Reproduction and Extension of LeNet-5 on MNIST

Project Report for the Course Requirement of
CS 725

Instructor: Prof. Abir De

Team Members:
Ajay Rangoji - 25M0802
Harshith Matta - 25M0834
G Jayavardhan - 25M0807
Vineeth Babu Ch - 25M0808
Pujith Sai Kumar K - 25M0787

November 25, 2025

Abstract

This report details the reproduction of the classic LeNet-5 architecture as proposed by LeCun et al. (1998) and its application to the MNIST handwritten digit classification task. We successfully reproduced the baseline results using the original architecture specifications (Tanh activation, Average Pooling). Furthermore, we extended the study by experimenting with a higher learning rate and a modern configuration using Adam optimizer and ReLU activation. Our results confirm the robustness of the LeNet-5 architecture and demonstrate the efficiency gains from modern optimization techniques.

1 Introduction

The LeNet-5 architecture is a pioneering Convolutional Neural Network (CNN) that laid the foundation for modern deep learning in computer vision. Proposed by Yann LeCun et al. in 1998, it was designed to recognize handwritten digits from the MNIST dataset. The goal of this project is to reproduce the original results and compare them against modern variations.

2 Literature Review & Implementation Details

2.1 Overview of the Original Paper

The seminal paper *"Gradient-Based Learning Applied to Document Recognition"* (1998) by LeCun et al. introduced several key concepts that define modern computer vision:

- **End-to-End Learning**: It demonstrated that feature extraction and classification could be learned simultaneously via gradient descent, replacing hand-crafted heuristics.
- **Convolutional Neural Networks (CNNs)**: It solidified the use of local receptive fields, shared weights, and subsampling (pooling) to achieve translation invariance.
- **Graph Transformer Networks (GTN)**: The paper also extensively discussed GTNs for recognizing strings of characters (e.g., zip codes), treating recognition as finding the optimal path through a graph.

2.2 Implementation Differences

While our implementation reproduces the core architecture and results, we adopted modern standard practices that differ slightly from the specific technical details of 1998:

- **C3 Layer Connections**: The original paper used a specific sparse connection table between S2 and C3 to break symmetry. We used standard dense connections (all-to-all maps).
- **Activation Function**: The paper used a Scaled Tanh function ($1.7159 \cdot \tanh(\frac{2}{3}x)$). We used standard Tanh and ReLU.
- **Output Layer**: The original model used Euclidean Radial Basis Function (RBF) units. We utilized a standard Linear layer with Softmax and Cross-Entropy Loss, which is the modern standard for classification stability.

3 Methodology

3.1 Architecture

We implemented the LeNet-5 architecture with the following layers:

- **C1**: Convolutional Layer (6 filters, 5×5 kernel)
- **S2**: Subsampling Layer (Average Pooling, 2×2, stride 2)
- **C3**: Convolutional Layer (16 filters, 5×5 kernel)
- **S4**: Subsampling Layer (Average Pooling, 2×2, stride 2)
- **C5**: Convolutional Layer (120 filters, 5×5 kernel)
- **F6**: Fully Connected Layer (84 units)
- **Output**: Fully Connected Layer (10 units)

3.2 Data Preprocessing

The MNIST dataset consists of 28×28 grayscale images. As the original LeNet-5 expects 32×32 inputs, we applied padding of 2 pixels on all sides. The pixel values were normalized using the dataset mean (0.1307) and standard deviation (0.3081).

4 Experiments

We conducted three experiments to evaluate the model:

1. **Baseline**: The classic configuration using SGD optimizer (LR=0.01, Momentum=0.9) and Tanh activation functions.
2. **SGD High LR**: Same as baseline but with a higher learning rate of 0.1 to test stability.
3. **Modern**: A modernized version using the Adam optimizer (LR=0.001) and ReLU activation functions.

All models were trained for 10 epochs with a batch size of 64.

5 Results

5.1 Performance Metrics

The table below summarizes the best test accuracy achieved by each configuration.

Experiment	**Optimizer**	**Best Accuracy (%)**
Baseline (Tanh)	SGD (0.01)	99.01
SGD High LR	SGD (0.1)	98.59
Modern (ReLU)	Adam (0.001)	99.01

Table 1: Comparison of Best Test Accuracies

5.2 Training Dynamics

The training loss and test accuracy curves are presented below.

