Student Name: HEMALAKCHANA N

Register Number: 510123106014

Institution: Adhiparasakthi College of Engineering

Department: B.E. Electronics and Communication Engineering

Date of Submission:08-05-2025

PHASE-1

Recognizing handwritten digits with deep learning for smarter AI applications

1.Problem Statement

In today's increasing digital world, automating the recognition of handwritten digits remains a challenge due to the vast diversity in individual writing styles, shapes, and sizes. Traditional image processing techniques often struggle with inconsistencies in handwriting. The lack of accuracy and adaptability in such systems limits their real-world applications in fields like postal services, banking and education

2.Object of the Project

- To develop a deep learning model capable of accurately recognizing and classifying handwritten digits from image data.
- To utilize popular datasets such as MNIST for training, validating, and testing the performance of the model.
- To explore and implement neural network architectures, particularly Convolutional Neural Networks (CNNs), which are effective for image recognition tasks.
- To evaluate model performance using appropriate metrics such as accuracy, precision, recall, and confusion matrix.
- To demonstrate real-world applicability in fields like automated data entry, postal sorting, bank check processing, and digital form recognition.

3.Scope of the Project

Features to Analyze:

- Manual extraction of features like pixel intensity, edges, or histograms before classification.
- Use of algorithms like Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees.
- These systems perform well on clean, structured data but struggle with variations in handwriting styles, distortions, or noise.

Limitations/Constraints:

- The model is heavily reliant on the quality and diversity of the training data (e.g., MNIST may not represent all real-world handwriting variations).
- The project is limited to recognizing digits (0–9) and does not extend to letters or symbols.
- The model treats each digit in isolation without understanding surrounding context or sequences (e.g., full addresses or numbers)

4.Existing System

Currently, digit recognition tasks are commonly handled using traditional machine learning or basic image processing techniques, often combined with manual feature extraction. These systems rely on:

- Manual extraction of features like pixel intensity, edges, or histograms before classification.

- Use of algorithms like Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees.
- These systems perform well on clean, structured data but struggle with variations in handwriting styles, distortions, or noise.

5. Proposed System

The proposed system utilizes a Convolutional Neural Network (CNN) for recognizing handwritten digits. This deep learning model is trained on the MNIST dataset, a widely-used benchmark that consists of 70,000 labeled images of handwritten digits (0–9). The system aims to achieve high accuracy and real-time performance, making it suitable for integration into smarter AI applications such as automated form readers, check scanners, and educational tools.

- Use the MNIST dataset or gather custom digit datasets using image capture or scanning devices.
- Use labeled images from the MNIST dataset.
- Optimization with Adam optimizer.

6. Data Sources

1. MNIST Dataset

Description: The most widely used dataset for handwritten digit recognition.

Content: 60,000 training images and 10,000 testing images of digits (0–9), 28x28 grayscale.

Source: Yann LeCun's website or via TensorFlow/Keras/PyTorch libraries.

2. EMNIST Dataset (Extended MNIST)

Description: An extended version of MNIST including handwritten letters and digits.

Content: Over 280,000 characters including digits and uppercase/lowercase letters.

Source: EMNIST on Kaggle

7. High-Level Methodology

1. Problem Definition

Identify the need to accurately recognize handwritten digits for smart applications such as OCR, banking, and education tools.

Define goals: high accuracy, real-time recognition, easy deployment.

2. Data Collection & Preprocessing

Use standard datasets like MNIST or EMNIST, or create a custom dataset.

Convert to grayscale:Normalize pixel values (e.g., scale from 0–255 to 0–1)

8. Tools and Technologies

Programming Language: Python

Deep Learning Frameworks: TensorFlow

Data Handling & Visualization: NumPy, Pandas, Matplotlib

Model Deployment: Flask , TensorFlow Lite , Stream lit / Grado

Hardware/Environment: Google Collab, NVIDIA GPU

9. Team Members and Roles

1.CHARUMATHI.V-Project Manager

2. HEMALAKCHANA.N-Data Scientist

3.SWATHI.K-Data Engineer

4.SANGAVI.S- Feature Engineering and Modeling

5.SARASWATHI.P-Documentation and presentation