MNIST Digit Classification with a Basic CNN

Devansh Sharma

March 25, 2025

1 Introduction

This project implements a simple convolutional neural network (CNN) in PyTorch to classify handwritten digits from the MNIST dataset. The goal is to build a lightweight, interpretable model and evaluate its performance through quantitative metrics and visual diagnostics.

2 Dataset

MNIST contains 60,000 training images and 10,000 test images of 28×28 grayscale handwritten digits (0–9). Each class is balanced.

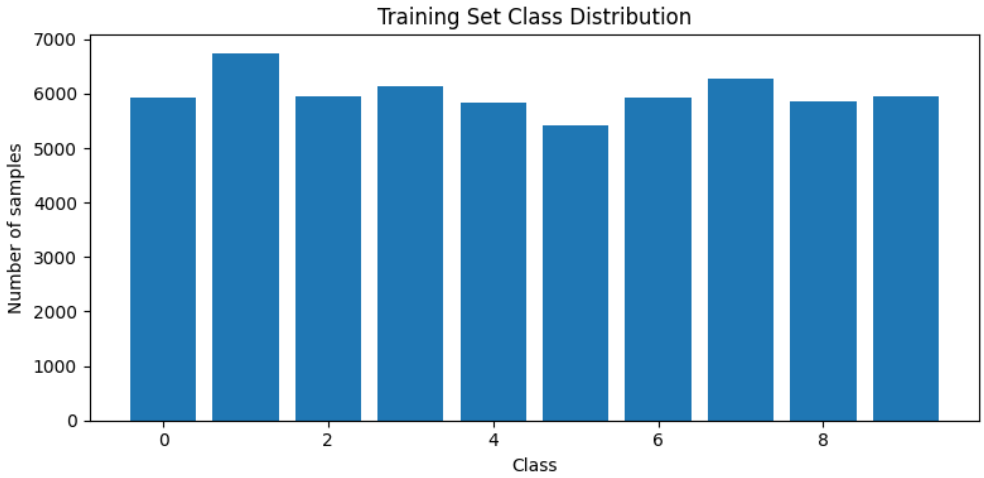

Figure 1: Training-set class distribution (10 classes, 6,000 samples each).

3 Model Design

The CNN architecture consists of two convolutional blocks (Conv2D → ReLU → MaxPool), followed by a flattened fully-connected layer of 128 units (ReLU + Dropout) and a final 10-way output layer (logits).

Training used CrossEntropyLoss with Adam (LR=1e-3) for 5 epochs on GPU (when available).

4 Results

Validation metrics by epoch:

Final test performance (checkpoint: $\texttt{mnist}_{epoch_5}.pth$) : $TestLoss = 0.0350$, $TestAccuracy = 99.05\%$

