

# Handwritten Digit Classification using CNN

Final Project Report

Name: Himanshu Shete

Roll No.: 23B0770

## 1. Objective

In this project, I built and trained a Convolutional Neural Network (CNN) using PyTorch to classify handwritten digits from the MNIST dataset. The goal was to apply everything I've learned about deep learning — from data handling to model evaluation — in a practical end-to-end pipeline.

## 2. Data Preparation and Augmentation

I used the MNIST dataset (28×28 grayscale images of digits 0–9).

To prepare the data:

- Normalized it using mean 0.1307 and std 0.3081, which helps speed up training and stabilize gradients.
- Applied data augmentation (random rotation and small shifts) to make the model more robust and prevent overfitting.
- Split the training data into 55,000 for training and 5,000 for validation.
- Visualized a few samples to ensure everything loaded correctly.

## 3. CNN Architecture and Training

My CNN had:

- 2 convolutional layers with ReLU + MaxPooling
- 1 hidden fully connected layer
- An output layer with 10 neurons (one for each digit)

I trained it using:

- CrossEntropyLoss (since it combines softmax + log loss)
- Adam optimizer with learning rate = 0.001
- Batch size = 64, trained for 10 epochs

After each epoch, I tracked training accuracy and validation loss to monitor performance.

## 4. Results and Discussions

The final results:

- Training Loss: 0.0275
- Validation Loss: 0.0371
- Test Loss: 0.0215
- Test Accuracy: 99.33%

The model performed really well, but I noticed a few misclassifications — especially between 4 & 9 or 2 & 7, which can look similar in handwriting.

Challenges:

- Choosing good normalization values
- Avoiding overfitting while keeping the model simple

Improvements:

- Adding dropout or batch norm
- Using a deeper architecture (like ResNet)
- Tuning with a learning rate scheduler

## Conclusion

This project helped me understand how all the pieces — data, model design, training, evaluation — come together in a deep learning workflow. Seeing the model hit over 99% accuracy was satisfying and showed the power of even a relatively simple CNN.