**Phase-2 Submission Template**

**Student Name:** Gopinath p

**Register Number:** 621823104017

**Institution:** PGP college of engineering and technology

**Department:** BE-computer science and engineering

**Date of Submission:** 10.05.2025

**Github Repository Link:** [Update the project source code to your Github Repository]

### **1. Problem State****ment**

### *This project focuses on the classification of handwritten digits using deep learning techniques. The core problem involves recognizing digits (0–9) from image data, which is a common task in optical character recognition (OCR). The importance lies in its application to real-world AI systems like postal code reading, bank check processing, and form digitization.*

### **2. Project Objective****s**

### *Build an accurate model that can classify handwritten digits from images.*

### *Compare the performance of a traditional machine learning model (e.g., SVM) with a deep learning model (e.g., CNN).*

### *Improve prediction accuracy using preprocessing and regularization techniques.*

*Ensure generalization across varied writing styles.*

### **3. Flowchart of the Project Workflow**

### 

### 

### **4. Data Description**

### *Dataset: MNIST Handwritten Digit Dataset*

### *Source: Kaggle or directly from Keras*

### *Data Type: Image (28x28 grayscale)*

### *Records: 60,000 training images, 10,000 test images*

### *Static Dataset*

### *Target Variable: Digit class (0 to 9)*

### **5. Data Preprocessing**

*Normalize pixel values (0–255 scaled to 0–1).*

*Reshape images for CNN input (28x28x1).*

*Encode labels as categorical (one-hot encoding).*

### 

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

*Visualize sample digits.*

*Analyze class distribution (should be uniform).*

*Check image intensity distribution.*

*Heatmaps for pixel intensity patterns.*

### 

### **7. Feature Engineer****ing**

Since it's image data, not much feature creation.

Use raw pixel values and convolutional filters.

Optionally apply PCA for dimensionality reduction on raw models.

### **8. Model Building**

*Models used: CNN (Convolutional Neural Network), SVM*

*Training/testing split: 80/20*

*Evaluation metrics: Accuracy, Precision, Recall, F1-score*

*CNN expected to outperform SVM due to spatial feature learning.*

### 

### 

### **9. Visualization of Results & Model Insights**

*Confusion matrix*

*Accuracy and loss curves (for CNN)*

*ROC curves (optional for multi-class)*

*Visualize filters and activations in CNN*

### 

### **10. Tools and Technologies Used**

* *Language: Python*
* *IDE: Jupyter Notebook / Google Colab*
* *Libraries: TensorFlow, Keras, matplotlib, seaborn, pandas, numpy, scikit-learn*

### 

### **11. Team Members and Contributions**

***Gopinath.p-Data Cleaning and process***

***Maharaja L- exploratory data analysis***

***Madhan M-Model development and evaluation***

***Manikhandan S-Documentation and reporting***