Raymond Yung
Report 4

My model is obviously an autoencoder, but instead of using a gaussian_noise function, I used convolution layers instead. I chose this method because the noise was already applied to our images, so there was no reason to add more noise. Furthermore, by choosing convolutional layers, not only did I have more control over the parameters, I also knew that it would be better suited for my images. I took some inspiration from our previous assignment, where we created our CNN. I reduced more of my parameters, using maxpool.

I needed to inversely mirror my encoding layer with my decoding layer, which is why I changed my inputs. This was done by checking the parameters in my encoding layer and identifying the correct values to put in accordingly.

Once these values were properly set, my total parameters for my model reached 3,552 total trainable parameters. These can be seen in the appendix below.

I didn't necessarily use a "gridsearch", rather I manually choose between results. Once I had found the best suited parameters, the other options I had left to really fine tune my model was choosing between the type of activation functions, optimizer, metrics, epochs, and batch number. Also, since this was an autoencoder, using accuracy or even rounded accuracy really didn't mean that much for this assignment, as the final grade for the assignment is assessed by seeing the readability of the final result.

I chose relu as my activation functions over selu simply because it seemed it could generalize better. Initially, I was going to use the "rounded_accuracy" from chapter 17 as my metric, but realized that pulling my model later resulted in an error since "rounded_accuracy" was a custom made function. So instead, I used MSE, simply because I was more familiar with it. I also kept nadams as my optimizer from my CNN that we created the week prior since I was confident with my previous model. Generally, my MSE scores have been consistent, around 3%, which could still mean that there was some overfitting. However, again with this project, a lot of the values would have to be done using our own human vision.

The last few adjustable parameters I had left were the number of epochs and batch_size. I had to choose a good number for both that prevent my model from overfitting and allow it to be generalized. The way I tested this was by simply trying all of them out. I tested epoch values from 10 and 60 and batch sizes between 32 and 64. At this point my mse scores were practically the same, but there were some minor differences in the result. You can see the results of my experiments in the table below. I ended choosing an epoch of 40 and a batch size of 64, because they seemed to generalize the best in my perspective, specifically looking at the last 6 in the bottom. The small details from this particular showed me whether or not a value was being overfit as in other cases like a batch size of 32 and epoch of 40, would show higher concentrations of pixels towards the middle of the image, which tells me that a it would do the same thing for a similar but still different image.

Appendix

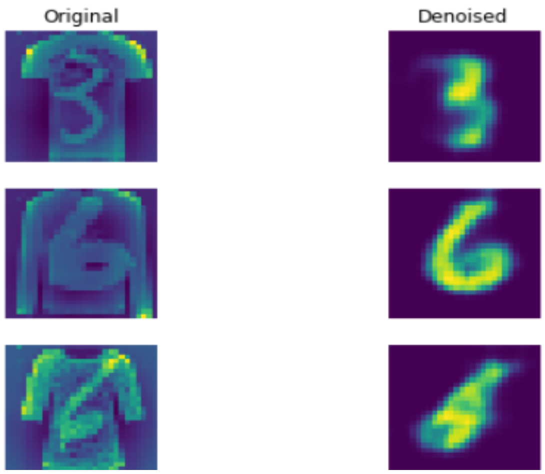
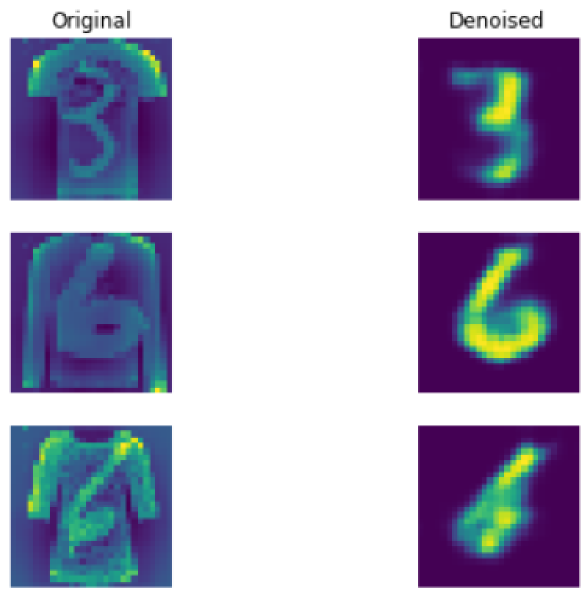
Figure 1: Encoder Summary

