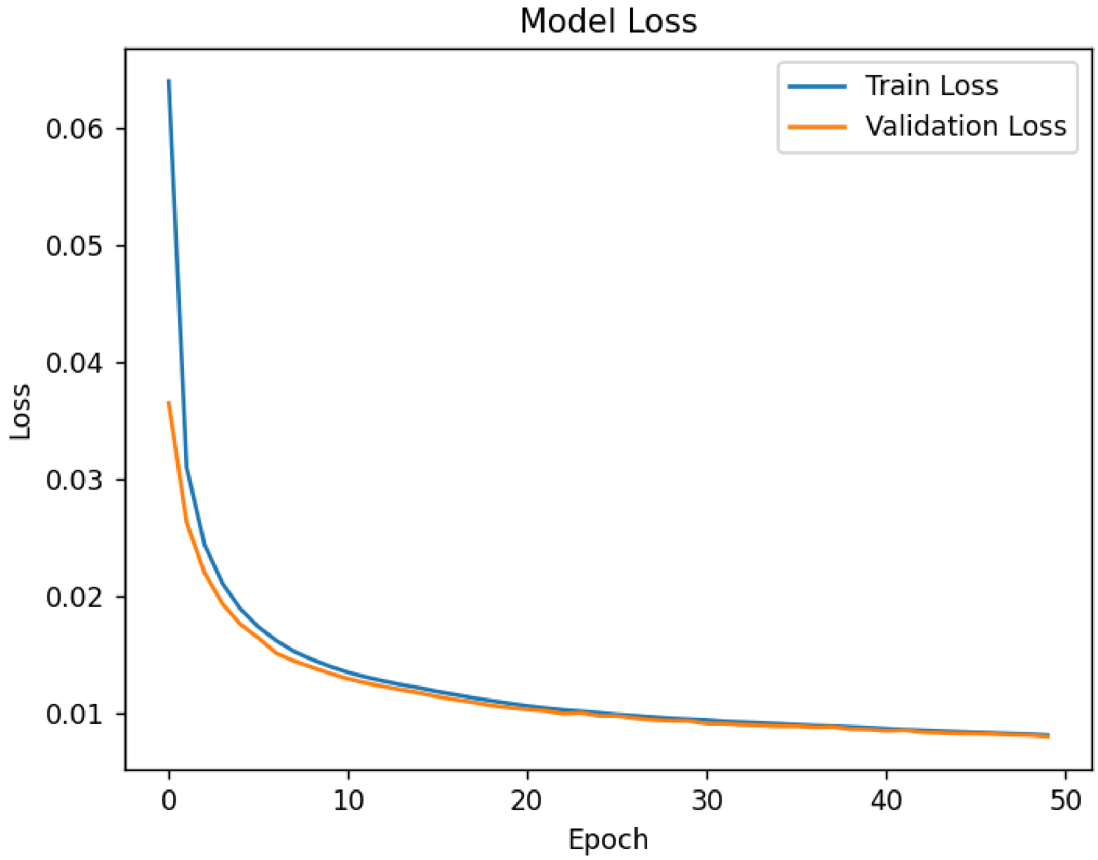
Assignment 5

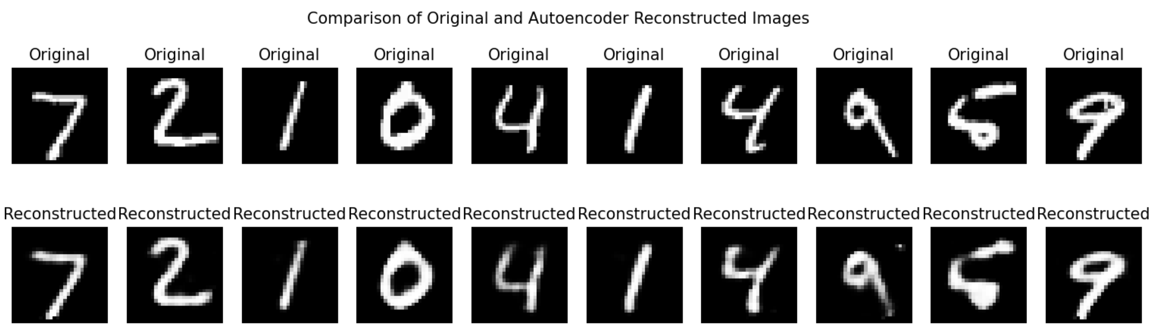
Exercise 1



Training/Validation Loss Plot:

Training Loss: The loss decreases progressively, suggesting that the model is minimizing errors as it is trained.

