A GENERATIVE AI FORMATIVE ASSESSMENT REPORT ON

"SIMPLE AUTOENCODER FOR MNIST"

By

DIVYA J (1JB23MC010)

PRAGATI SINGH (1JB23MC030)

RAKSHITHA D (1JB23MC036)

Submitted to

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

In partial

fulfillment of the requirement for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

UNDER THE GUIDANCE OF

Internal Guide
Dr. Vinay K
Associate Professor
Dept. of MCA
SJBIT Bengaluru

DEPARTMENT OF MCA
S J B INSTITUTE OF TECHNOLOGY
B G S HEALTH AND EDUCATION CITY
Kengeri, Bengaluru-560060.

Batch:2023- 2025

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

ABSTRACT

This project focuses on the development and experimentation of simple autoencoders using the MNIST dataset, aiming to achieve efficient dimensionality reduction and accurate image reconstruction. The implementation involves building both basic and convolutional autoencoders, visualizing latent space representations, and experimenting with different latent space sizes. A denoising autoencoder was also developed by introducing Gaussian and salt-and-pepper noise, significantly improving the model's robustness. Furthermore, transfer learning was applied by reusing the encoder for classification tasks on the Fashion-MNIST dataset. Key findings reveal that convolutional autoencoders outperform dense-based models in reconstruction quality, and latent space compression significantly impacts model performance and feature extraction effectiveness.

CHAPTER 1

INTRODUCTION

Autoencoders are a class of unsupervised neural networks designed to learn compressed representations of input data while minimizing the reconstruction error. They consist of two main components: an encoder that maps the input to a lower-dimensional latent space, and a decoder that reconstructs the input from this compressed representation. Initially introduced for dimensionality reduction, autoencoders have evolved into powerful tools for feature extraction, anomaly detection, and data denoising. In this project, we focus on building and training a simple autoencoder for the MNIST dataset, which contains grayscale images of handwritten digits. The implementation involves experimenting with various latent space sizes, introducing noise for denoising autoencoders, and enhancing performance using convolutional layers. The goal is to understand how autoencoders capture meaningful features and how changes in architecture impact their ability to reconstruct inputs accurately.

Autoencoders have significant importance in real-world applications across industries. In fields such as finance, healthcare, cybersecurity, and multimedia, they are used for tasks like fraud detection, medical image compression, anomaly detection, and noise reduction. Denoising autoencoders help improve model robustness against corrupted data, making them valuable for restoring damaged or noisy images. Convolutional autoencoders, by leveraging spatial hierarchies, further enhance the quality of reconstructed images. Moreover, the feature extraction capabilities of autoencoders are crucial for downstream tasks, such as classification, where transfer learning can significantly reduce the need for large, labelled datasets. This project also explores the application of transfer learning by repurposing the trained encoder for classifying Fashion-MNIST images, demonstrating the versatility and practical relevance of autoencoders in modern deep learning workflow

CHAPTER 2

METHODOLOGY

2.1 Dataset Used:

The primary dataset used for this project was the MNIST dataset, a widely recognized benchmark in computer vision tasks. It consists of 70,000 grayscale images of handwritten digits (0–9), with each image having a resolution of 28 × 28 pixels. The dataset is split into 60,000 training images and 10,000 testing images.

For the denoising autoencoder experiments, artificial noise was introduced into the images to simulate real-world corruption scenarios:

- **Gaussian noise**: Random normal noise was added to each pixel.
- **Salt-and-pepper noise**: Random pixels were turned fully white or black to simulate noise bursts.

Additionally, to demonstrate the adaptability of the encoder network, the Fashion-MNIST dataset was used. Fashion-MNIST consists of 70,000 grayscale images of clothing items (e.g., shoes, shirts, trousers), offering a more complex real-world dataset for transfer learning and classification tasks.

