# AI 100 Midterm Project

MNIST Digit Classification Using CNN

**Group Members:**

*Nawaf Alqahtani*

*Nasser Almalki*

*Mohammed Ajwah*

*Yazeed Alshehri*

*Maria Almansour*

## 1. Problem Definition

The goal of this project was to classify handwritten digits (0–9) using the MNIST dataset. Each image is 28x28 pixels and grayscale. This is a multi-class classification problem because the model must predict one of 10 possible digit classes.

## 2. Dataset Description

The MNIST dataset contains 60,000 training images and 10,000 testing images. Each image shows a handwritten digit from 0 to 9. We split the training data into a training set and a validation set so we could monitor performance while training. The images were converted to tensors and normalized before being fed into the model.

## 3. Model Architecture

The model was implemented using PyTorch and trained on a standard CPU environment.

We used a simple Convolutional Neural Network (CNN). We chose CNN because it works well for image data.

The model includes:

Two convolutional layers

Max pooling layers

One fully connected hidden layer

One output layer with 10 neurons

We used:

Cross-Entropy Loss

Adam optimizer

5 epochs

Batch size of 128

Learning rate of 0.001

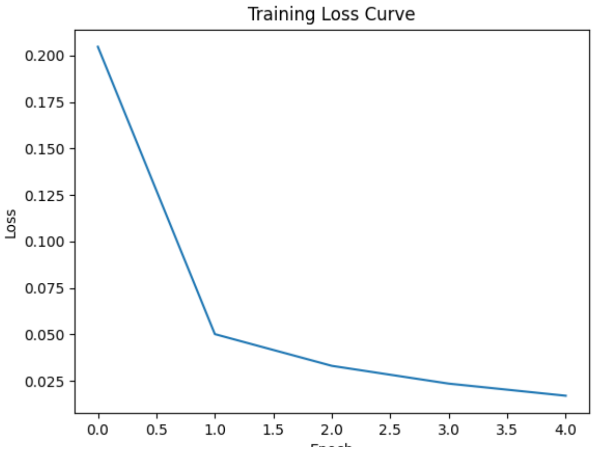
## 4. Results

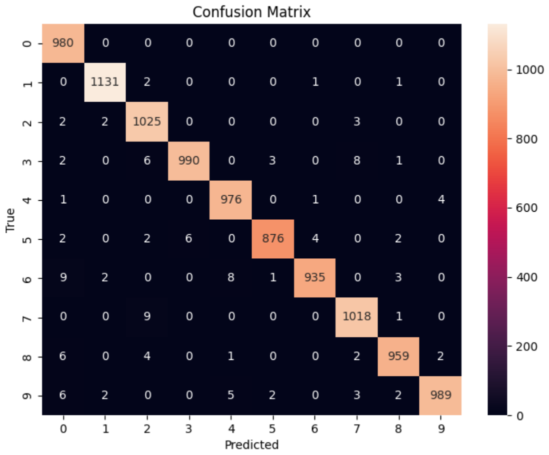
After training the model for 5 epochs, the final test accuracy was 98.79%.

The validation accuracy increased steadily across epochs, showing that the model was learning properly. Both training and validation loss decreased over time, which indicates stable learning without major overfitting.

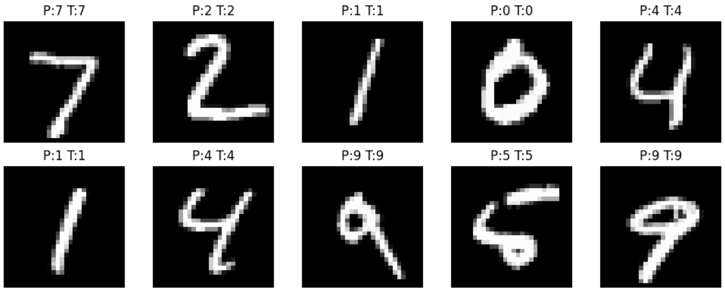
From the confusion matrix, most digits were classified correctly. However, some digits such as 4 and 9 were occasionally misclassified because their handwritten shapes can look similar. This shows that while the model performs very well, it can still struggle when digits share visual similarities.

Overall, the results demonstrate that a simple CNN is highly effective for the MNIST dataset.





| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 980 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1131 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 2 | 2 | 1025 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| 3 | 2 | 0 | 6 | 990 | 0 | 3 | 0 | 8 | 1 | 0 |
| 4 | 1 | 0 | 0 | 0 | 976 | 0 | 1 | 0 | 0 | 4 |
| 5 | 2 | 0 | 2 | 6 | 0 | 876 | 4 | 0 | 2 | 0 |
| 6 | 9 | 2 | 0 | 0 | 8 | 1 | 935 | 0 | 3 | 0 |
| 7 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 1018 | 1 | 0 |
| 8 | 6 | 0 | 4 | 0 | 1 | 0 | 0 | 2 | 959 | 2 |
| 9 | 6 | 2 | 0 | 0 | 5 | 2 | 0 | 3 | 2 | 989 |



x

## 5. Lessons Learned

This project helped us understand why Convolutional Neural Networks are effective for image classification tasks. Convolution layers extract important visual features such as edges and shapes, which allows the model to recognize patterns in handwritten digits.

We also learned the importance of splitting data into training and validation sets. Monitoring validation accuracy helped ensure that the model was improving and not simply memorizing the training data.

Although the model performed very well on MNIST, this dataset is relatively simple. More complex datasets would likely require deeper models and more training time.

## 6. Group Contributions

Nawaf Alqahtani – Project coordination and final integration

Nasser Al Malki – CNN architecture implementation

Mohammed Ajwah – Training process and hyperparameter setup

Yazeed Alshehri – Evaluation and confusion matrix analysis

Maria Almansour – Report writing and formatting