Student name: Priyadharshini.S

Register number:622423121033

Institute:salem college of engineering and technology

Department: Biomedical engineering

Date of submission:26.04.2025

GitHup Respository : ***https://github.com/Priya9791/NM-PROJECT-.git***

Recongnizing handwritten digits with deep learning for smarter AI applications

Problem statement:

The goal is to develop an intelligent system capable of accurately recognizing and classifying handwritten digits (0–9) from images using deep learning techniques.

Traditional optical character recognition (OCR) systems often struggle with variations in human handwriting styles.

By leveraging convolutional neural networks (CNNs), this project aims to improve recognition accuracy and robustness, enabling practical applications in automated data entry, banking, postal services, and educational tools.

Objective of the project:

To design and implement a deep learning model, specifically a Convolutional Neural Network (CNN), capable of recognizing handwritten digits (0–9).

To train the model using the MNIST dataset, ensuring high accuracy and generalization on unseen data.

To evaluate the model's performance using standard metrics such as accuracy, precision, recall, and confusion matrix.

To demonstrate the real-world application of the model in tasks like automatic digit recognition for forms, postal codes, or digital input systems.

To optimize the model architecture for improved performance while minimizing training time and computational resources.

Scope of the project:

Dataset Scope:

The project will utilize the MNIST dataset, consisting of 60,000 training and 10,000 testing grayscale images of handwritten digits (0–9), each sized 28x28 pixels.

Model Development:

A Convolutional Neural Network (CNN) will be designed and trained to classify the digits with high accuracy.

Technologies Used:

The project will be developed using Python and deep learning libraries such as TensorFlow or PyTorch, along with tools like NumPy, Matplotlib, and OpenCV.

Application Scope:

The trained model can be applied to real-world scenarios like OCR systems, automated form readers, postal code recognition, and mobile banking.

Data sources:

MNIST Dataset

Description: A benchmark dataset of 70,000 labeled grayscale images of handwritten digits (0–9).

Size: 60,000 training images and 10,000 test images.

Format: 28x28 pixel images in grayscale.

Source: Available freely from multiple platforms including:

Optional Extended

EMNIST: Extended MNIST includes handwritten letters and digits.

Kuzushiji-MNIST: Japanese cursive character dataset.

High – level of methodology:

Data Collection:-

Use the MNIST dataset, consisting of labeled images of handwritten digits.

Data Preprocessing:-

Normalize pixel values (0–255 scaled to 0–1).

Reshape images as required by the model input.

Split data into training, validation, and testing sets.

Model Design:-

Build a Convolutional Neural Network (CNN) with layers such as convolution, pooling, dropout, and fully connected layers.

Model Training:-

Train the CNN using the training data.

Use a loss function like categorical cross-entropy and an optimizer such as Adam or SGD.

Apply techniques like dropout or batch normalization to prevent overfitting.

Model Evaluation:-

Evaluate the model using the test set with metrics like accuracy, loss, and confusion matrix.

Model Deployment (Optional/Future Scope)

Integrate the trained model into a user interface or API for real-time digit recognition.

Tools and techonolgy:

Programming Language:-

Python: Widely used for AI and deep learning due to its simplicity and powerful libraries.

Deep Learning Frameworks:-

TensorFlow or Keras: For building and training neural networks.

PyTorch (alternative): Another popular deep learning library.

Libraries:-

NumPy: **For numerical operations and array handling.**

Matplotlib / Seaborn: For visualizing data, model performance, and training progress.

Pandas: For data handling (optional).

scikit-learn: For evaluation metrics like confusion matrix and accuracy.

Development Environment:-

Jupyter Notebook or Google Colab: For interactive development and visualization.

VS Code / PyCharm (optional): For script-based development.

Dataset:-

MNIST Dataset: Standard dataset for training and testing digit recognition models.

Hardware:-

GPU Acceleration (optional): For faster training using platforms like Google Colab or local GPU.

Team members and roles:-

Prabhavathi:

Problem statement

Objectives of the project

Priyadharshini:

Scope of project

Data source

Ramachandran:

High – level methodology

Tool and technology