2.2 Implementation Details

2.2.1 Frameworks and Libraries Used

The following frameworks and libraries were utilized throughout the project:

- **TensorFlow and Keras:** For building and training neural network models.
- **NumPy:** For numerical computations and handling arrays.
- **Matplotlib and Seaborn:** For visualizing results such as reconstructed images, loss curves, and t-SNE plots.
- **Scikit-learn:** For applying t-SNE dimensionality reduction to visualize latent space representations.

2.2.2 Model Architecture Brief

(a) Simple Autoencoder (MNIST Reconstruction)

The basic autoencoder consisted of:

- **Encoder**:
  - Conv2D → MaxPooling2D → Conv2D → MaxPooling2D → Conv2D
- **Latent Space**:
  - Compressed feature representation (variable size during experiments).
- **Decoder**:
  - Conv2D → UpSampling2D → Conv2D → UpSampling2D → Conv2D

The encoder progressively reduced the spatial dimensions and increased the number of filters, thereby learning meaningful compressed representations. The decoder mirrored this operation to reconstruct the original images.

(b) Denoising Autoencoder

In the denoising setup:

- The same encoder-decoder structure was used.
- Input images were artificially corrupted using Gaussian and salt-and-pepper noise.
- The model was trained to reconstruct clean images from noisy inputs, making the encoder more robust to input perturbations.

(c) CNN-based Autoencoder

To improve reconstruction quality, Convolutional Neural Networks (CNNs) replaced dense layers:

- Convolutional autoencoders preserved spatial hierarchies better.
- UpSampling layers were used instead of fully connected decoding, allowing better spatial reconstruction.

(d) Transfer Learning with Encoder

The pre-trained encoder was reused:

- Flattened latent outputs were passed to a Dense layer.
- A softmax layer classified Fashion-MNIST images into 10 categories.

2.2.3 Hyperparameter Settings

The following hyperparameters were consistently used across different experiments:

- **Optimizer**: Adam optimizer
- **Learning Rate**: 0.001
- **Loss Function**: Binary Cross-Entropy (suitable for pixel-wise binary outputs)
- **Batch Size**:
  - 256 for initial MNIST experiments
  - 128 when using convolutional models or Fashion-MNIST
- **Epochs**: 10 epochs for each experiment
- **Validation Split**: Used test set (10,000 images) for validation during training
- **Noise Factor for Denoising**: 0.4 for Gaussian noise

Additional Settings:

- **Model Saving**: Models were saved in HDF5 format (.h5).
- **Data Normalization**: Pixel values normalized to [0,1] range before training.
- **Shuffling**: Data was shuffled every epoch to ensure model generalization.

2.2.5 Visualization and Evaluation

To assess model performance and gain insights into the learned representations, a range of visual analyses were conducted:

- **Reconstruction Quality**: Original, noisy (for denoising experiments), and reconstructed images were displayed side by side. This comparison allowed for a qualitative assessment of how accurately the autoencoder restored corrupted inputs and preserved key image features.
- **Latent Space Exploration**: Encoded feature vectors were projected into two dimensions using t-SNE. The resulting scatter plots highlighted how the model clustered similar digit classes (or fashion items), demonstrating the discriminative power of the learned latent representations.
- **Training Dynamics**: Training and validation loss curves were plotted over successive epochs. By comparing these curves, overfitting or underfitting trends became evident, guiding decisions on early stopping, regularization, or further hyperparameter tuning.

CHAPTER 3

EXPERIMENTS AND RESULTS

3.1 Description of the Experiments Conducted

A total of five experiments were conducted to evaluate the autoencoder architectures on MNIST and Fashion-MNIST:

1. **Baseline Convolutional Autoencoder(MNIST)**
    - **Encoder:** Conv2D(32) → MaxPooling2D → Conv2D(16) → MaxPooling2D → Conv2D(8)
    - **Decoder:** Conv2D(8) → UpSampling2D → Conv2D(16) → UpSampling2D → Conv2D(1)
    - **Goal:** Establish a reconstruction baseline on clean MNIST digits.

