Clustering MNIST Digits Using Deep Autoencoders and Deep Embedded Clustering (DEC)

Magferat Jahan Diba
ID : 24241113
CSE425 Project Report
BRAC University
magferat.jahan.diba@g.bracu.ac.bd

***Abstract*—In this paper, we explore unsupervised learning approaches for image clustering using deep neural networks. We focus on the MNIST Digit dataset and implement two models: a convolutional autoencoder combined with K-Means clustering, and Deep Embedded Clustering (DEC). The objective is to transform image data into meaningful latent representations that allow effective clustering without labels. We evaluated cluster quality using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. Our results demonstrate that the DEC model achieves better cluster separation and compactness than the standard Autoencoder + KMeans pipeline. Visualizations using t-SNE further validate the clustering structure learned by each model. This study highlights the power of deep unsupervised methods in learning discriminative features for image clustering.**

***Index Terms*—clustering, mnist, mnist digit, autoencoder, k-means, dec, unsupervised learning**

I. Introduction

Clustering is a fundamental unsupervised learning task where the goal is to group similar data points without predefined labels. Traditional clustering methods like K-Means and DBSCAN operate directly on raw input data, which may not be optimal for complex datasets like images. With the success of deep learning in supervised tasks, researchers have explored using neural networks to learn lower-dimensional, more meaningful representations suitable for clustering. In this work, we explore two deep clustering methods on the MNIST dataset: a convolutional Autoencoder followed by KMeans, and Deep Embedded Clustering (DEC), which incorporates clustering loss directly into training. We aim to assess and compare their performance, analyze learned representations, and understand their advantages and limitations.

II. Related Work

Traditional clustering techniques like KMeans or DBSCAN rely on distance metrics in the original input space. However, high-dimensional data (such as images) often reside in lower-dimensional manifolds. Autoencoder-based clustering addresses this by learning compressed features. Deep Embedded Clustering [1] combines feature learning and clustering in a unified framework. Other methods like Siamese and Triplet networks use contrastive losses but require some supervision. Our work focuses on two unsupervised variants: Autoencoder + KMeans and DEC.

III. Dataset Analysis

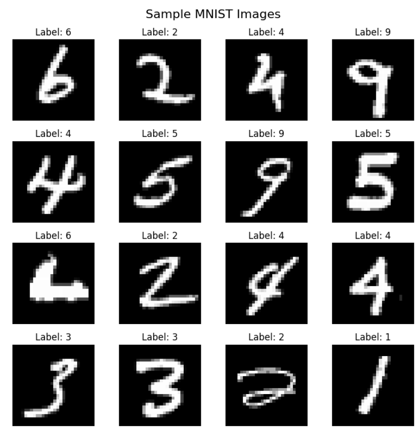

Fig. 1. Sample MNIST digits (individual images)

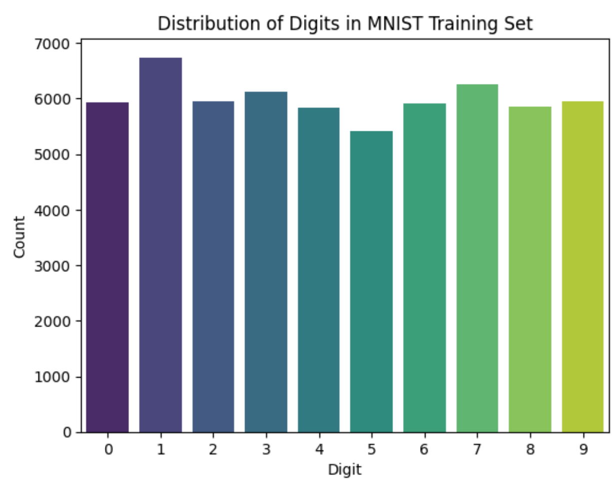

Fig. 2. MNIST digit distribution in training set

We use the MNIST dataset containing 70,000 grayscale images of handwritten digits (0-9), each sized 28x28 pixels. For training, we use the 60,000 training images, and the remaining 10,000 are reserved for testing and visualization. Each image is normalized to [-1, 1] to accelerate training. No labels are used during training to simulate a purely unsupervised setting.

IV. Methodology

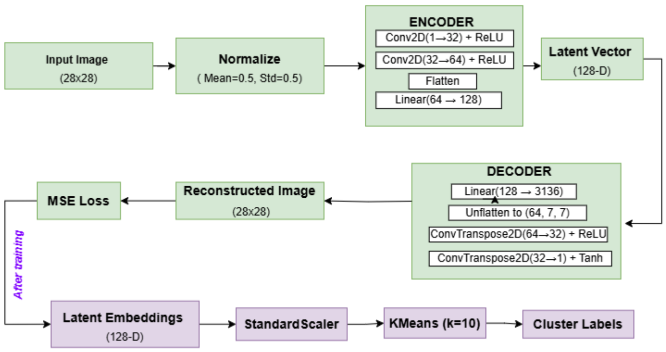

Fig. 3. Autoencoder Architecture

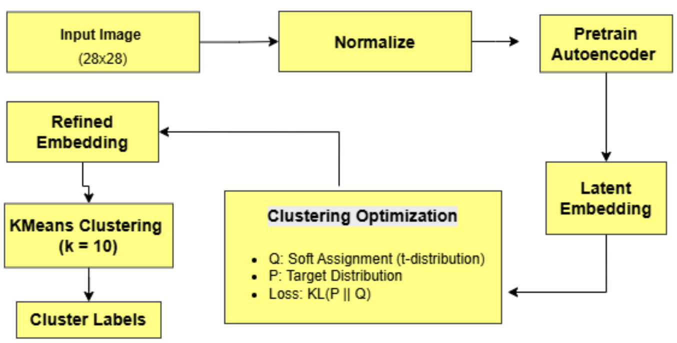

Fig. 4. Deep Embedded Clustering (DEC)

A. *Autoencoder + KMeans*

We use a convolutional autoencoder composed of:

- Encoder: Conv2D layers reducing spatial dimensions, followed by a fully connected layer to produce a latent embedding (e.g., 10D).
- Decoder: Mirrors the encoder using ConvTranspose2D layers.
- Loss Function: Mean Squared Error (MSE) for image reconstruction.