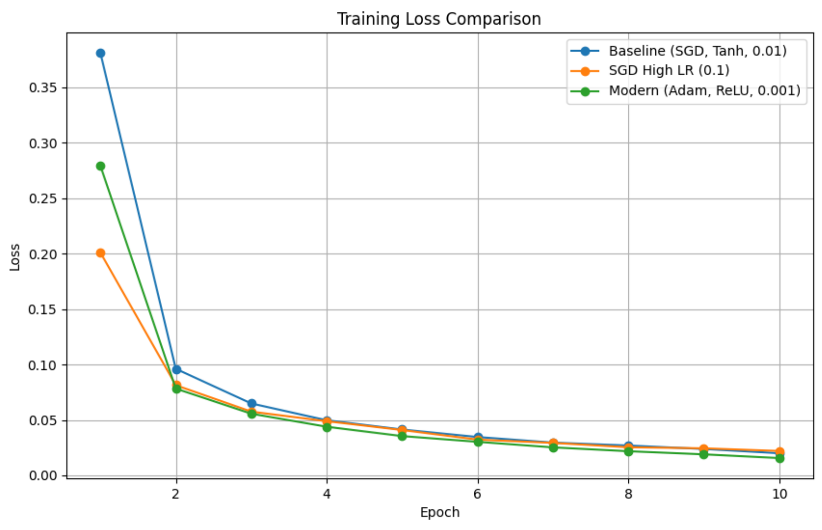


Figure 1: Training Loss over Epochs

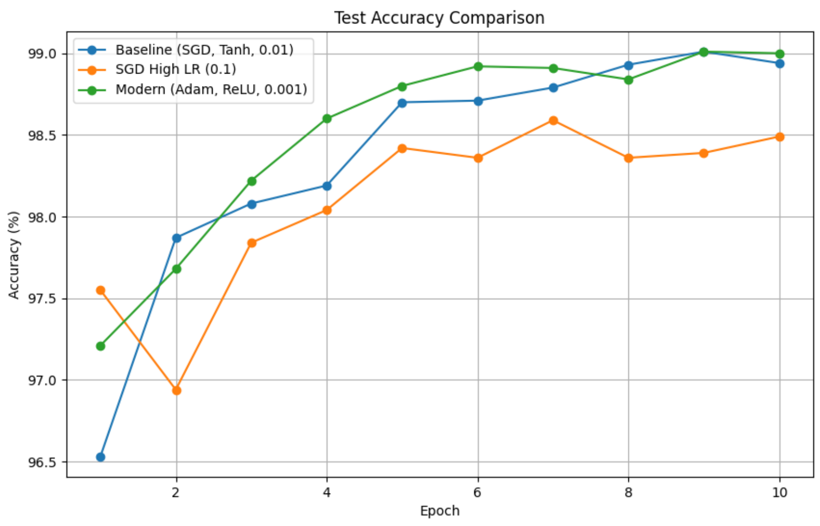
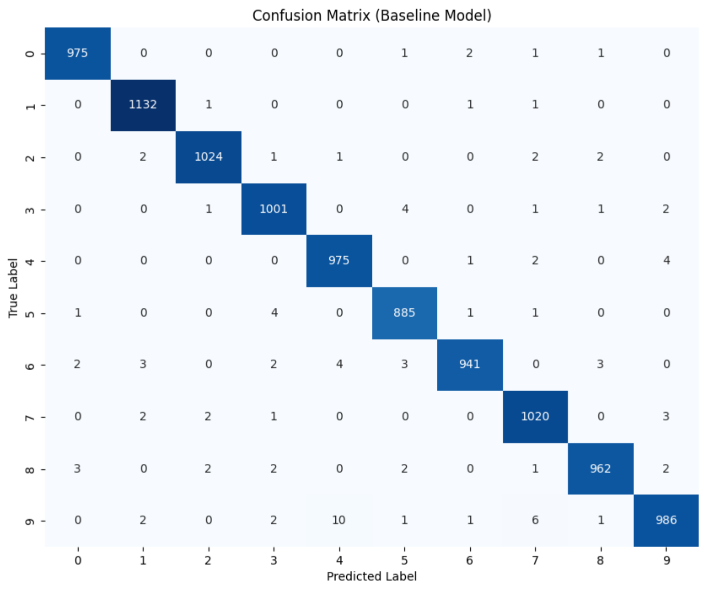


Figure 2: Test Accuracy over Epochs

5.3 Error Analysis

To better understand the model's performance, we analyzed the confusion matrix for the Baseline model (Figure 3). The model shows high precision across all digits, with very few misclassifications.



	0	1	2	3	4	5	6	7	8	9
0	975	0	0	0	0	1	2	1	1	0
1	0	1132	1	0	0	0	1	1	0	0
2	0	2	1024	1	1	0	0	2	2	0
3	0	0	1	1001	0	4	0	1	1	2
4	0	0	0	0	975	0	1	2	0	4
5	1	0	0	4	0	885	1	1	0	0
6	2	3	0	2	4	3	941	0	3	0
7	0	2	2	1	0	0	0	1020	0	3
8	3	0	2	2	0	2	0	1	962	2
9	0	2	0	2	10	1	1	6	1	986

Figure 3: Confusion Matrix for Baseline Model

5.4 Additional Experiments

5.4.1 Comparison with Simple MLP

We compared the LeNet-5 architecture with a simple Multi-Layer Perceptron (MLP) consisting of two hidden layers (256 and 128 units). As shown in Figure 4, LeNet-5 outperforms the MLP, demonstrating the effectiveness of convolutional layers in capturing spatial hierarchies in image data.