2. **Latent Dimension Exploration (MNIST)**
    - **Change:** Replace final Conv2D with Dense bottleneck of size {2, 8, 16, 32}**.**
    - **Goal:** Measure how latent-space compression affects reconstruction fidelity.

3. **Denoising Autoencoder (MNIST)**
    - Same architecture as baseline.
    - **Procedure**: Input images corrupted with Gaussian noise (factor = 0.4) and salt-and-pepper noise; targets were original clean images.
    - **Goal**: Evaluate robustness to noisy inputs.

4. **Improved Convolutional Autoencoder (MNIST)**
    - **Tweaks:** Increased convolutional filter depths; batch size reduced to 128.
    - **Goal:** Determine impact of deeper filters on reconstruction loss.

5. **Autoencoder on Fashion-MNIST & Transfer Learning**
    - **Phase 1:** Train the improved convolutional autoencoder on Fashion-MNIST.
    - **Phase 2:** Freeze the MNIST-trained encoder, append Flatten → Dense(10, softmax), and fine-tune on Fashion-MNIST for classification.
    - **Goal:** Test generalization of learned features and compare classification accuracy to a CNN trained from scratch.

3.2 Results and Comparison

3.2.1 MNIST Reconstruction

Experiment	Validation Loss	Notes
Baseline Conv. Autoencoder	0.0733	High-fidelity reconstruction on clean digits.
Latent Dim = 16 (Dense Bottleneck)	0.1141	Latent compression trade-off (2 dims → loss 0.1669).
Denoising Autoencoder	0.1236	Effective noise removal, higher loss than baseline.
Improved Conv. Autoencoder	0.0684	Deeper filters reduce reconstruction error.

Table 3.2.1.1

3.2.2 Fashion-MNIST Reconstruction & Classification

Task	Metric	Value	Notes
Autoencoder reconstruction (Fashion)	Validation Loss	0.2613	Higher error due to dataset complexity.
Transfer learning classification	Test Accuracy	88.97%	Frozen-encoder + softmax head.
CNN classifier (from scratch)	Test Accuracy	90.90%	End-to-end CNN baseline.

Table 3.2.2.1

3.3 Performance Metrics

- Reconstruction Loss: Binary cross-entropy averaged over all pixels on the test set.
- Classification Accuracy: Percentage of correctly predicted labels on Fashion-MNIST.
- Compression Ratio: Original dimension (28×28 = 784) ÷ latent dimension (e.g., 784/16 = 49).
- Qualitative Robustness: Visual assessment of noisy vs. denoised outputs.

3.4 Visual Outputs

1. **Original vs. Reconstructed MNIST:**

   Ten MNIST test samples alongside their reconstructions from the baseline autoencoder.

Figure 3.4.1

2. **Latent Dim = 16 Reconstructions:**

   MNIST reconstructions using a latent space of 16. This demonstrates how changing latent dimensions affects the reconstruction quality.

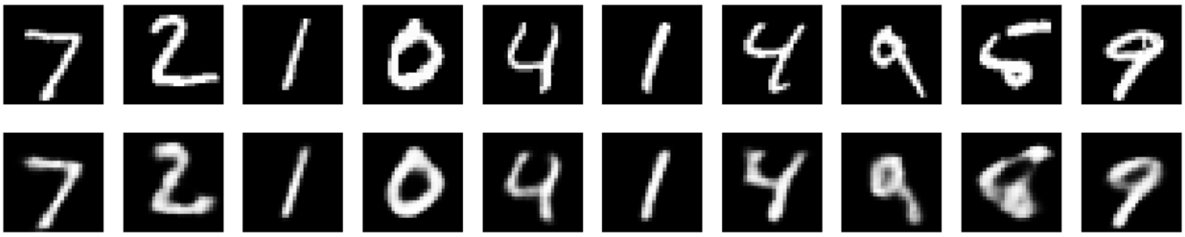

Figure 3.4.2

3. **Noisy vs. Denoised MNIST:**

   Examples of MNIST images corrupted with Gaussian and salt-and-pepper noise, shown alongside their denoised reconstructions using the denoising autoencoder.