Validation Loss: The validation loss also decreases progressively throughout the epochs.

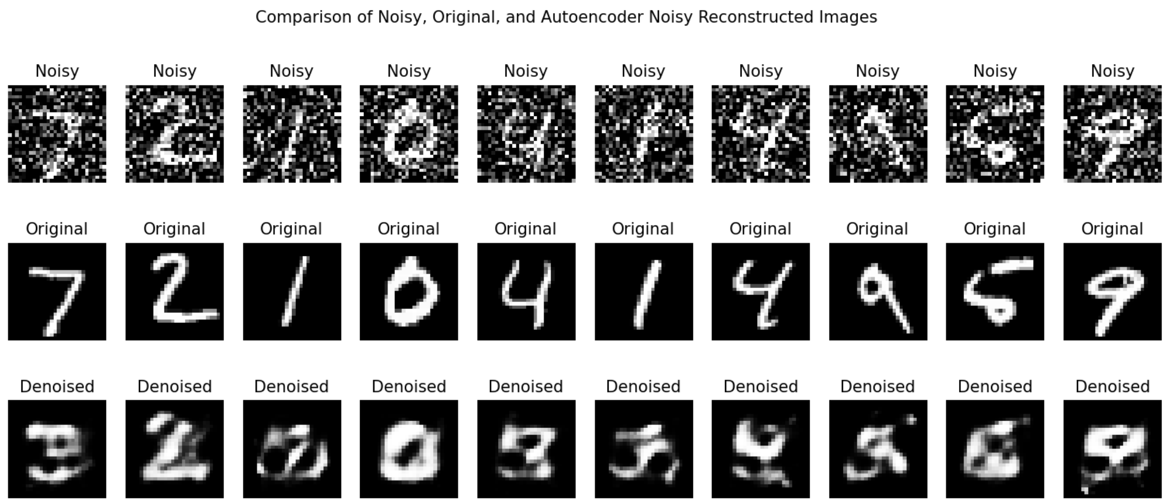


```
Autoencoder Reconstruction Error: 0.007932995
```

<u>Original and Reconstructed Images, and Reconstruction Error:</u>

<u>Original/Reconstructed images comparison:</u> As we can see from the image above, the Autoencoder does a very good job in reconstructing the images from the dataset. In most cases, the original image and its corresponding reconstruction are almost identical, and only in a few cases we can see minimal differences.

<u>Reconstruction Error:</u> The reconstruction error (Mean Squared Error) is very low in this case, indicating that the autoencoder does a very good job in reconstructing the images.



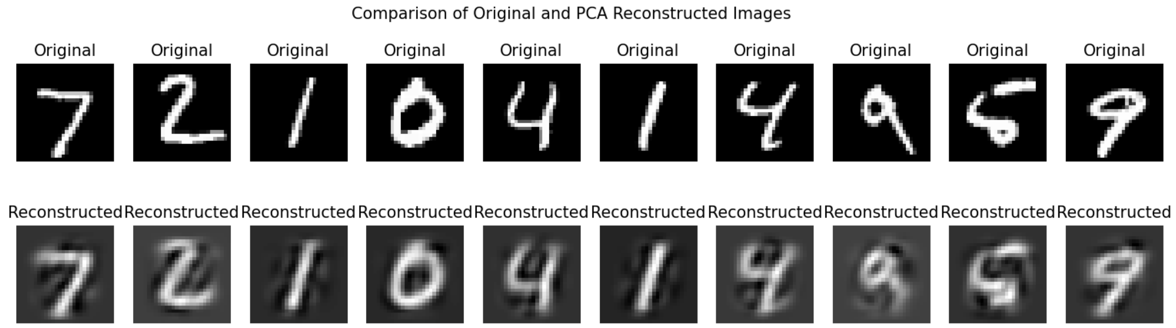

```
Autoencoder Noisy Reconstruction Error: 0.07042574
```

<u>Noisy, Original and Reconstructed Images, and Noisy Reconstruction Error:</u>

<u>Noisy/Original/Reconstructed images comparison:</u> As we can see from the image above, the Autoencoder struggles in reconstructing the images when noise is added, in most cases, while in some cases it does a good job.

