Nurdin Khoirurizka (諾丁)
312540025

Report Clustering Methods

Homework 2

NYCU

This report was discus about implementation of auto-encoder on MNIST dataset. The embedding result of will use as dataset on clustering and classification. The method that be used on this clustering was k-mean and agglomerative. Then classification used neural-network with 4 layer. Besides that, same method will implement on handwriting A-Z dataset to know the performance of the method when its implement on another dataset.

On the first experiment, the dataset MNIST split to 60,000 data train and 10,000 data test. Then, the dataset was split on 10 label, from 0 to 9. The distribution of dataset and each label shown on Figure 1.

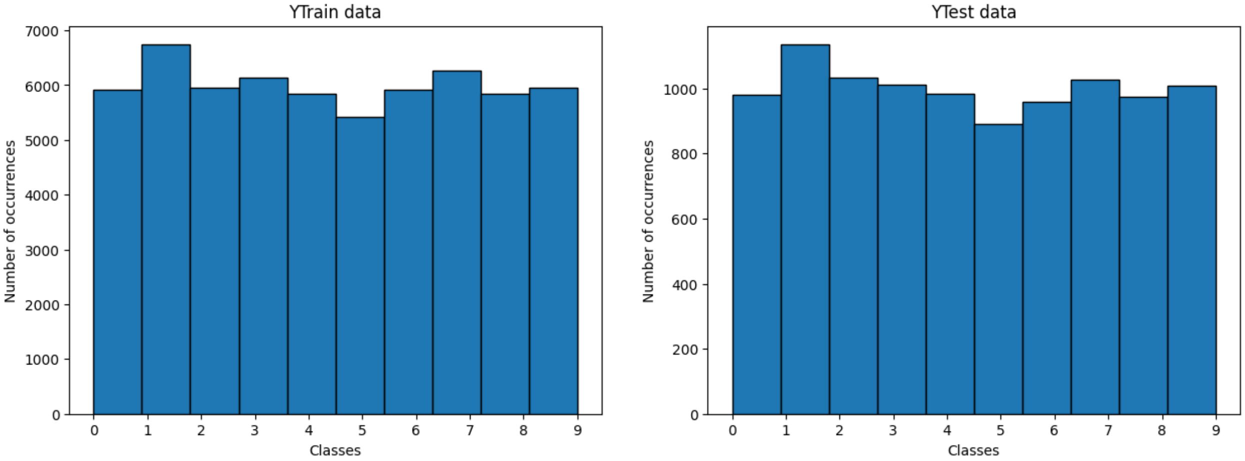

Figure 1. MNIST dataset distribution.

The auto-encoder consists of two main components: an encoder and a decoder. The encoder compresses the input into a latent space representation, while the decoder reconstructs the input from this representation. The goal of the auto-encoder is to minimize the reconstruction error, ensuring that the output closely matches the original input. By training on the MNIST dataset, the auto-encoder learns to capture the features of handwritten digits.

In this experiment, the architecture of the auto-encoder consists of an input layer with 784 neurons, an encoder with 128 neurons, a latent space, and a decoder with 784 neurons. The auto-encoder model is compiled using the RMSprop optimizer and binary cross-entropy loss. The model is trained on the training dataset and validated using the test dataset, running for 100 epochs with shuffling allowed.

The training results showed a loss of 0.0694 and a validation loss of 0.0690, which were the optimum values achieved over 100 epochs, as shown in the graph in Figure 2. To evaluate the model, five random images from the MNIST test data were chosen and tested with the auto-encoder model. The comparison of the input and output of the auto-encoder, displayed in Figure 3, clearly demonstrates that the model successfully achieved its purpose by ensuring the output is similar to the original input.

