Yash Bansal (2022CS51133) Nikhil Zawar (2022CS11106)

COL333 Assignment 3: Part 2

1 Introduction

This report illustrates the architecture, results and various optimizations considered for the model to recognize unlabelled data and cluster them, The VAE has two main parts: an encoder, which compresses the image into a smaller set of values (mean and variance) that represent the main features of the image. A decoder, which takes these values from the latent space and tries to reconstruct the original image. To make the VAE more useful, we apply a Gaussian Mixture Model (GMM) to the latent values. The GMM groups similar images together into clusters by finding patterns in the latent space. After training, the VAE and GMM together can classify images by assigning each one to a cluster. We check the quality of the model by looking at the accuracy of these classifications and by comparing how close the reconstructed images are to the originals using metrics like Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM).

We have achieved validation accuracy of around $\tilde{9}7\%$ for 80-20 train-validation split on the given training dataset.

2 Final Accuracy, Loss, SSIM Plots

Both the training and test MSE decrease with increasing epochs, which is expected as the model improves its predictions. The MSE stabilizes toward the later epochs, showing that the model is likely converging. The SSIM metric increases steadily for both training and test sets, which suggests that the generated images become structurally more similar to the originals as training progresses.

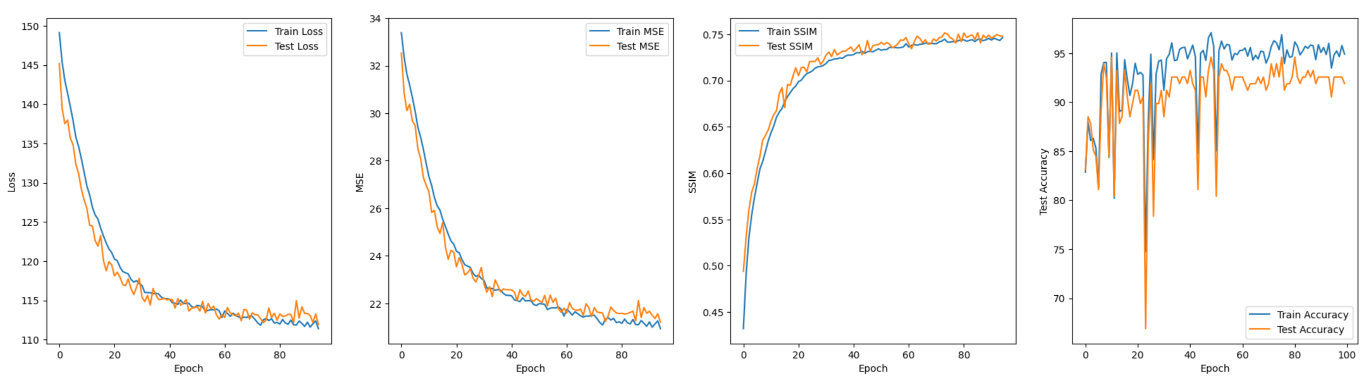


Figure 1: Plot for the final model used

3 VAE Architecture

- **Input image size:-** 28×28 (Flattened to 784)
- **Batch size:-** 128

- **Loss function:-** Binary Cross Entropy Loss combined with Kullback-Leibler (KL) divergence loss
- **Optimizer:-** Adam optimizer with initial learning rate 0.001
- **Latent dimension:-** 2
- **Total parameters:-** Approx. 810012

Layer	**Output Size**	**Layer Details**
Common FC1	400	Linear(784, 400), BatchNorm1d(400), Tanh
Common FC2	196	Linear(400, 196), BatchNorm1d(196), Tanh
Common FC3	48	Linear(196, 48), BatchNorm1d(48), Tanh
Mean FC	2	Linear(48, 16), BatchNorm1d(16), Tanh, Linear(16, 2)
Log Variance FC	2	Linear(48, 16), BatchNorm1d(16), Tanh, Linear(16, 2)
Latent Sampling	2	Sampled from Gaussian distribution based on mean and variance
Decoder FC1	16	Linear(2, 16), BatchNorm1d(16), Tanh
Decoder FC2	48	Linear(16, 48), BatchNorm1d(48), Tanh
Decoder FC3	196	Linear(48, 196), BatchNorm1d(196), Tanh
Decoder FC4	400	Linear(196, 400), BatchNorm1d(400), Tanh
Output Layer	784	Linear(400, 784), Sigmoid
Output Reshape	$1 \times 28 \times 28$	Reshape to original image dimensions

