	0,9609	0,9542
29	0,9634	0,9578	0,9618	0,9527
30	0,9643	0,9567	0,9554	0,9545



Graph of fine-tuned classifier validation accuracies with different encoders

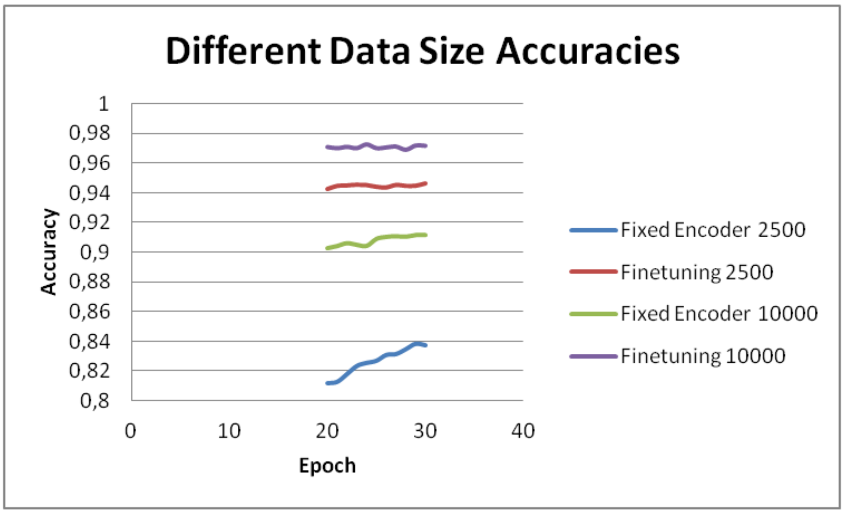
As fine-tuning allows the autoencoder to produce better features for the classifier, it thus provide higher accuracy on classification. The encoder is able to provide better feature on the original training samples to compensate for variations noise may introduce provided the trained data captures these variation and more. Adding noise may also suppress some statistics in the data the encoder can optimized on from the classification error, which could not be recovered.In this regard, the more random and wider the variation, the harder to retrieve lost features, making random masking and salt and pepper producing the least accuracies respectively.

As Gaussian noise is a constant standard variation to the input, same linear metafold exist in the shifted space, with some signals in this variation not recovered, providing less features for classification than with the original image on this dataset

Effect of Data size

Observed effects accuracies with varying number of samples: - 2,500 vs 10,000.

Epoch	Fixed Encoder 2500	Finetuning 2500	Fixed Encoder 10000	Finetuning 10000
20	0,8116	0,9423	0,9028	0,9710
21	0,8125	0,9444	0,9043	0,9702
22	0,8177	0,9447	0,9062	0,9711
23	0,8231	0,9452	0,9051	0,9702
24	0,8252	0,9449	0,9044	0,9728
25	0,8266	0,9438	0,9091	0,9702
26	0,8305	0,9434	0,9105	0,9707
27	0,8311	0,9450	0,9109	0,9713
28	0,8344	0,9444	0,9106	0,9691
29	0,8380	0,9445	0,9117	0,9720
30	0,8370	0,9460	0,9117	0,9718



More data samples provides higher generalization. As the data size increases, so does accuracies increases.