Nurdin Khoirurizka (諾丁)
312540025

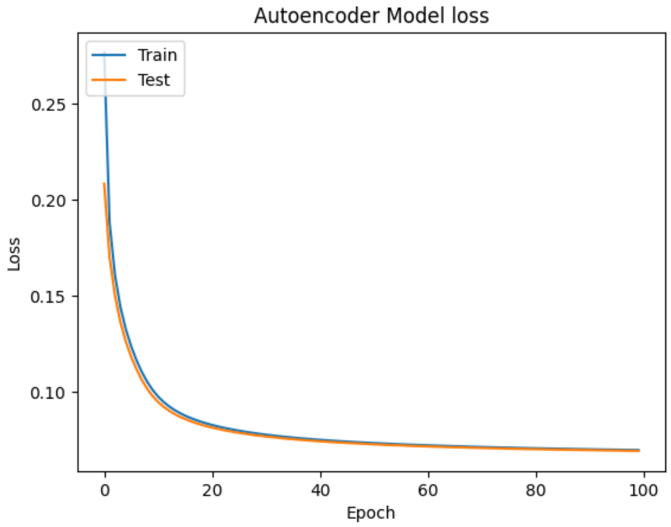

Figure 2. Auto-encoder result of MNIST model loss

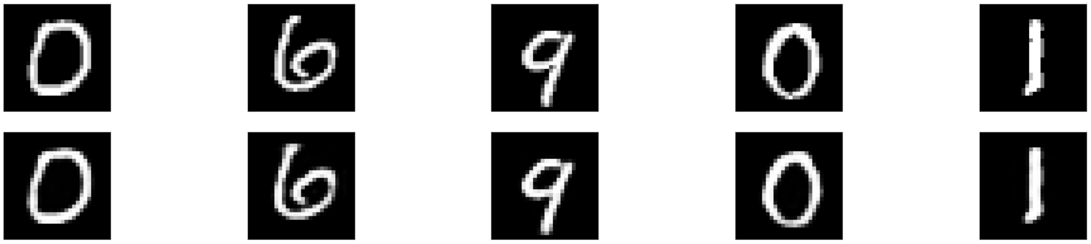

Figure 3. Auto-encoder result of MNIST

The output images from the auto-encoder are clustered using K-means and agglomerative clustering, both configured to create 10 clusters. The results of the K-means clustering, shown in Figure 4, and the agglomerative clustering, shown in Figure 5, indicate that the clustering results were not satisfactory. The data was almost uniformly distributed, making it difficult to discern distinct clusters corresponding to labels 0 to 9. This issue is further illustrated in Figure 6 for K-means and Figure 7 for agglomerative clustering. Given these challenges, classification methods are needed as an alternative approach to accurately distinguish between the handwritten digit labels.

