PHASE 1-SUBMISSION TEMPLATE

Student Name:	Ajeem.S
Register Number:	620123106001
Instiution Name:	AVS ENGINEERING COLLEGE
Branch:	ECE
Date of submission:	30.04.25

1.Problem Statement: In many real-world scenarios, such as digitizing handwritten documents, automating postal services, and processing bank checks, recognizing handwritten digits is essential. Manual digit recognition is time-consuming and prone to error. This project aims to develop an intelligent deep learning system capable of accurately recognizing handwritten digits to enhance the efficiency and reliability of such applications.

2.Objectives of the project :

- Build a deep learning model that can classify handwritten digits (0-9) with high accuracy.
- Train the model on a widely-used dataset to ensure robustness and generalizability.
- Evaluate the model's performance using appropriate metrics.
- Deploy the model in a simple web interface or notebook for demonstration.

3.Scopes of the Project:

- Features to be developed: Image preprocessing, CNN-based digit classifier, performance dashboard.
- Limitations/Constraints:
- Dataset limited to grayscale 28x28 images (MNIST).
- Deep learning model limited to convolutional neural networks (CNNs).

- Deployment limited to a local or simple web-based interface using Streamlit or similar.

4.Data Sources:

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

- Source: Publicly available on Kaggle, also accessible via Keras and TensorFlow libraries

- Type: Public, static dataset

- Format: Images of handwritten digits (28x28 pixels, grayscale)

5.High Level Methadology:

- Data Collection:

Download MNIST dataset using TensorFlow/Keras libraries or from Kaggle.

- Data Cleaning:

Normalize pixel values, ensure image-label pairs are correct, reshape data if needed.

- Exploratory Data Analysis (EDA):

Visualize sample digits, check label distribution, analyze pixel intensity patterns.

- Feature Engineering:

Apply normalization and reshape images as needed for CNN input.

- Model Building:

Design and train a Convolutional Neural Network (CNN) with layers like Conv2D, MaxPooling, and Dense. Experiment with dropout and regularization to avoid overfitting.

- Model Evaluation:

Use accuracy, precision, recall, confusion matrix, and validation loss/accuracy curves.

- Visualization & Interpretation:

Display predictions on test images, training history graphs, confusion matrix, and sample misclassified digits.

- Deployment:

Build an interactive app using Streamlit where users can draw/upload digits and get predictions.

6.Tools And Technologies:

- Programming Language: Python
- Notebook/IDE: Google Colab / Jupyter Notebook / VS Code
- Libraries:
- Data Processing: numpy, pandas
- Visualization: matplotlib, seaborn
- Modeling: TensorFlow, Keras
- Deployment: Streamlit (optional)

7.Team Members And Roles:

- Kaviyarasu P - Model architecture, training, and evaluation
- Kaviyarasu T - Data preprocessing, feature engineering, and testing
- Gokul K - Exploratory data analysis, visualizations, and documentation
- Ajeem S - Deployment interface, user interaction, and final presentation