<u>Reconstruction Error:</u> The reconstruction error (Mean Squared Error) is higher than before indicating that the autoencoder doesn’t perform as well when it comes to reconstructing the noisy images.

<u>Exercise 2</u>

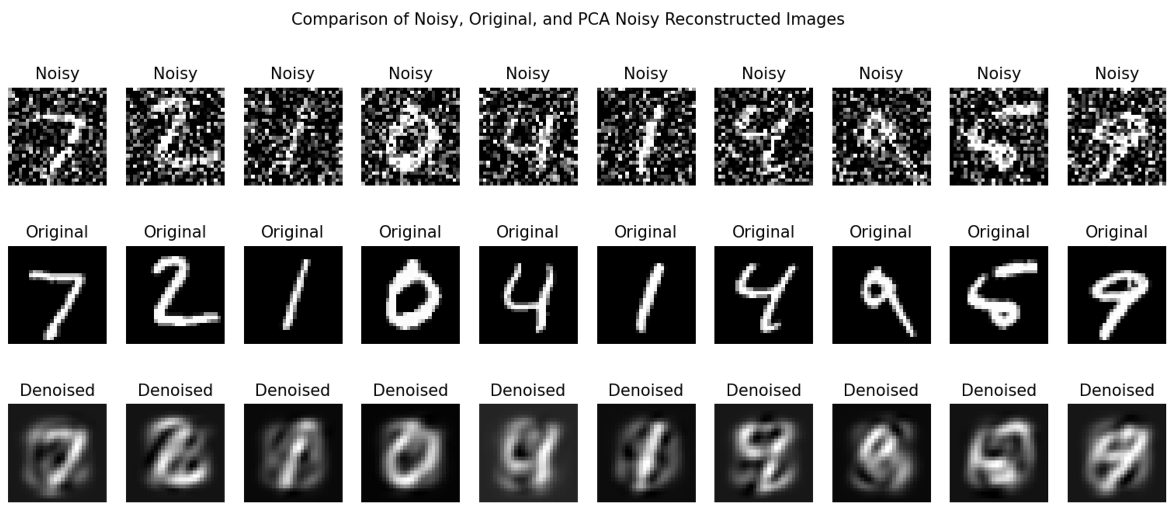



```
PCA Reconstruction Error: 0.01682822
```

Original and Reconstructed Images, and Reconstruction Error:

Original/Reconstructed images comparison: As we can see from the image above, PCA does a good job in reconstructing the images from the dataset. In most cases, the original image and its corresponding reconstruction can be easily identified, and only in a few cases we can see important differences. Also, the background in the reconstructions is different (more grey) than the background in the original images.

Reconstruction Error: The reconstruction error (Mean Squared Error) is low in this case, indicating that PCA does a good job in reconstructing the images.




```
PCA Noisy Reconstruction Error: 0.032673255851108425
```

Noisy, Original and Reconstructed Images, and Noisy Reconstruction Error:

Noisy/Original/Reconstructed images comparison: As we can see from the image above, in most cases we can identify the denoised reconstructed image and its corresponding original image, but still PCA struggles a bit to reconstruct the images correctly when noise is added.

Reconstruction Error: The reconstruction error (Mean Squared Error) is higher than before indicating that PCA doesn't perform as well when it comes to reconstructing the noisy images.

