Student Name: H.Shanmathi

Register Number: 732323106046

Institution: SSM College of Engineering

Department: Electronics and Communication Engineering

Date of Submission: April 23, 2025

1.Problem Statement

The problem is to develop a system that can accurately recognize handwritten digits (0-9) from images, enabling smarter AI applications such as automated form processing, digit-based authentication systems, and educational tools. This problem is important because handwritten digit recognition is a foundational application in computer vision and deep learning, with real-world implications for improving efficiency in data entry, enhancing accessibility, and reducing human error in digit-based systems.

2.Objectives of the Project

Build a deep learning model to classify handwritten digits with high accuracy (targeting at least 98% accuracy on a standard dataset).

Generate insights into the performance of different neural network architectures for digit recognition.

Create a deployable application or interface to demonstrate real-time handwritten digit recognition.

Analyze the impact of data preprocessing and feature engineering on model performance.

3.Scope of the Project

Features to Analyze or Build:

Preprocessing of handwritten digit images (e.g., normalization, grayscale conversion).

Deep learning models such as Convolutional Neural Networks (CNNs) for digit classification.

A simple user interface to upload or draw digits and display predictions.

4.Data Sources

The project will use the MNIST dataset, a widely-used public dataset available from sources like Kaggle, TensorFlow, or the UCI Machine Learning Repository. The dataset contains 70,000 grayscale images of handwritten digits (60,000 training, 10,000 testing), each 28x28 pixels. The dataset is static and will be downloaded once for use in the project.

5.High-Level Methodology

Data Collection: Download the MNIST dataset from TensorFlow/Keras or Kaggle using Python libraries.

Data Cleaning: Check for missing or corrupted images, normalize pixel values (0-255 to 0-1), and ensure consistent image dimensions (28x28 pixels).

Exploratory Data Analysis (EDA): Visualize sample images, analyze the distribution of digits (0-9), and check for class imbalance using histograms and heatmaps.

Feature Engineering: Apply image preprocessing techniques such as normalization, centering, and possibly augmentation (e.g., rotation, scaling) to improve model robustness.

Model Building: Experiment with Convolutional Neural Networks (CNNs) and possibly simpler models like Multi-Layer Perceptrons (MLPs) for comparison. CNNs are suitable due to their ability to capture spatial patterns in images.

Model Evaluation: Use accuracy, precision, recall, and confusion matrices to evaluate model performance. Employ k-fold cross-validation to ensure robustness.

Visualization & Interpretation: Display model performance metrics using plots (e.g., accuracy/loss curves) and visualize misclassified digits to understand model weaknesses.

Deployment: Deploy the model as a web application using Streamlit or Flask, allowing users to upload or draw digits and view predictions.

6.Tools and Technologies

Programming Language: Python

Notebook/IDE: Google Colab for model development, VS Code for deployment-related coding

Libraries:

Data Processing: pandas, numpy.

Visualization: matplotlib, seaborn.

Modeling: TensorFlow/Keras for deep learning, scikit-learn for evaluation metrics.

Optional Tools for Deployment: Streamlit or Flask for creating a web-based interface.

7.Team Members and Roles

[SHANMATHI.H]: Responsible for data collection, model building, and deployment.

[SWETHA.K]: Handles data cleaning, EDA, and visualization.

[K.SIBI]: Focuses on feature engineering and model evaluation.

[S.SUJITH]: Designs and implements the web interface for deployment.