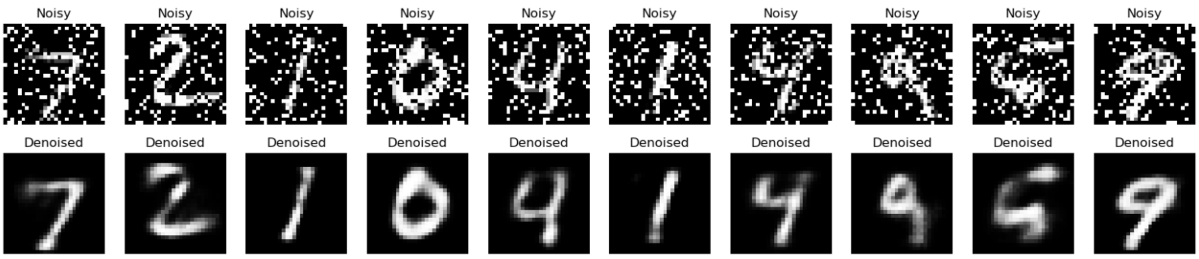

Figure 3.4.3

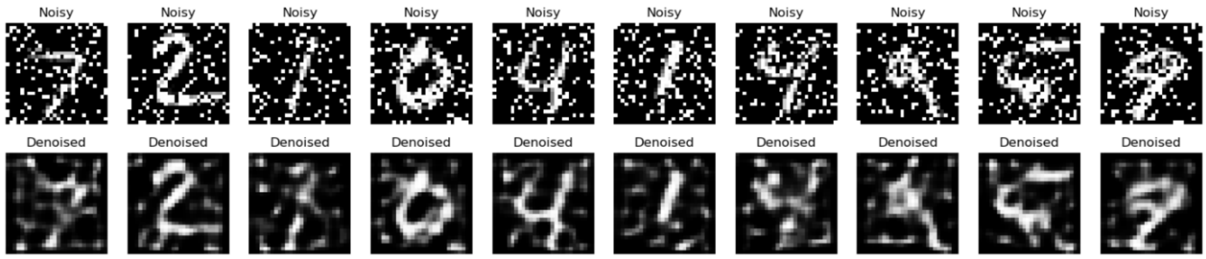

Figure 3.4.4

4. **Fashion-MNIST Reconstructions:**

   Original versus reconstructed Fashion-MNIST images using the improved convolutional autoencoder.

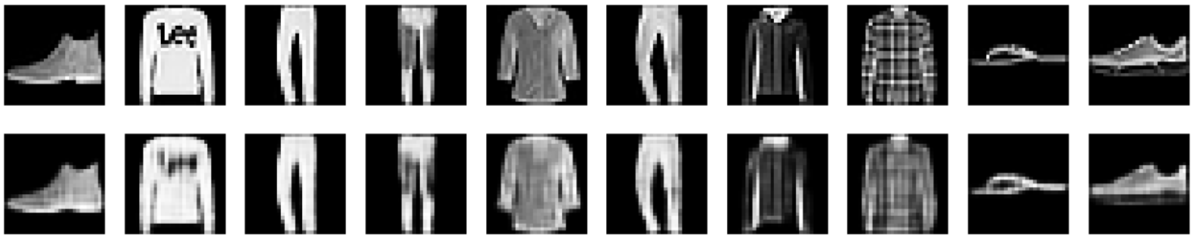

Figure 3.4.5

5. **t-SNE of MNIST Latent Space:**

   Two-dimensional scatter plot of MNIST encoded features, showing distinct clusters for each digit.