Table 1: VAE Architecture Description

4 GMM Implementation

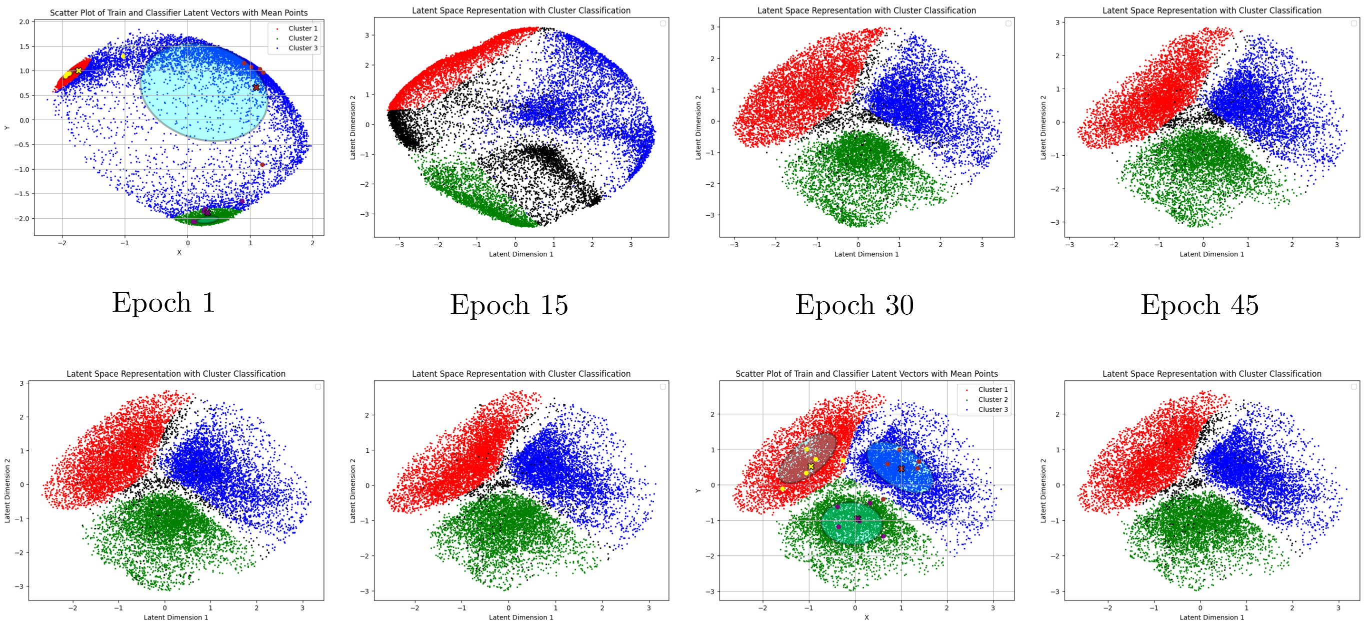
The GMM architecture is designed as an Expectation-Maximization (EM) algorithm that iteratively updates the means, covariances, and mixing coefficients of Gaussian distributions representing each cluster.

- **Initial Parameters:** Cluster Means: Initialized as the average latent vector for each class, allowing clusters to begin near known class representations. Covariances: Each cluster is initialized with a small, isotropic covariance matrix (‘0.1 * I‘), where ‘I‘ is the identity matrix. Mixing Coefficients: Set uniformly, so each cluster has an equal initial probability.
- **Expectation-Maximization (EM) Loop :**E-Step(Expectation): Calculates the responsibility of each cluster for every latent vector using the Gaussian probability density function (PDF). It accounts for the distance of each point from the mean of each cluster, normalized by each cluster's covariance.
- **Expectation-Maximization (EM) Loop**: M-Step(Maximization): Updates the GMM parameters by computing the mean vectors, covariances, mixed coeffs which are calculated as the weighted average of latent vectors.
- **Convergence:** The loop stops if the relative change in means, covariances, and mixing coefficients falls below a defined tolerance (‘tol = 1e-15‘), or a maximum iteration limit (1000) is reached.

