MNIST Handwritten Digit Classifier

Project Overview

This project implements a deep learning model to recognize handwritten digits (0–9) using the MNIST dataset. The objective is to train a neural network that takes an image of a digit as input and predicts the correct number it represents. This project demonstrates the complete machine learning workflow: Data loading, preprocessing, model building, training, evaluation, and model saving.

Dataset

Total images: 70,000

Training images: 60,000

Test images: 10,000

Image size: 28 × 28 pixels

Color channel: Grayscale (1 channel)

Classes: 10 digits (0–9).

Data Preprocessing

Pixel values are normalized from 0–255 to 0–1. Images are reshaped from (28, 28) to (28, 28, 1) to include the channel dimension required for convolutional neural networks.

Model Architecture

Conv2D (32 filters) detects basic patterns. MaxPooling reduces size. Conv2D (64 filters) detects complex shapes. MaxPooling reduces dimensions further. Flatten converts 2D features to 1D. Dense layer (128 neurons) learns high-level patterns. Output Dense layer (10 neurons, Softmax) predicts digit probabilities.

Training

Optimizer: Adam

Loss Function: Sparse Categorical Crossentropy

Epochs: 5

Validation Split: 10%.

Results

Test Accuracy: Approximately 98–99%.

Model Saving

The trained model is saved as mnist_model.h5 for reuse without retraining.

How to Run

1. Install dependencies: pip install -r requirements.txt
2. Run training script: python main.py

Technologies Used

Python, TensorFlow/Keras, NumPy.

Conclusion

This project shows how deep learning models learn visual patterns and classify images, a technique applicable to medical imaging, facial recognition, and object detection.