MNIST Digit Classification

Varun Sri Teja(23b1861)

June 11, 2025

1. Objective

The goal of this assignment is to build a neural network using PyTorch to classify handwritten digits from the MNIST dataset. We compare a simple fully connected neural network with a convolutional neural network (CNN), evaluate accuracy, and visualize the results.

2. Dataset

- **MNIST Dataset:** 60,000 training images and 10,000 test images.
- Each image is 28x28 grayscale.
- 10 classes: digits from 0 to 9.

3. Model Architectures

3.1 Simple Neural Network (NN)

- `Linear(784 → 128) → ReLU`
- `Linear(128 → 64) → ReLU`
- `Linear(64 → 10)`

3.2 Convolutional Neural Network (CNN)

- `Conv2D(1 → 6, kernel=5) → ReLU → MaxPool(2x2)`
- `Conv2D(6 → 16, kernel=5) → ReLU → MaxPool(2x2)`
- Flatten: $16 \times 4 \times 4 = 256$
- `Linear(256 → 120) → ReLU`
- `Linear(120 → 84) → ReLU`
- `Linear(84 → 10)`

4. Training Details

- Optimizer: Adam
- Loss Function: CrossEntropyLoss
- Learning Rate: 0.001
- Epochs: 10

5. Results and Evaluation

5.1 Training Loss and Accuracy

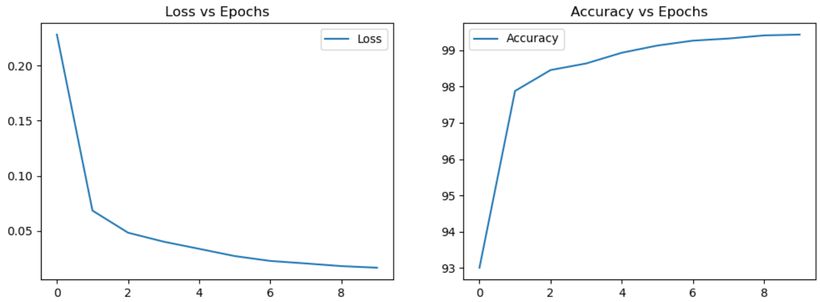


Figure 1: CNN

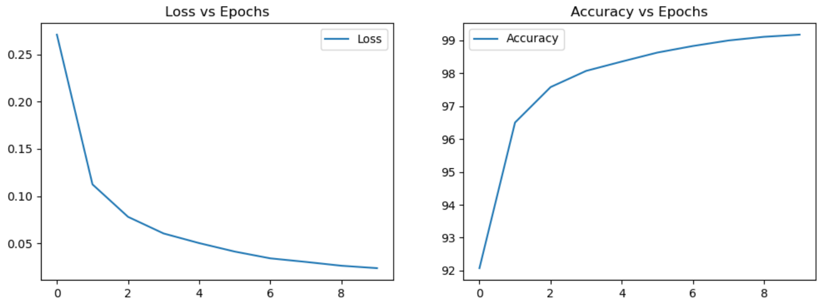
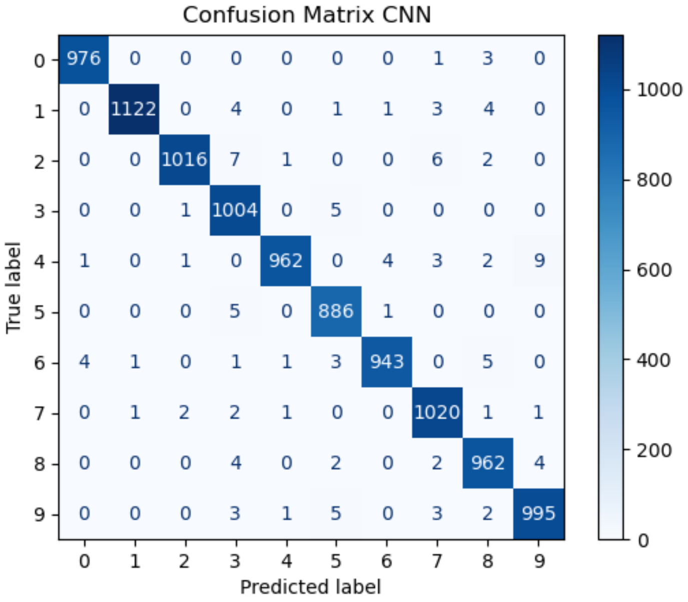


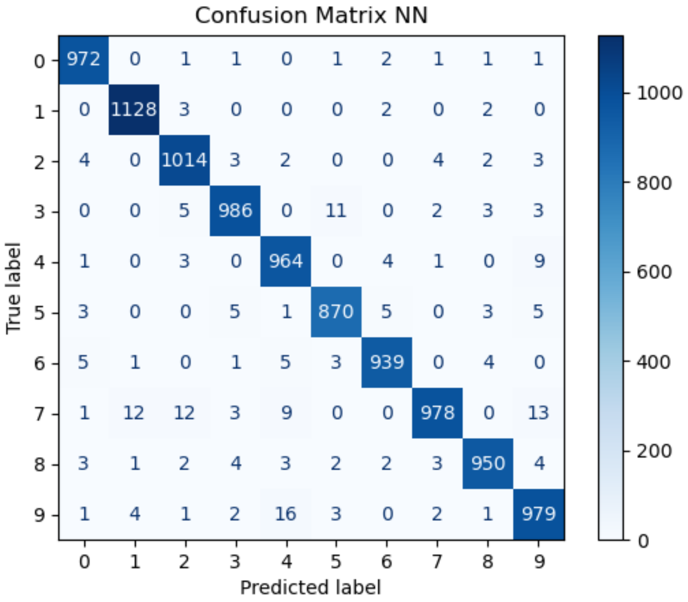
Figure 2: NN

5.2 Test Accuracy

- **Simple NN Accuracy:** 97.80%
- **CNN Accuracy:** 98.86%

5.3 Confusion Matrix





6. Key Observations

- CNN outperforms a simple NN in both accuracy and robustness.
- Most errors occur in visually similar digits (e.g. 4 and 9, 3 and 5).
- Visualization helps spot failure cases easily.