**Phase-2: Recognizing handwritten digits with learning for smart AI application**

**Student Name:** GOKUL. M

**Register Number:** 510923205020

**Institution:** Global institute of engineering and technology, melvisharam

**Department:** B.tecch-IT

**Date of Submission:** 07/05/2025

**GitHub Repository Link:** [Update the project source code repository link]

**1.Problem Statement**

Handwritten digit recognition is a foundational task in pattern recognition and AI-driven automation. Despite its simplicity, accurately recognizing digits in varied handwriting styles remains challenging due to inconsistencies in human writing, noise, and varied input formats. This project aims to build a machine learning-based smart system capable of classifying handwritten digits accurately, enabling applications like digital form reading, bank check processing, postal code recognition, and education technology.

**2.Project Objectives**

 To develop a robust AI model capable of recognizing and classifying handwritten digits (0–9).

To use supervised learning techniques (especially CNN) for high accuracy.

To design a user-friendly interface or module for integrating the model into smart applications.

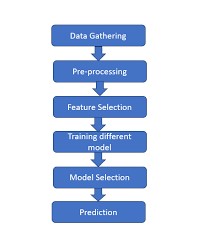


To explore real-time digit prediction through optional automation tools.

To ensure adaptability and reusability of the model in various environments (web, mobile, embedded).



**3.Flowchart of the Project Workflow**



**4.Data Description**

**Dataset**: MNIST (Modified National Institute of Standards and Technology)

**Size**: 70,000 images (60,000 training + 10,000 testing)



**Image format**: 28x28 grayscale

**Label format**: Integer values from 0 to 9

**Data Source**: Kaggle, TensorFlow/Keras datasets

**5.Data Preprocessing**

* + **Normalization: Scale pixel values from [0, 255] to [0, 1]**
  + **Reshaping: Convert images to shape (28, 28, 1) for CNNs**
  + **Encoding: One-hot encode the labels**
  + **Splitting: Divide the dataset into training, validation, and test sets**
  + **Noise Handling (optional): Apply filters to remove background noise**

**6.Exploratory Data Analysis (EDA)**

* + **Class distribution: Ensure all digits are equally represented**
  + **Sample visualization: Plot random samples using matplotlib**
  + **Pixel intensity distribution**
  + **Misclassified sample analysis (after model evaluation)**

**7.Tools and Technologies Used**

**a. Programming Language & Notebook**

* + **Language**: Python 3.x
  + **IDE/Notebook**: Jupyter Notebook or Google Colab

**b. Libraries and Frameworks**

* **NumPy**: For numerical computations
* **Pandas**: For handling tabular data
* **Matplotlib / Seaborn**: For plotting and data visualization
* **TensorFlow / Keras**: For building and training deep learning models
* **Scikit-learn**: For performance metrics like confusion matrix and accuracy

**c. Model Architecture**

* **Convolutional Neural Network (CNN)**: o Conv2D layers for feature extraction
* MaxPooling for downsampling o Dense layers for classification
* **Activation Functions**: ReLU, Softmax
* **Loss Function**: Categorical Crossentropy
* **Optimizer**: Adam

**d. Optional Automation Tools**

* **TensorFlow Lite**: For deploying models on mobile devices
* **Flask** or **Streamlit**: For web application development
* **Docker**: For containerizing the application
* **AutoML Tools** (Optional): e.g., Keras Tuner for hyperparameter tuning

**8. Team Members and Contributions**

Jeevanandam B Exploratory data analysis, data description

Sharukesh J Tools and technologies used, problem statement

|  |  |
| --- | --- |
| Harish L    Ganesh T    Gokul M | Data preprocessing    Project objective    Flowchart of the project workflow |