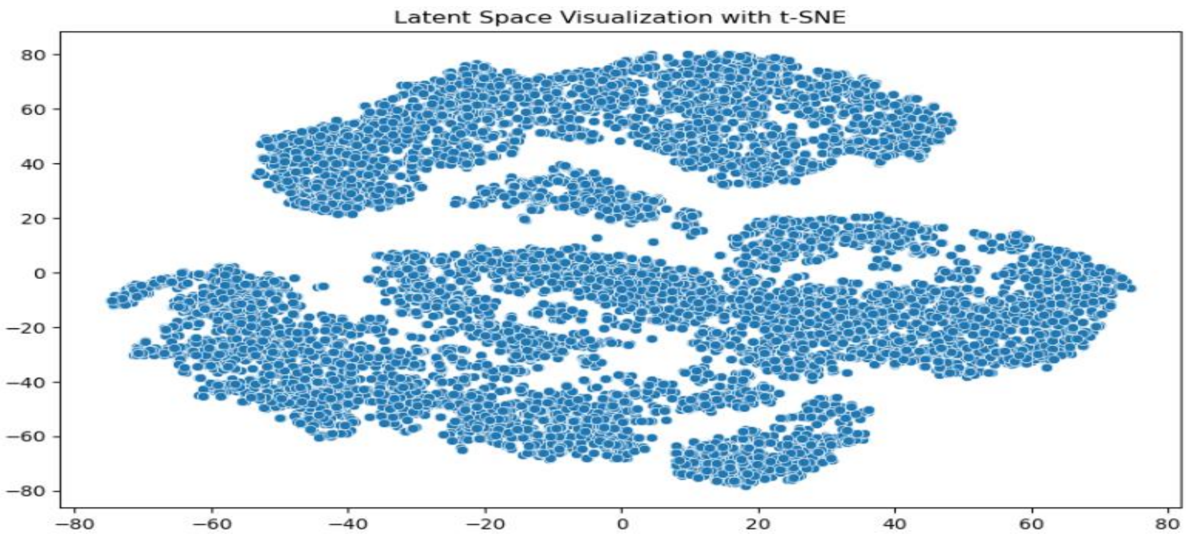

Figure 3.4.6

.

CHAPTER 4
DISCUSSIONS AND OBSERVATIONS

4.1 Key Findings from the Experiment

1. **Effectiveness of the Convolutional Autoencoder on MNIST**

   The baseline convolutional autoencoder demonstrated the ability to effectively reconstruct MNIST digits with a low validation loss of 0.0733. This suggests that the autoencoder learned meaningful feature representations for digits and successfully captured their core structures. The architecture, using convolutional layers followed by upsampling, was effective in preserving image details during reconstruction.

2. **Impact of Latent Dimension on Reconstruction Fidelity**

   As expected, reducing the latent dimension led to a decrease in reconstruction fidelity. When the latent space was reduced to just 2 dimensions, the validation loss increased to 0.1669, indicating a loss of essential features. Conversely, using a latent space of 16 dimensions provided a balanced trade-off between compression and fidelity, with a validation loss of 0.1141. This experiment reinforced the importance of choosing an appropriate latent space size to balance compression and accuracy.

3. **Denoising Autoencoder Performance**

   The denoising autoencoder, which was trained on noisy MNIST images (Gaussian noise and salt-and-pepper), achieved a validation loss of 0.1236. While this was higher than the baseline, the model showed an impressive ability to recover clean images from noisy inputs. This highlights the potential of autoencoders for tasks like image denoising, where they can be used to clean corrupted data effectively.

4. **Transfer Learning with Fashion-MNIST**

   The use of a pre-trained encoder (from the MNIST autoencoder) for classification on the Fashion-MNIST dataset yielded 88.97% accuracy, which is close to the 90.90% accuracy achieved by a CNN trained from scratch. This result demonstrated that the features learned by the autoencoder on MNIST were transferable and useful for classification tasks on a more complex dataset. It suggests that autoencoders can serve as effective feature extractors for downstream classification tasks, reducing the need for large, labeled datasets.