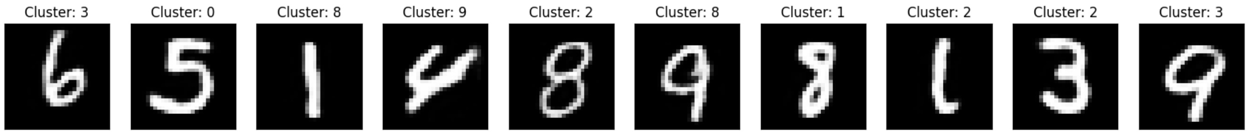

Figure 4. Clustering result of MNIST using k-mean

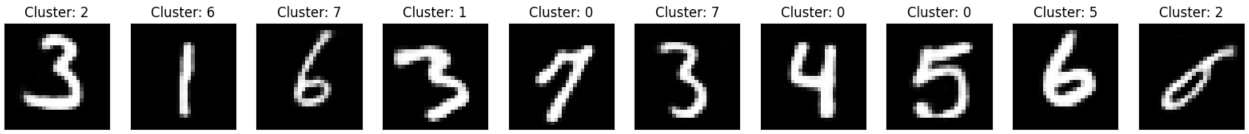

Figure 5. Clustering result of MNIST using agglomerative

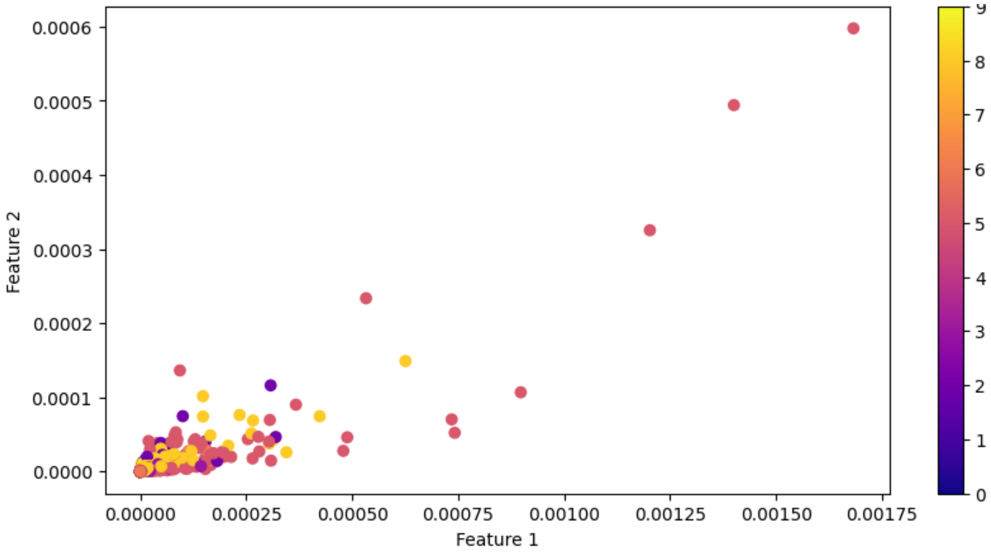

Figure 6. Clustering distribution of MNIST using K-mean

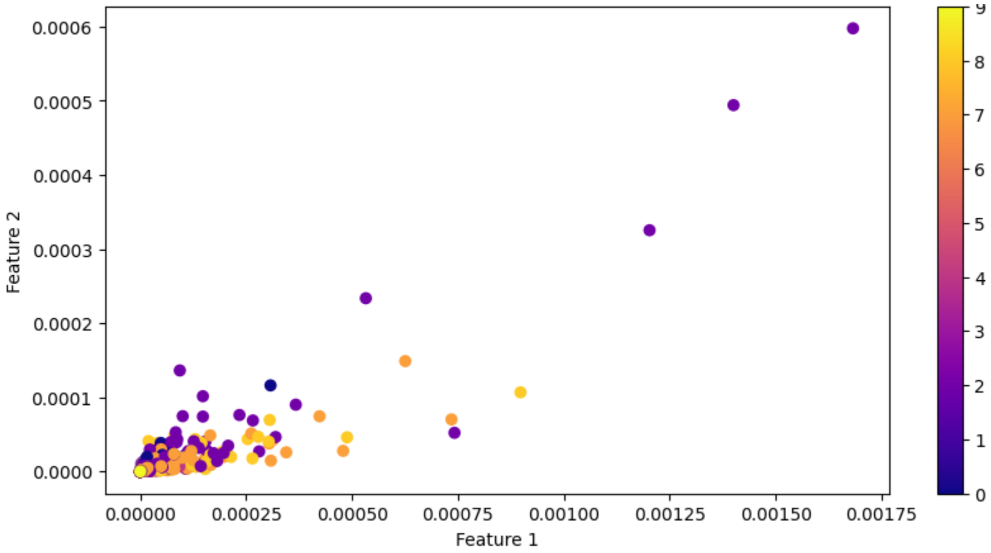

Figure 7. Clustering distribution of MNIST using agglomerative

The classification of the MNIST dataset is implemented using a multi-layer neural network. The architecture includes an input layer with 784 neurons, three hidden layers with 256, 64, and 32 neurons respectively, and an output layer with 10 neurons. The model is compiled using the Adam optimizer and binary cross-entropy loss, and is trained on the MNIST training data and validated using the MNIST test data.

After training, the model achieved a loss of 0.0011, an accuracy of 0.9987, a validation loss of 0.0260, and a validation accuracy of 0.9807. This indicates a validation error rate of 1.93%, demonstrating that the model is highly accurate. To further validate the model's accuracy, 10 random test samples were selected and their labels were predicted. The results, shown in Figure 8, confirm that the model's predictions are accurate, outperforming the clustering results.

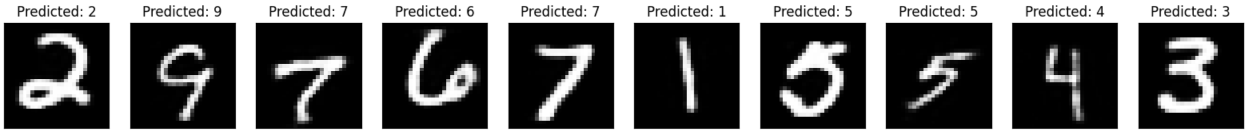

Figure 8. Classification result of MNIST using multi-layer neural network

To demonstrate the effectiveness of using an auto-encoder as a preprocessing step for multi-layer neural network classification, this strategy was applied to another dataset: handwritten characters from A to Z.

The handwritten character dataset consists of 372,451 images, which were split into 85% training data and 15% test data. This resulted in 316,583 training images and 55,868 test images. Since the dataset includes handwritten characters from A to Z, it contains 26 labels, with label 0 representing 'A', label 1 representing 'B', and so on up to label 25 representing 'Z'. The labels were split between the training and test datasets with similar distributions, as shown in Figure 9.

