**Phase-2**

**Student Name:** **Vennila R**

**Register Number:** 422023104059

**Institution:** Sri Rangapoopathi College of Engineering

**Department:** Computer Science and Engineering

**Date of Submission:** 03-05-2025

**Github Repository Link:** [**https://github.com/vennilara/NM-Phase-2-.git**](https://github.com/vennilara/NM-Phase-2-.git)

# Problem Statement

In real-world scenarios like automated form processing and postal address recognition, handwritten digit recognition remains a crucial challenge. This project aims to develop a robust deep learning model to accurately recognize handwritten digits from scanned images using convolutional neural networks (CNNs).

Type of Problem: Classification

Why It Matters: Accurate digit recognition enables smarter AI- powered applications, improves data entry automation, and reduces manual errors in digit processing systems

# 

# Project Objectives

* Build a deep learning model (CNN) to classify handwritten digits from 0-9.
* Achieve high accuracy and generalizability on unseen data.
* Compare performance with traditional machine learning models like SVM and Random Forest.
* Deploy the model in a scalable,application-ready form.

# Flowchart of the Project Workflow

# 

# Data Description

* + - Datase Name: Handwritten English Characters and Digits Datase
    - Source: <https://www.kaggle.com/datasets/sujaymann/handwritten-english-characters-and-digits>
    - Type of Data: Image
    - Format: Grayscale images of characters and digits
    - Records: Thousands of labeled character/digit samples
    - Target Variable: Digit (0-9)
    - Static or Dynamic: Static

# Data Preprocessing

* Normalized pixel values (0-1 scale)
* Resized images to uniform dimensions (e.g., 28x28 pixels)
* Encoded labels
* Split into training and testing set (e.g.80:20)

# Exploratory Data Analysis (EDA)

* Visualized digit distribution using countplots
* Displayed sample images
* Checked image dimensions, brightness, and noise patterns
* Summary: Balanced dataset with consistent image format, minor noise handled via normalization

# Feature Engineering

* + - * None explicitly required due to raw pixel input to CNN
      * Flattened image arrays for traditional ML models
      * Optional: PCA for dimension reduction in ML models (not CNN)

# Model Building

* Models Used:
* Convolutional Neural Network (CNN)
* Support Vector Machine (SVM) for comparison
* Why These Models:
* CNNs are state-of-the-art for image classification
* SVMs are strong traditional baselines

Evaluation Metrics: Accuracy, Precision, Recall, F1-Score Train/Test Split: 80/2

# Visualization of Results & Model Insights

* + Confusion matrix to detect misclassifications
  + Accuracy and loss curves across epochs
  + Sample predictions vs. actual values
  + CNN achieved over 98% accuracy, outperforming traditional models

# Tools and Technologies Used

* + Language: Python
  + IDE/Notebook: Google Colab
  + Libraries: TensorFlow/Keras, NumPy, Matplotlib, Scikit-learn,Seaborn
  + Visualization Tools: Matplotlib, Seaborn

# Team Members and Contributions

* Vennila R-Data Cleaning and Preprocessing
* Vaishnavi P- EDA
* Vijayalakshmi R- CNN Model Development
* Varalakshmi N- Evaluation and Cleaning and Reporting