5 Visualization & Clustering analysis

The latent clusters and their ellipses are visualized in a scatter plot with 3 colos - red, green, blue - to distinguish different clusters. Ellipses are drawn around clusters to show covariance structures, indicating the extent and orientation of each Gaussian component.

Points from the classifier dataset are overlaid on the plot, marked with unique colors for different labels. Mean points for each class are also plotted using a distinct marker (‘X‘) for visual clarity.



Epoch 1 | Epoch 15 | Epoch 30 | Epoch 45

Epoch 60 | Epoch 85 | Final Ellipse | Final Clusters

Figure 2: Cluster evolution at different epochs

6 Manifold

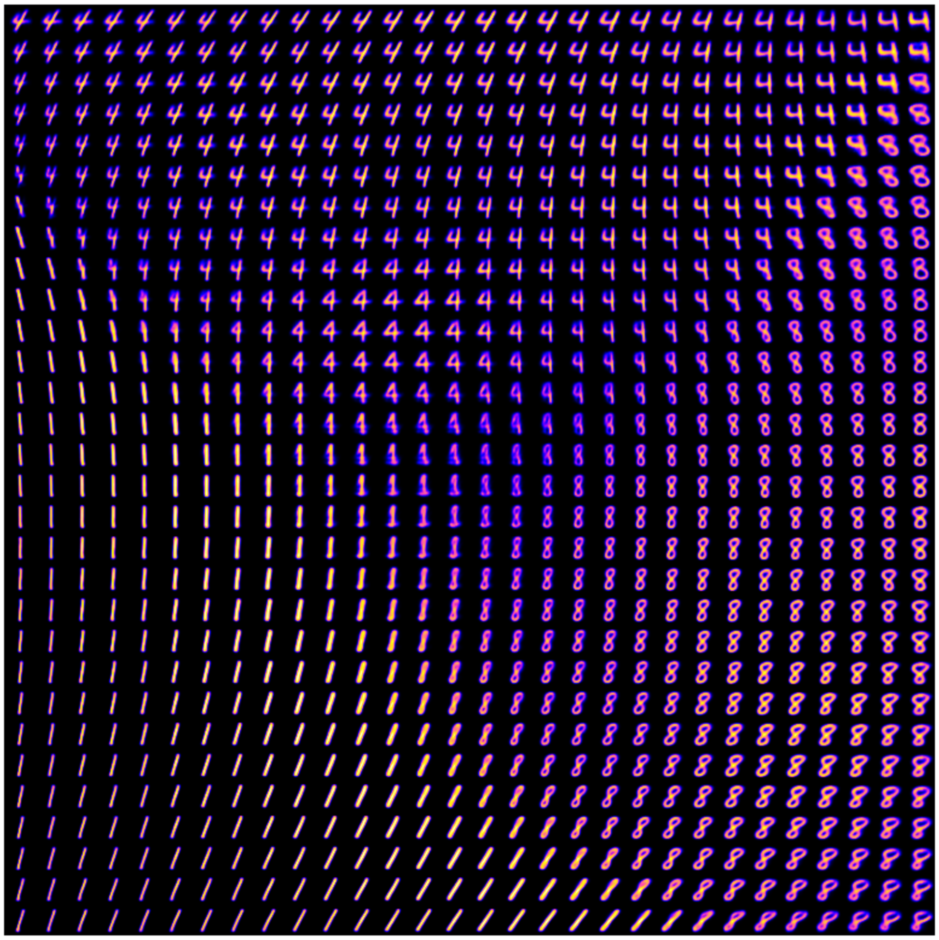


Figure 3: Manifold of the Latent Space Distribution