**Recognizing Handwritten Digits using Deep Learning for Smart AI Application**

**Phase-1 Submission Template**

**Student Name:** VASUTHA S T

**Register Number:** 731123104703

**Institution:** Government College of Engineering, Erode

**Department:** Computer Science And Engineering

**Date of Submission:** 28-04-2025



# 1.Problem Statement

* Recognizing handwritten digits is a challenging real-world problem due to the wide variation in human handwriting.
* It is crucial for applications such as postal mail sorting, bank check processing, and digitizing handwritten notes.
* Goal: Develop an accurate deep learning model that can classify handwritten digits (0-9).

# 2.Objectives of the Project

* Develop a deep learning model to classify handwritten digits.
* Achieve high accuracy in prediction.
* Understand the key features of handwritten characters through model learning.
* Evaluate and optimize the model for best performance.

# 3.Scope of the Project

**Features:**

* Image classification using Convolutional Neural Networks (CNNs).
* Analysis of model performance.

**Limitations:**

* Dataset limited to MNIST (28x28 grayscale images).
* No real-time handwriting recognition (static images only).

# 4.Data Sources

Dataset Used: MNIST Handwritten Digits Dataset.

Source: Public dataset from [Kaggle / official MNIST site].

Type: Static dataset (downloaded once).

Details: 60,000 training images, 10,000 testing images, each image of size 28x28 pixels.

# 5.High-Level Methodology

**Data Collection**:

Download MNIST dataset from Keras/TensorFlow datasets.

**Data Cleaning**:

Normalize pixel values (scale from 0-255 to 0-1).

**Exploratory Data Analysis (EDA)**:

Visualize sample digits, check data distribution.

**Feature Engineering**:

Reshape and normalize images; one-hot encode labels.

**Model Building:**

Create a CNN with layers like Conv2D, MaxPooling, Flatten, Dense.

**Model Evaluation:**

Use accuracy score, loss plots, confusion matrix.

**Visualization & Interpretation:**

Plot training/validation loss and accuracy graphs.

**Deployment (Optional):**

Deploy as a simple web app using Streamlit/Gradio.

# 6.Tools and Technologies

Programming Language: Python

Notebook/IDE: Google Colab / Jupyter Notebook

Libraries:

Data Processing: numpy, pandas

Visualization: matplotlib, seaborn

Deep Learning: TensorFlow, Keras

Optional Deployment Tools: Streamlit or Gradio (for simple UI deployment)

# 7.Team Members and Roles

| **Name** | **Role** |
| --- | --- |
| Vasutha S T | Data Collection |
| Kiruthika S | Data cleaning |
| Deepakala C M | Exploratory Data Analysis |
| Vijayadharshini B | Model Building and Evaluation |
| Kaviyasri RV | Visualization and Final Presentation |