Phase-2 Submission Template

Student Name :Ajeem S

Register Number:620123106001

Institution:AVS Engineering College

Department:ECE

Date of Submission:10/05/2025

GitHub Repository Link:

1. Problem Statement

This project focuses on the automatic recognition of handwritten digits using deep learning techniques.

- Problem Type: Classification

- Dataset Insight: The dataset consists of grayscale images of handwritten digits (0-9), labeled accordingly.

- Relevance: Accurate digit recognition is essential in fields such as postal automation, bank cheque digitization, and intelligent form processing. This project contributes to smarter and faster AI applications in real-world scenarios.

2. Project Objectives

- Develop a convolutional neural network (CNN) model to classify handwritten digits with high accuracy.

- Apply deep learning techniques for pattern recognition in image data.
- Enhance the model using regularization and augmentation techniques for robustness.
- Updated Goals: Post-EDA, the objective expanded to include data augmentation and architecture optimization for higher generalization.

3. Flowchart of the Project Workflow

Data Collection -> Preprocessing -> Data Augmentation -> CNN Model Design -> Training -> Evaluation -> Visualization -> (Optional: Deployment)

4. Data Description

- Dataset Name and Source: MNIST Handwritten Digits Dataset (from Kaggle)
- Type of Data: Structured (Image - Grayscale, 28x28 pixels)
- Number of Records and Features: 60,000 training images, 10,000 testing images
- Nature: Static
- Target Variable: Digit class (0-9)

5. Data Preprocessing

- Missing Values: None
- Normalization: Pixel values scaled to [0, 1]
- Shape Conversion: Images reshaped to (28, 28, 1)
- Label Encoding: Applied one-hot encoding to digit labels
- Data Augmentation: Used random rotations, zoom, and shifts to increase dataset variety

6. Exploratory Data Analysis (EDA)

- Univariate Analysis: Countplot of digit frequency
- Visual Analysis: Sample images plotted to observe writing variations
- Insights:
  - All digit classes are balanced
  - Digits such as '3', '5', and '8' show higher misclassification risk due to visual similarity

7. Feature Engineering

- Used raw image pixels as input features
- Applied data augmentation to artificially expand the training dataset
- CNN extracts deep features automatically-no manual feature extraction required

8. Model Building

- Models Used:
  - Basic CNN with 2 Conv layers + MaxPooling
  - Enhanced CNN with Dropout and BatchNormalization
- Justification: CNNs are highly effective for image classification due to spatial feature extraction
- Train/Test Split: 80/20 split from training set
- Evaluation Metrics:
  - Accuracy: [Insert your accuracy, e.g., 98.7%]
  - F1-Score, Precision, Recall (for performance on individual digits)

9. Visualization of Results & Model Insights

- Confusion Matrix: Shows per-class accuracy and common misclassifications
- Accuracy/Loss Curves: Tracked training and validation metrics over epochs
- Feature Maps: Visualized CNN layers to understand learned patterns
- Performance Comparison: Bar graph comparing accuracy of baseline and enhanced models

10. Tools and Technologies Used

- Programming Language: Python
- IDE/Notebook: Google Colab
- Libraries: TensorFlow, Keras, NumPy, pandas, matplotlib, seaborn
- Visualization Tools: matplotlib, seaborn

11. Team Members and Contributions

Name	Responsibility
Kaviyarasu T	Data preprocessing, CNN model development, training and evaluation

Kaviyarasu P	Exploratory Data Analysis (EDA), data visualization, confusion matrix
Ajeem S	Data augmentation, feature map visualization, performance improvements
Gokul K	Documentation, GitHub management, and presentation preparation