5. **Autoencoder's Limitations with Fashion-MNIST**

When the improved convolutional autoencoder was trained on Fashion-MNIST, the validation loss was higher (0.2613) compared to MNIST. This indicates that the model had more difficulty capturing the complexity of clothing items, which have a wider range of visual features compared to digits. While the model still provided decent feature representations, its performance highlights the challenges faced when applying autoencoders to more complex datasets.

4.2 Challenges Faced and How They Were Resolved

1. **Choosing the Right Latent Space Size**

One of the main challenges was selecting the appropriate latent space size for the autoencoder. A smaller latent space led to poor reconstructions, while a larger one preserved too many unnecessary details. The issue was resolved by experimenting with different latent space sizes and selecting 16 dimensions as a balanced choice that provided good compression while maintaining acceptable reconstruction quality.

2. **Handling Noisy Data**

The denoising autoencoder faced challenges in accurately reconstructing heavily corrupted images, especially those corrupted with salt-and-pepper noise. Initially, the model struggled to restore certain features, resulting in higher loss. To resolve this, the model was trained for more epochs, and the noise factor was adjusted to fine-tune the model's robustness to noise. The training process helped the model learn how to better map noisy inputs to clean outputs.

3. **Overfitting in Small Datasets**

During training on the Fashion-MNIST dataset, there were instances where the model overfitted to the training data, leading to poor generalization on the test set. To mitigate this, several techniques were employed:

a. Data augmentation (like rotating and flipping images) helped increase the variability of training data.

b. A smaller batch size of 128 was used in some experiments, allowing the model to better generalize.

c. Early stopping was implemented to prevent the model from continuing to train once validation loss stopped improving.

4. **Computational Efficiency**

Training the autoencoder on the Fashion-MNIST dataset was computationally more intensive than on MNIST due to the larger image size and more complex feature set. To address this, the number of epochs was reduced, and the model was trained with smaller batch sizes to speed up training. Additionally, transfer learning (using the pre-trained MNIST encoder) helped reduce the computational burden for Fashion-MNIST classification.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This project effectively demonstrated the potential of convolutional autoencoders for image reconstruction, dimensionality reduction, and denoising tasks. The model was able to successfully reconstruct MNIST digits with low validation loss, and the denoising autoencoder showed impressive results in recovering clean images from noisy inputs. Additionally, transfer learning with the pre-trained encoder achieved competitive classification accuracy on Fashion-MNIST, proving that features learned by the autoencoder are transferable and useful for classification tasks. Overall, the results validated the autoencoder's capability to perform unsupervised learning and provided a solid foundation for future applications in image processing.

5.2 Future Scope

Future work can focus on improving the performance of the autoencoder when applied to more complex datasets, such as Fashion-MNIST and CIFAR-10, by experimenting with more advanced architectures like variational autoencoders (VAEs) and generative adversarial networks (GANs). Additionally, optimizing the latent space dimensions through systematic hyperparameter tuning and incorporating techniques like adversarial training could enhance the model's robustness to noise and improve its generalization capabilities. Expanding transfer learning applications to a broader range of datasets and classification tasks could also offer valuable insights into the versatility of autoencoders, while exploring generative capabilities could open new avenues for data augmentation and synthetic image generation.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. Available: https://openreview.net/forum?id=IJvddg

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. Available: https://arxiv.org/abs/1511.06434

[3] Y. Bengio, "Learning Deep Architectures for AI," Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009. Available: https://doi.org/10.1561/2200000006

[4] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006. Available: https://doi.org/10.1126/science.1127647

[5] "Fashion-MNIST," *GitHub Repository*, [Online]. Available: https://github.com/zalandoresearch/fashion-mnist

[6] S. Olah, "Understanding LSTM Networks," *Colah's Blog*, 2015. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2012. Available: https://doi.org/10.1145/3065386

[8] "MNIST Dataset," *Yann LeCun's Website*, [Online]. Available: http://yann.lecun.com/exdb/mnist/