The auto-encoder as a preprocessing step was aimed to improve the classification performance on this new dataset, validating the strategy's effectiveness across different types of data.

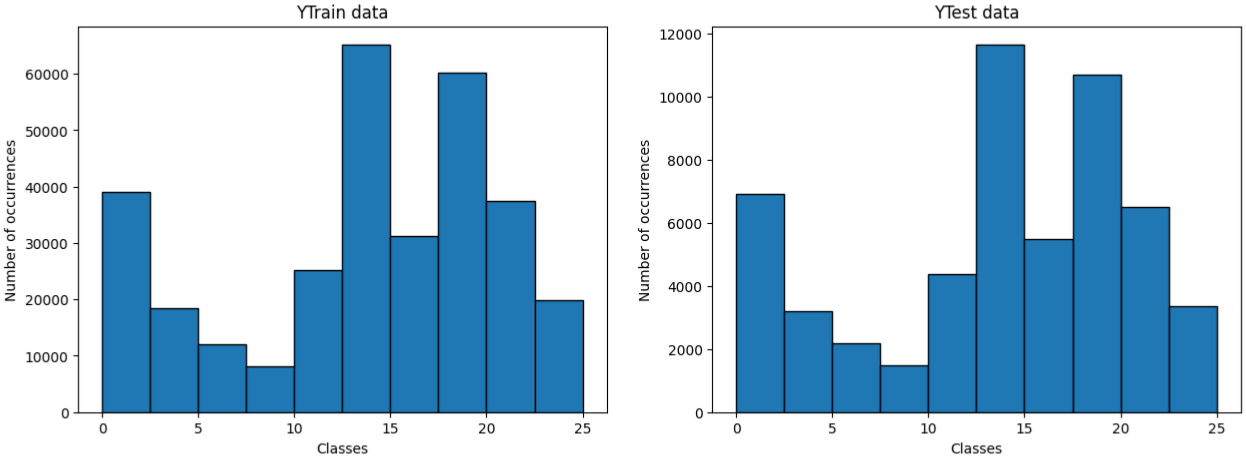

Figure 9. Handwriting character A-Z dataset distribution

For this dataset, the auto-encoder architecture includes an input layer with 784 neurons, an encoder with 128 neurons, a latent space, and a decoder with 784 neurons. The model was compiled using the RMSprop optimizer and binary cross-entropy loss. It was trained on the handwritten character A-Z training data for 100 epochs and validated on the test data. The auto-encoder achieved a loss of 0.0814 and a validation loss of 0.0815, as shown in the training graph in Figure 10. To evaluate the auto-encoder's performance, five random test images were selected and tested, with the results displayed in Figure 11.