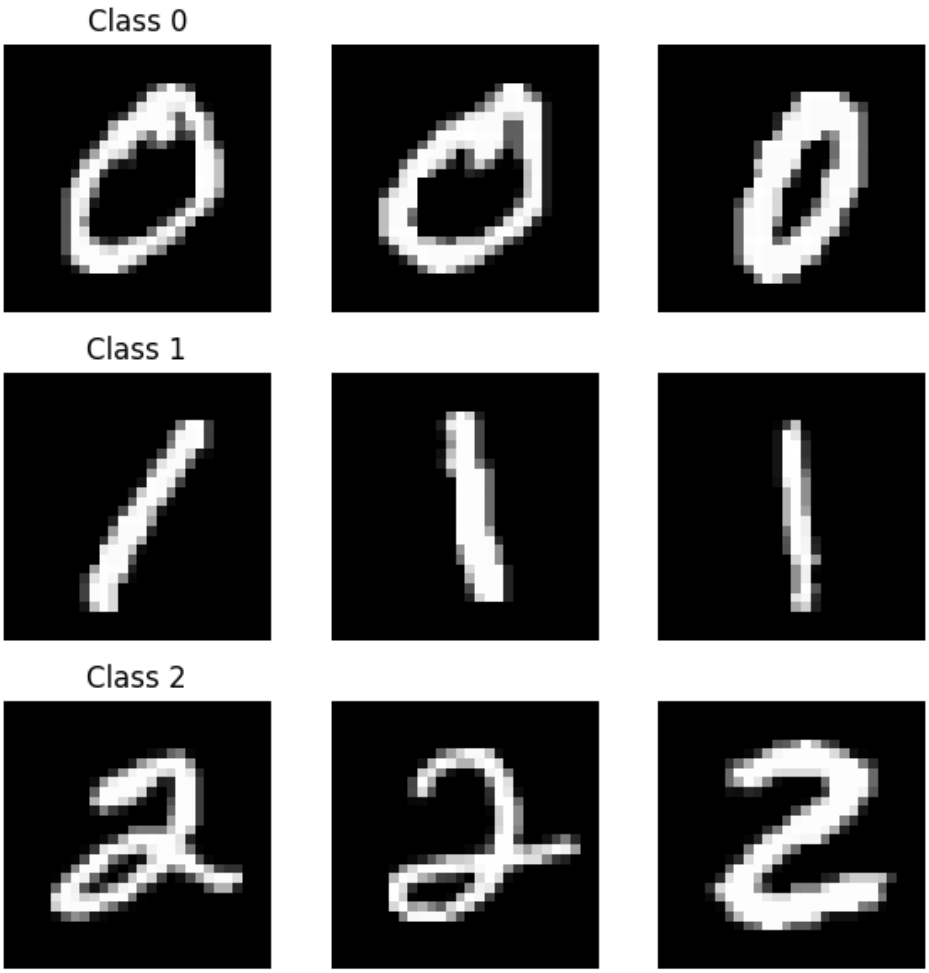

Figure 2: Example digit images from three different classes.

5 Conclusion

Our simple two-block CNN achieves 99.05% accuracy on MNIST, demonstrating strong performance with minimal complexity. Future work could explore data augmentation, deeper architectures, or self-super pretraining to push beyond this baseline on more challenging datasets.

Table 1: Network architecture summary

Layer	Kernel / Units	Activation	Output Shape
Conv1	3×3, 32 filters	ReLU	32×28×28
MaxPool	2×2	—	32×14×14
Conv2	3×3, 64 filters	ReLU	64×14×14
MaxPool	2×2	—	64×7×7
Flatten	—	—	3136
FC1	128 units	ReLU + Dropout(0.25)	128
FC2	10 units	—	10 logits

Table 2: Validation loss and accuracy per epoch

Epoch	Validation Loss	Validation Accuracy
1	0.0443	98.56%
2	0.0376	98.73%
3	0.0272	99.12%
4	0.0261	99.16%
5	0.0350	99.05%

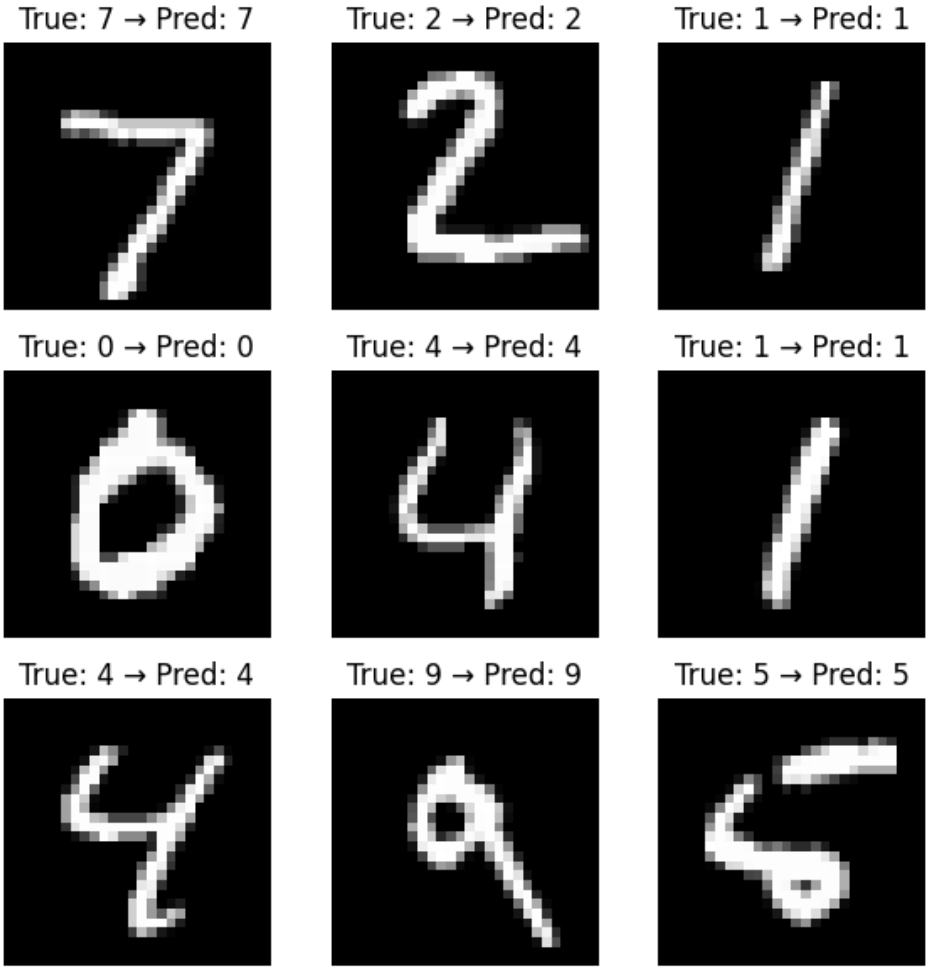

Figure 3: Predictions vs. ground truth on nine test images.