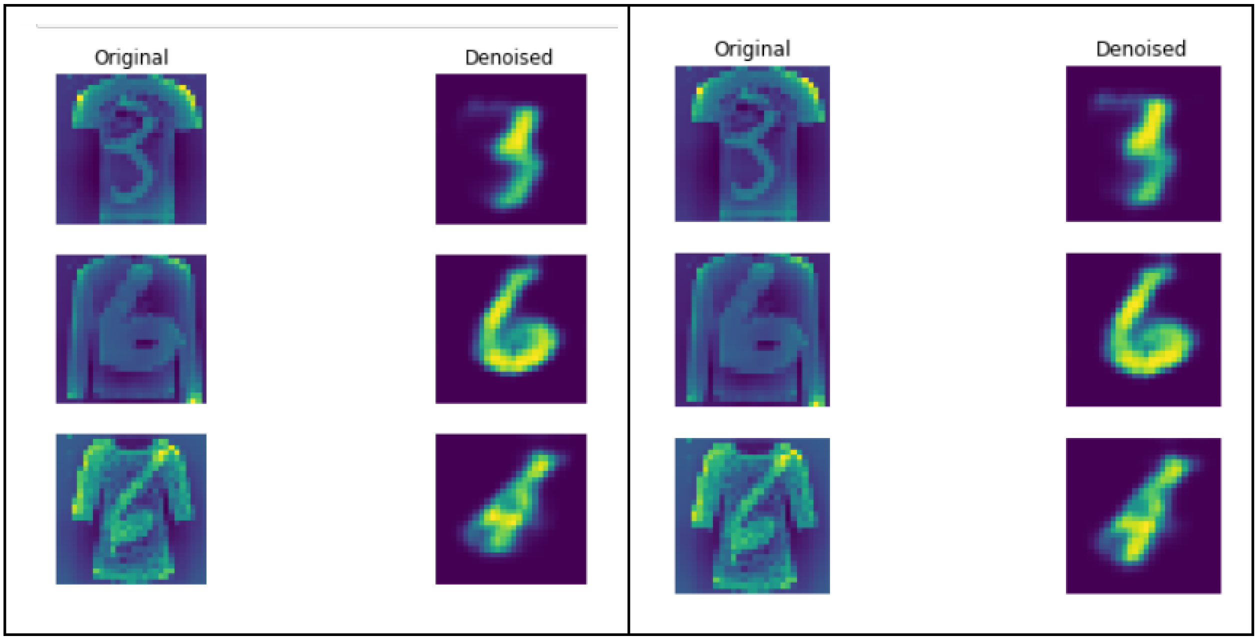
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
reshape (Reshape)            (None, 28, 28, 1)         0         
_________________________________________________________________
conv2d (Conv2D)              (None, 28, 28, 1)         325       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 7, 7, 1)           0         
_________________________________________________________________
flatten (Flatten)            (None, 49)                0         
_________________________________________________________________
dense (Dense)                (None, 11)                550       
=================================================================
Total params: 875
Trainable params: 875
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Decoder Summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 196)               2352      
_________________________________________________________________
reshape_1 (Reshape)          (None, 14, 14, 1)         0         
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 28, 28, 1)         325       
=================================================================
Total params: 2,677
Trainable params: 2,677
Non-trainable params: 0
_________________________________________________________________
```

Table 1 : Epoch Batch Size Experiments

Epoch 10; Batch Size 64	Epoch 40; Batch Size 64
	
Epoch 60; Batch Size 32	Epoch 40; Batch Size 32


Original
Denoised
Original
Denoised