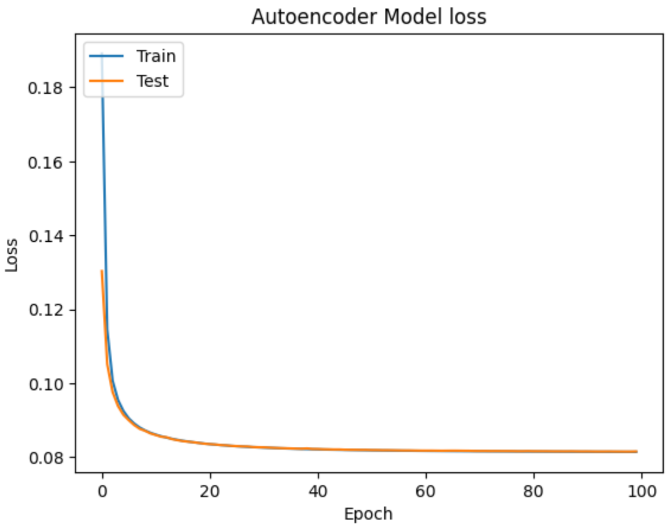

Figure 10. Auto-encoder result of handwriting character A-Z model loss

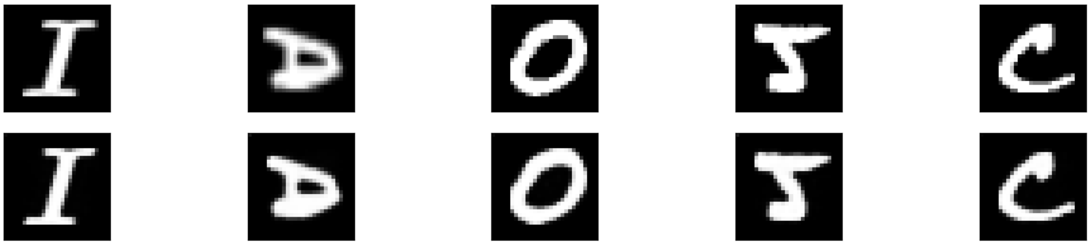

Figure 11. Auto-encoder result of handwriting character A-Z

For the downstream task, we conducted experiments on both clustering and classification using a dataset labeled A to Z. The clustering method employed was similar to what was implemented on a previous dataset, utilizing k-means and agglomerative clustering with 26 clusters corresponding to the alphabet size. However, due to the mixed feature distribution in the dataset, it posed challenges for the clustering algorithms to create accurate clusters.

The clusters generated by k-means are illustrated in Figure 12, while those from agglomerative clustering are shown in Figure 13. Subsequently, when these clusters were tested on five randomly chosen test images, both k-means and agglomerative clustering produced inaccurate results. This outcome is presented in Figure 14 and Figure 15.

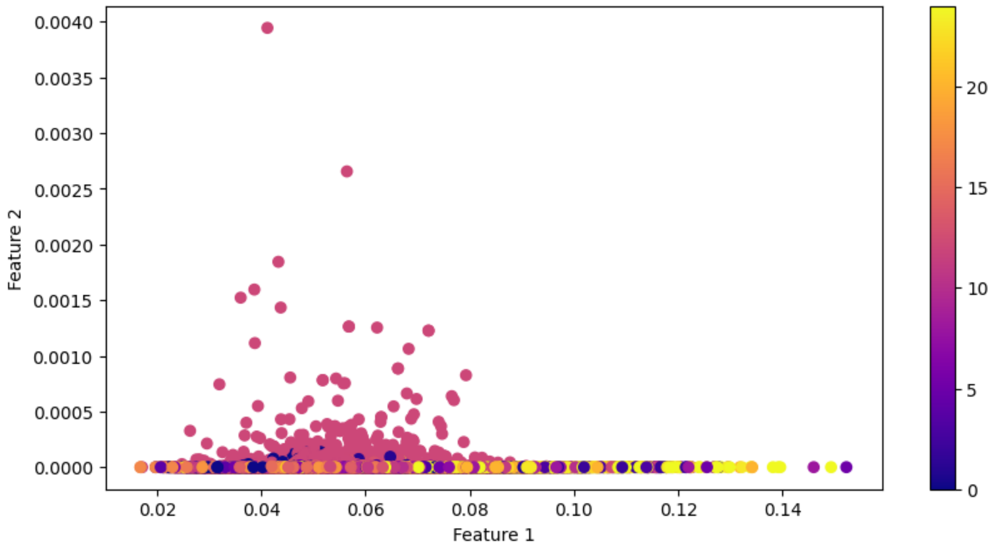

Figure 12. Clustering distribution of handwritten character A-Z using K-mean

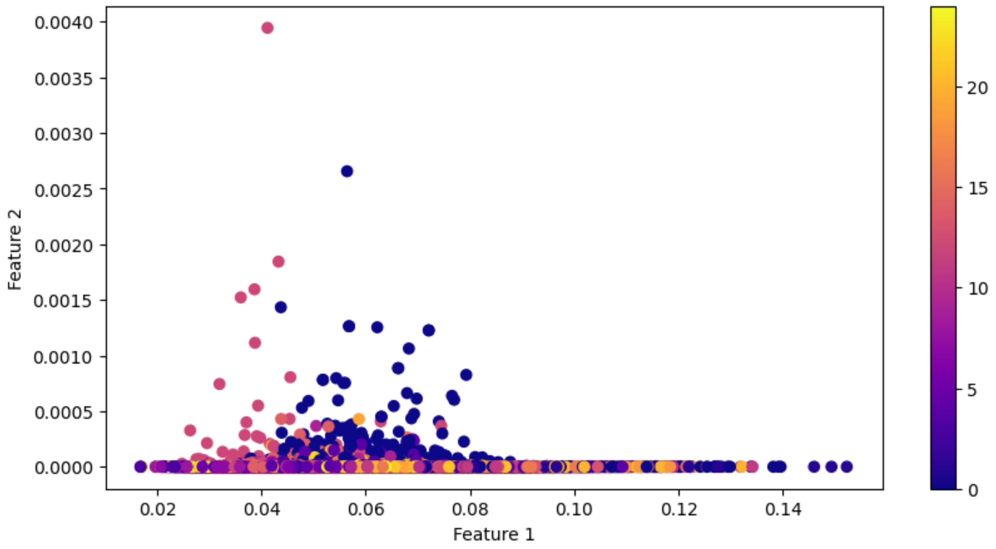

Figure 13. Clustering distribution of handwritten character A-Z using agglomerative

