Phase-2 Submission Template

Student Name:M.R.Tameema Naazmi

Register Number: 712523121024

Institution: PPG Institute of Technology

Department: Biomedical Engineering

Date of Submission: 02-05-2025

Github Repository Link:
https://github.com/Tameema14naazmi/NM_Deep-Learning-AI.git

1. Problem Statement

The goal of this project is to develop a deep learning model that can accurately recognize handwritten digits. This capability is fundamental for applications like automated form processing, postal address recognition, and digital handwriting analysis.
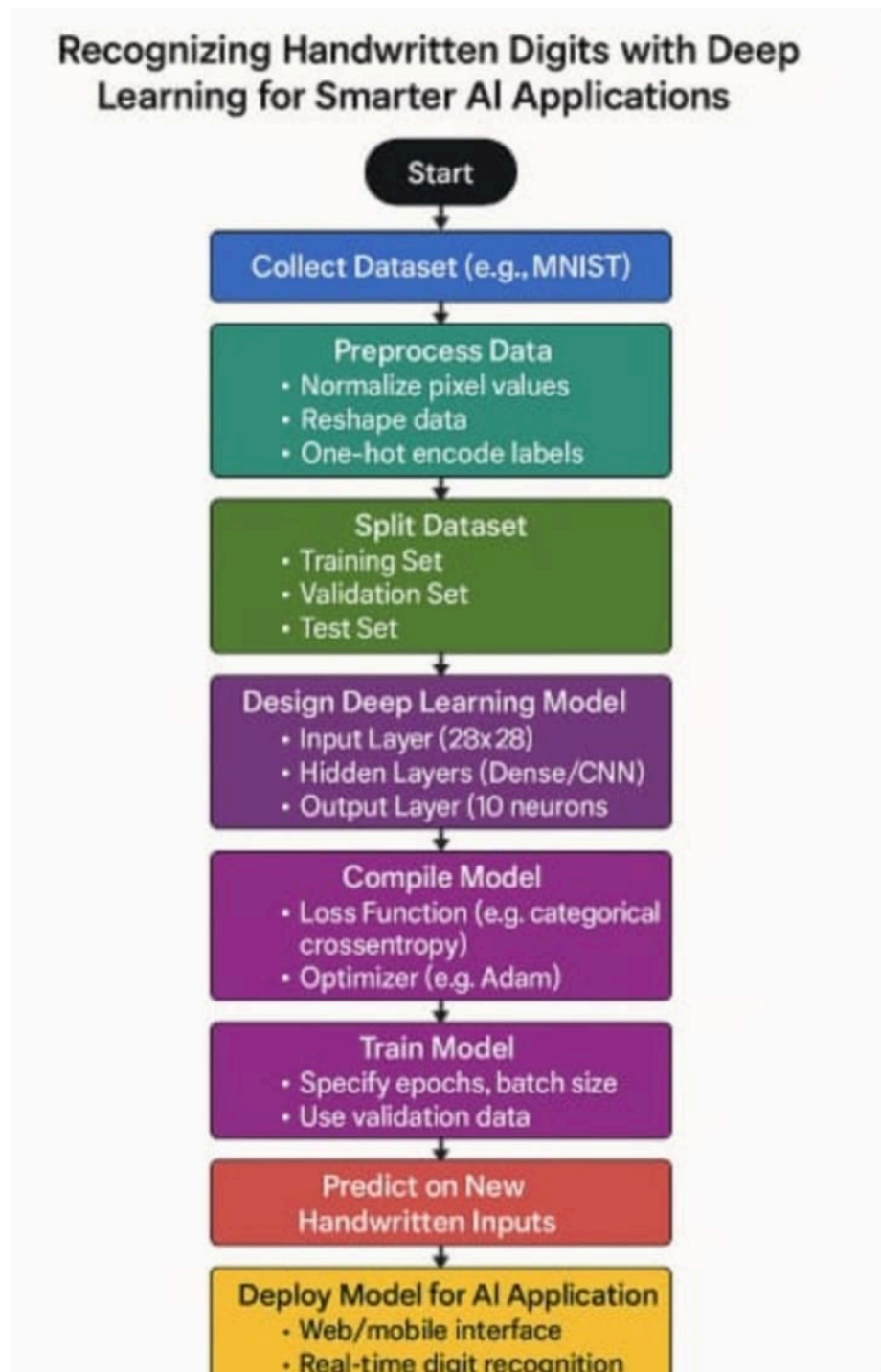
2. Project Objectives

To build a robust digit recognition model using deep learning.

To understand & visualize how deep learning interprets handwritten digits.

To evaluate the model's accuracy & optimize it using various techniques

To explore real-world applications of digit recognition

3. Flowchart of the Project Workflow

Recognizing Handwritten Digits with Deep Learning for Smarter AI Applications

Start

Collect Dataset (e.g., MNIST)

Preprocess Data
- Normalize pixel values
- Reshape data
- One-hot encode labels

Split Dataset
- Training Set
- Validation Set
- Test Set

Design Deep Learning Model
- Input Layer (28x28)
- Hidden Layers (Dense/CNN)
- Output Layer (10 neurons

Compile Model
- Loss Function (e.g. categorical crossentropy)
- Optimizer (e.g. Adam)

Train Model
- Specify epochs, batch size
- Use validation data

Predict on New Handwritten Inputs

Deploy Model for AI Application
- Web/mobile interface
- Real-time digit recognition

4.Data Description

Dataset: MNIST (Modified National Institute of Standards and Technology

Features: 28x28 grayscale images of digits (0–9)

Samples: 60,000 training and 10,000 testing image

Format: Each image is represented as a 784-dimensional vector (flattened)

5.Data Processing

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

Reshape images to (28, 28, 1) for CNN input

One-hot encode the target labels (0–9)

Split data into training, validation, and test sets

6. Exploratory Data Analysis (EDA)

Visualize sample images

Check class distribution & plot average image per digit

Analyze pixel intensity distribution

7. Feature Engineering

Image reshaping and scaling

Augmentation (rotation, shift, zoom) to improve generalization

Optional: PCA for dimensionality reduction (not commonly used with CNNs)

8. Model Building

Model Type: Convolutional Neural Network (CNN)

Architecture:

- Input → Conv2D → ReLU → MaxPooling
- Conv2D → ReLU → MaxPooling
- Flatten → Dense → Dropout → Output (Softmax)

Optimizer: Adam

Loss Function: Categorical Crossentropy

Metrics: Accuracy

9. Visualization of Results & Model Insights

Plot training vs validation accuracy/loss curves

confusion matrix

Visualize correct and incorrect predictions

Use Grad-CAM or saliency maps to see what the model "sees"

10. Tools and Technologies Used

Programming Language: Python

Libraries: TensorFlow, Keras, NumPy, Pandas, Matplotlib, Seaborn

Environment: Jupyter Notebook / Google Colab

Others: scikit-learn for evaluation metrics

11. Team Members and Contribution

Team Leader:M.R. Tameema Naazmi

Responsible for overall project management, coordination among team members, setting milestones, and ensuring timely delivery.

 M. Ashwini – Data analyst Handles data preprocessing, visualization, and analysis. Prepares the dataset for training and testing.

K. Monisha – Deep Learning Engineer

Designs and develops the Convolutional Neural Network (CNN). Trains and optimizes the model for high accuracy.

G. Sharmila Devi – Software Developer

Develops the user interface & deploys the trained model using a web or mobile platform.

Pratiksha – QA Analyst :Tests the model's performance and accuracy…