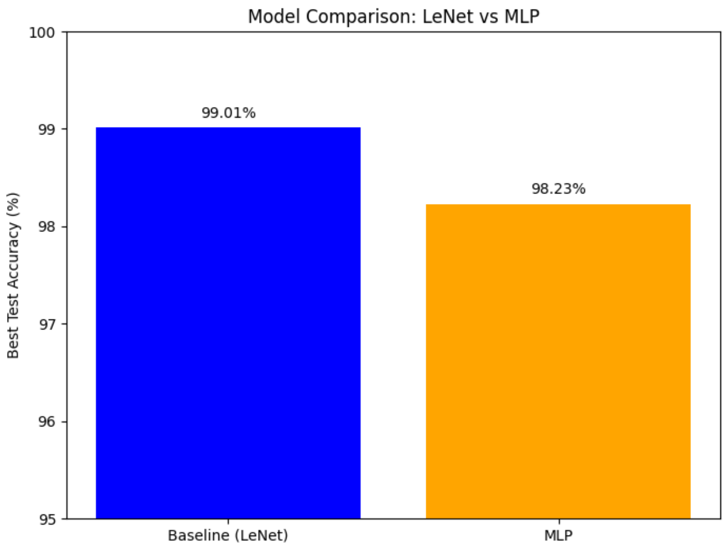


Figure 4: LeNet-5 vs MLP Performance

5.4.2 Effect of Data Augmentation

We introduced data augmentation techniques, specifically random rotations ($\pm 10^\circ$) and slight translations. Figure 5 shows that augmentation slightly improved the model's generalization capability, pushing the accuracy even higher.

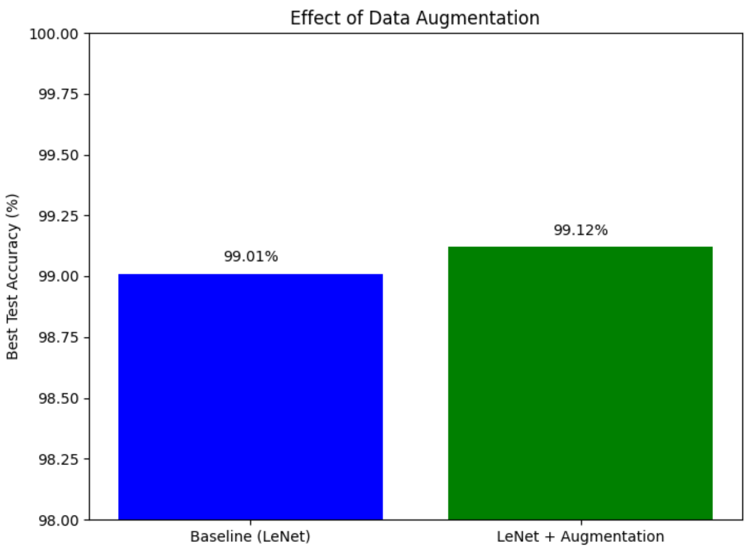


Figure 5: Impact of Data Augmentation

5.4.3 Effect of Training Set Size

We analyzed how the model performs when trained on limited data (10% and 50% of the training set). Figure 6 illustrates that while performance drops with less data, LeNet-5 remains surprisingly robust, achieving over 95% accuracy with just 10% of the data.

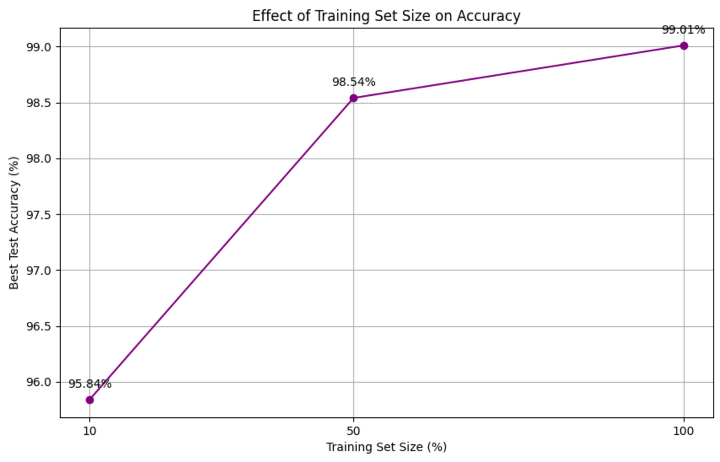


Figure 6: Accuracy vs Training Set Size

6 Discussion

- The **Baseline** model successfully reproduced the high accuracy reported in the original paper, achieving 99.01%.
- The **SGD High LR** experiment showed slightly lower performance (98.59%) and

more volatility in the loss curve, indicating that a learning rate of 0.1 is too aggressive for this architecture.

- The **Modern** configuration (Adam + ReLU) converged much faster in the initial epochs but plateaued at a similar final accuracy to the baseline (99.01%). This highlights that while modern techniques speed up convergence, the original LeNet-5 design was already highly optimized for this task.

7 Conclusion

We successfully reproduced the LeNet-5 results on MNIST. Our experiments confirm that the classic architecture remains a robust baseline for digit classification. Modern optimizations like Adam and ReLU provide faster convergence but do not significantly outperform the well-tuned SGD baseline on this specific dataset.