After training, we extract latent embeddings from the encoder and apply KMeans clustering with k=10.

B. *Deep Embedded Clustering (DEC)*

DEC modifies the autoencoder by removing the decoder after pretraining and optimizing a clustering objective:

- Cluster centers are initialized via KMeans.
- Soft assignments are computed using Student's t-distribution.
- Loss: DEC minimizes the KL divergence between soft cluster assignments q (using Student's t-distribution) and a target distribution p that emphasizes confident predictions, thereby refining the cluster boundaries iteratively.

DEC combines unsupervised representation learning and clustering in a single model.

C. *Hyperparameters and Regularization*

Table I summarizes the key training hyperparameters used for both the Autoencoder + KMeans pipeline and the Deep Embedded Clustering (DEC) model.

TABLE I
Training Hyperparameters for Autoencoder + KMeans and DEC

Hyperparameter	**Autoencoder + KMeans**	**DEC**
Latent Dimension	10	10
Batch Size	256	256
Optimizer	Adam (lr = 0.001)	Adam (lr = 0.001)
Training Epochs	103	30 (pretraining) + 103 (clustering)

We did not use explicit regularization techniques (e.g., dropout, batch normalization) or data augmentation in either model. These can be explored in future extensions.

V. Evaluation Results

We evaluate using three clustering metrics:

Model	***Silhouette***	***DB Index***	***CH Index***
Autoencoder + KMeans	0.42	1.92	4985.74
DEC	0.78	0.28	251805.08

DEC shows significantly better compactness and separation. Both methods used the same encoder structure, which highlights the benefit of incorporating clustering loss during training.

VI. Visualization

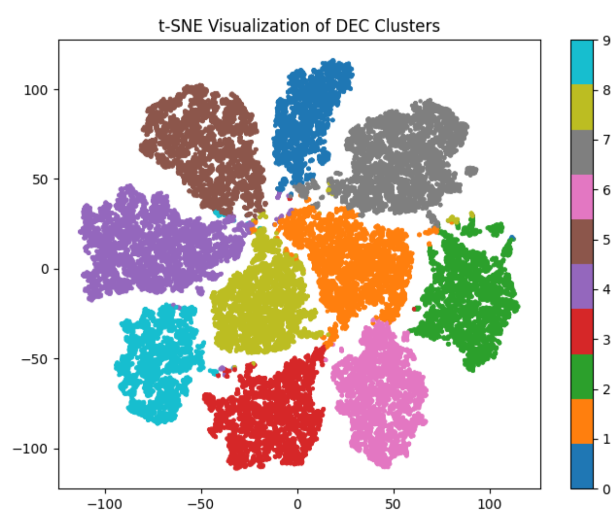

Fig. 5. t-SNE Visualization (DEC)

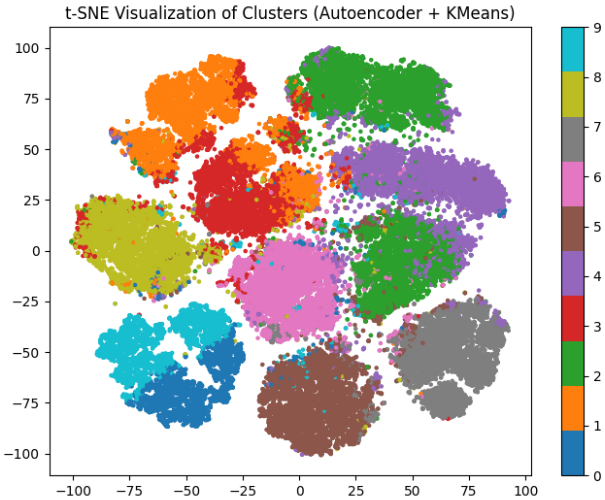

Fig. 6. t-SNE Visualization (Autoencoder + KMeans)

We used t-SNE to project 10D embeddings to 2D for visualization. DEC shows more clearly defined and tighter clusters compared to the raw autoencoder embeddings.

VII. Limitations And Challenges

- DEC requires careful initialization; poor cluster centers can destabilize training.
- t-SNE is non-deterministic and cannot be used as a quantitative metric.
- No regularization or data augmentation used; results may vary on noisier data.
- Evaluation depends on label alignment; true unsupervised evaluation remains challenging.

VIII. Conclusion

This study implemented and compared two unsupervised deep clustering methods on MNIST. We find that DEC, through joint representation learning and clustering optimization, produces better quality clusters than Autoencoder + KMeans. These results suggest that task-specific clustering objectives enhance deep clustering performance. Future work could explore attention-based encoders, other datasets like Fashion-MNIST, and contrastive learning methods.

References

[1] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," 2016. [Online]. Available: https://arxiv.org/abs/1511.06335