Autoencoder Assignment Report

Objective:

The objective of this assignment is to implement and evaluate both Fully Connected and Convolutional Autoencoders (CAE) for image reconstruction tasks using the MNIST dataset. The focus is on building compact latent representations (bottlenecks) and understanding their effect on reconstruction quality.

Dataset:

You will use the MNIST handwritten digit database, which is available through TensorFlow and PyTorch libraries.

Example (TensorFlow):

(x_train, _), (x_test, _) = tensorflow.keras.datasets.mnist.load_data()

Part 1: Fully Connected Autoencoder

- Implement a fully connected autoencoder using TensorFlow or PyTorch.
- The autoencoder should have a bottleneck layer with only 2 neurons.
- Use Mean Squared Error (MSE) as the objective (loss) function.
- Train your model on MNIST training data and validate on test data.
- Plot training and test loss curves.
- Randomly select 10 test images, encode and decode them, and visualize the reconstructions.
- Experiment with different bottleneck sizes and layer depths, and compare the resulting reconstructions.
- Discuss how bottleneck size and architecture complexity impact the reconstruction quality.

Part 2: Convolutional Autoencoder (CAE)

- Implement a convolutional autoencoder using Conv2D and Conv2DTranspose layers (TensorFlow) or the PyTorch equivalents.
- Use MSE as the loss function.
- Start with a bottleneck of size 2.
- Train on MNIST and plot training and test loss curves.
- Select 10 test images, encode and decode them, and visualize the outputs.
- Evaluate if the decoded images are recognizable. If not, modify the CAE architecture by increasing the bottleneck size or number of layers.
- Your goal is to find the smallest bottleneck size that still allows for visually recognizable reconstructions.

Deliverables:

- Python code for both autoencoders.
- Plots of training/test loss and decoded images.
- This PDF report summarizing methods, architectures, and results.

Tools and Libraries:

- Python
- TensorFlow or PyTorch
- NumPy, Matplotlib