Student Name: Kamalesh. M, Thulasiraman. S, Muzamil Arafath. V.T, Pazhaniyammal .A

Register Number: 510823243019,510823243306,510823243032,510823243037

Institution: Ganadipathy Tulsi's Jain Engineering College

Department: B.Tech (Artificial Intelligence and Data Science)

Date of Submission: 26-04-2025

---

1. Problem Statement

The ability to recognize handwritten digits is fundamental to many AIpowered applications, such as automated postal sorting, bank check processing, and digitized form recognition. Traditional machine learning methods face challenges in generalizing well across diverse handwriting styles. This project addresses the need for a more robust, accurate solution using deep learning, specifically convolutional neural networks (CNNs), to recognize digits written by hand and enhance the performance of AI systems in real-world scenarios.

2. Objectives of the Project

To develop a deep learning model capable of accurately classifying handwritten digits.

To explore and compare the performance of various CNN architectures. To generate visual insights into how the model makes decisions (e.g., with activation maps).

To build a simple interactive tool for testing the digit recognition model.

3. Scope of the Project

Features to be Analyzed/Built:

Handwritten digit classification using CNNs.

Visualization of model performance.

Interactive digit prediction interface.

4. Data Sources

Dataset: MNIST Handwritten Digits Dataset

Source: [Kaggle/UCI/Official MNIST site]

Description: An extended version of MNIST including letters and digits. It provides multiple splits such as EMNIST Balanced, EMNIST Letters, EMNIST Digits, etc., and is designed to test classification models on a wider variety of handwritten characters.

Type: Public and static dataset (not updated in real-time).

5. High-Level Methodology

Data Collection:

Download the MNIST dataset from TensorFlow/Keras datasets module or Kaggle.

Data Cleaning:

MNIST is pre-cleaned but normalization (0–1 scaling) will be applied. No missing values expected.

Exploratory Data Analysis (EDA):

Use visualizations (matplotlib/seaborn) to inspect class distribution and pixel intensity patterns.

Feature Engineering:

Images will be reshaped and normalized. No complex feature engineering needed due to CNN capabilities.

Model Building:

Experiment with CNN architectures (e.g., LeNet-5, simple 3-layer CNN). Use ReLU activations and Softmax for output.

Model Evaluation:

Use accuracy, precision, recall, F1-score, and confusion matrix. Validation split and test set evaluation will be included.

Visualization & Interpretation:

Present training curves, confusion matrix, and sample predictions with heatmaps/Grad-CAM if possible.

Deployment:

Deploy via Streamlit or Gradio to create a simple web app where users can draw a digit and get predictions.

.

6. Tools and Technologies

1. Programming Language:

Python: Widely used in machine learning and AI development due to its simplicity, flexibility, and vast ecosystem of libraries.

2. Development Environment:

Google Colab: Cloud-based Jupyter Notebook environment that supports free GPU acceleration, ideal for deep learning projects.

Jupyter Notebook : Local development and testing of code in an interactive format.

3. Core Libraries and Frameworks:

NumPy: For efficient numerical computations and array manipulations.

Pandas: For data manipulation, exploration, and preprocessing.

Matplotlib & Seaborn: For creating visualizations such as training curves, confusion matrices, and digit samples.

TensorFlow & Keras: Deep learning frameworks used to build, train, and evaluate CNN models for image classification tasks.

4. Model Evaluation & Visualization Tools

Scikit-learn: For metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

TensorBoard: For monitoring model training, losses, and activations.

Grad-CAM : To visualize CNN feature maps and understand what the model is learning.

7.Team Members and Roles

Kamalesh.M - Development

Pazhaniyammal. A -Design

Muzamil Arafath. V.T- Communication

Thulasiraman .S - Documentation