Comparing PCA and Autoencoder Results

1. **Reconstruction Quality**:
    - **Autoencoder**: Non-linear transformations allow the autoencoder to capture complex features in the data, making it more flexible for capturing image details. The reconstructed images from the autoencoder tend to be sharper and more similar to the original images. This can be seen from the results when we compare the reconstructed images and the reconstruction error.
    - **PCA**: PCA is limited in capturing complex, non-linear patterns, which can result in blurred and less detailed reconstructions. This can be seen from the results when we compare the reconstructed images and the reconstruction error.
2. **Denoising Capability**:
    - **Autoencoder**: As we can see from the denoised reconstructions, the autoencoder generally performs ok in removing noise due to its ability to learn non-linear mappings, making it good for denoising tasks. The reconstruction error in this case is lower than the PCA reconstruction error for the denoised images.
    - **PCA**: When comparing the results (visual and reconstruction error) of PCA and the autoencoder for the denoised reconstructed images, we can see that PCA performs a somewhat better than the autoencoder in this case.
3. **Computational Complexity**:
    - **Autoencoder**: Training an autoencoder can be computationally intensive, especially with larger datasets and deeper architectures.
    - **PCA**: PCA is generally faster to compute as it involves a simple linear transformation.
4. **Interpretability**:
    - **Autoencoder**: The latent space in an autoencoder may not always be interpretable, as it depends on learned features, which can be hard to analyse directly.

- **PCA**: PCA provides clear interpretability since each principal component represents a direction of maximum variance in the data.

- **Autoencoder** performs better for image reconstruction due to its ability to capture non-linear relationships.
- **PCA** is a useful technique for linear dimensionality reduction but is limited in capturing complex image structures, which affects its reconstruction quality for images. However, it remains faster and computationally lighter, making it a good choice for simpler tasks.

Advantages and Disadvantages of Autoencoders and PCA

Autoencoders

Advantages

1. **Non-linear transformation**: Unlike PCA, autoencoders can learn non-linear transformations, which makes them effective for more complex datasets that aren't well-represented in a linear space.
2. **Flexible architecture**: Autoencoders are highly customizable. They can be adjusted with additional layers, different activation functions, or regularization techniques to fit various types of data.
3. **Handles large, complex datasets**: Autoencoders can scale well with large datasets, especially when using deep neural network architectures.
4. **Feature learning**: They can learn useful features automatically from raw data, which can improve performance on tasks like image or text classification.
5. **Denoising**: Denoising autoencoders can effectively learn representations even with noisy data, making them useful for data cleaning and feature extraction tasks.

Disadvantages

1. **Requires more data and computational power**: Training an autoencoder, especially a deep one, requires more data and computational resources than PCA. This can be a disadvantage in low-data situations.
2. **Complexity and training time**: Autoencoders require significant time and expertise to tune and train. They are prone to issues like overfitting if not managed carefully.
3. **Difficult to interpret**: Unlike PCA, which provides clear principal components, the learned representations in autoencoders are often hard to interpret, making it challenging to understand the contribution of each component.
4. **Risk of poor reconstruction**: If not trained correctly, autoencoders can have poor reconstruction accuracy, especially when working with highly variable datasets.

Principal Component Analysis (PCA)

Advantages

1. **Simplicity and interpretability**: PCA is a straightforward, linear method that is easy to interpret and implement. The resulting principal components are ordered by variance, making it easy to understand how much information each component holds.
2. **Low computational cost**: PCA is computationally cheaper and faster than training an autoencoder, making it suitable for real-time applications or exploratory data analysis.
3. **Effective for linear data**: For data with linear relationships, PCA can be very effective, often capturing the majority of the variance with only a few components.
4. **Reduced risk of overfitting**: Because PCA is a linear and non-parametric method, it has a lower risk of overfitting than neural network-based methods, especially with small datasets.

Disadvantages

1. **Limited to linear transformations**: PCA is a linear method, so it struggles with capturing complex, non-linear structures in data. This can result in poor performance for data with non-linear patterns.
2. **Sensitive to scaling**: PCA is sensitive to the scale of the data. Features with higher variance dominate, so it requires careful preprocessing and scaling for balanced results.
3. **Not suitable for large, complex datasets**: While computationally inexpensive, PCA becomes difficult to compute for very large datasets or when the number of dimensions is extremely high.
4. **Fixed components**: Once trained, PCA components are fixed and cannot be adapted to new data without re-running the algorithm on the full dataset, which can be impractical for streaming data.