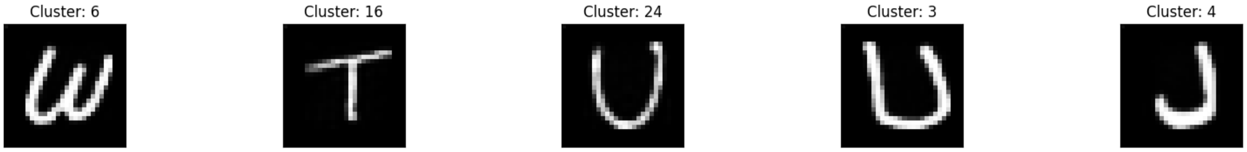

Figure 14. Clustering result of handwriting character A-Z using k-mean

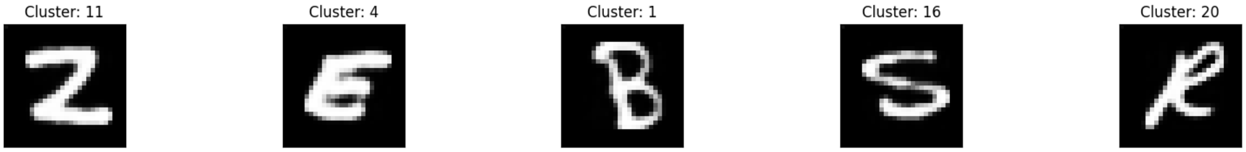

Figure 15. Clustering result of handwriting character A-Z using agglomerative

The other experiment is using classification with input from result of auto-encoder. This classification was implemented using a multi-layer neural network to classify the handwritten characters A-Z. The classifier consists of an input layer with 784 neurons, three hidden layers with 256, 64, and 32 neurons respectively, and an output layer with 26 neurons. The model was compiled using the Adam optimizer and binary cross-entropy loss. It was trained on the handwritten character A-Z training data and validated on the test data. The training results showed a loss of 0.00092, an accuracy of 0.9974, a validation loss of 0.0062, and a validation accuracy of 0.9909, indicating a classification error rate of 0.91%. To further verify the model's performance, 10 random test images were selected and predicted using the model, with the prediction results shown in Figure 16.

Based on the results of the auto-encoder for reconstructing images and the multi-layer neural network for classification, it can be concluded that this approach is both accurate and precise for predicting handwritten digits (0-9, MNIST) and handwritten characters (A-Z).

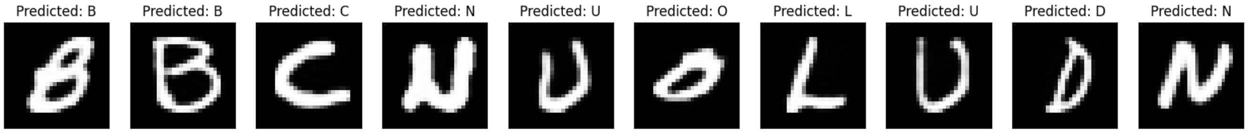

Figure 16. Classification result of handwriting character A-Z using multi-layer neural network

The conclusion of this experiment is by the implementation of auto-encoders on the MNIST and Handwriting A-Z datasets for clustering and classification tasks. An auto-encoder was successfully trained on the MNIST dataset, achieving low reconstruction error, but clustering results using K-means and agglomerative methods were unsatisfactory due to uniform data distribution. In contrast, a multi-layer neural network for classification on the MNIST dataset achieved high accuracy 0.9987 and low error rates, significantly outperforming clustering methods. Similarly, for the Handwriting A-Z dataset, the neural network achieved a classification accuracy of 0.9974 with an error rate of 0.91%, demonstrating the robustness of the approach. The study confirms the effectiveness of auto-encoders as a preprocessing step, enhancing classification performance across different datasets and validating the combined use of auto-encoders and neural networks for accurate handwritten recognition.