**Phase-2**

**Student Name:**M.Hamsalekha

**Register Number:** 620123243016

**Institution:** AVS Engineering college

**Department:** B.Tech . Artificial intelligence and data science

**Date of Submission:** 10.05.2025

**Github Repository Link:**



# Problem Statement

The goal is to build a deep learning model that can recognize handwritten digits (0–9) from images. This is a supervised multi-class classification problem.

Accurate digit recognition enables automation in banking, postal services, education, and more. It’s a key step toward smarter AI systems that can interpret human handwriting and streamline data entry tasks.

# Project Objectives

The objective is to build a CNN-based model to accurately classify handwritten digits on the MNIST (Modified National Institute of Standards and Technology) dataset. Key technical goals include effective preprocessing, model training, and enabling real-time prediction. After data exploration, the focus has expanded to include improving interpretability due to challenges in distinguishing similar digits.

**3. Flowchart of the Project Workflow**

Start -- Collect daraset -- Preprocess data -- Built CNN model -- Train the model --Evaluate the model -- Test with new inages -- Integrate into smart AI applications -- Emd

# 4.Data Description

* *Dataset name and origin: MNIST dataset and libraries such as TensorFlow and PyTorch.*
* *Type of Data: Image data (unstructured)*

*Grayscale images of handwritten digits.*

* *Number of Records & Features:*

*70,000 total images: 60,000 for training, 10,000 for testing in MNIST dataset.*

*Each image is 28x28 pixels (784 features).*

* *Dataset:Static dataset*
* *Target Variable: Digit label (0–9), used for supervised classification.*

**5.Data Preprocessing**

1. *Reshaping Images : Convert 28x28 pixel images into a format suitable for the model*
2. *.Normalization : Scale pixel values from 0–255 to 0–1 by dividing by 255 to help the model train faster and more accurately.*
3. *One-Hot Encoding Labels : Transform digit labels (0–9) into one-hot vectors for multi-class classification.*
4. *Handling Duplicates & Missing Values : Check and remove any duplicate or missing records (though MNIST usually has none).*
5. *.Data Type Conversion : Ensure pixel data is in float32 and labels are integers.*
6. *Standardization (Optional) : mostly for non-image data*
7. *.Train-Test Split : Split data into training and testing sets to evaluate model performance.*

# 6. Exploratory Data Analysis (EDA)

* *Univariate Analysis:*

*1.Pixel values mostly near 0; few high values represent strokes.*

*2.Labels are evenly distributed (balanced classes).*

*3.Boxplots show central pixels vary more—useful for recognition.*

* *Bivariate/Multivariate Analysis:*

*1.Adjacent pixels are moderately correlated.*

*2.Mean intensity varies by digit (e.g., ‘1’ has lower, ‘8’ higher).*

*3.Pixel vs label plots show consistent pattern*

* *Insights Summary:*

*1.Center pixels are more important.*

*2.Mean intensity can be a useful feature.*

*3.Balanced data supports good model learning.*

# 7. Feature Engineering

* *Added mean intensity and non-zero pixel count for stroke density.*
* *Created region-wise pixel sums to capture stroke location.*
* *Used binning to group pixel intensity levels.*
* *Applied PCA (optional) for dimensionality reduction.*

Justification:These features highlight digit structure and reduce noise, improving model accuracy.

# 8. Model Building

* *Logistic Regression and Random Forest are used.*
* *Logistic Regression:Fast , predictable and good for understanding the patterns.*
* *Random Forest :Suitable for image data flattened into high-dimensional vectors.*
* *Logistic Regression and Random Forest achieved ~92% and ~96% accuracy. Both were evaluated using accuracy, precision, recall, and F1-score.*

# 9. Visualization of Results & Model Insights

1. *Confusion Matrix shows where models misclassify digits.*
2. *ROC Curve reveals class-wise prediction quality .*
3. *Feature Importance (RF) highlights which pixels impact predictions most.*
4. *Bar Charts compare accuracy and F1-score: Random Forest outperforms Logistic Regression overall.*

# 10. Tools and Technologies Used

* Programming Language: Python
* IDE/Notebook: Jupyter Notebook
* Libraries: pandas, numpy , matplotlib/seaborn, scikit-learn, TensorFlow
* Visualization Tools: Matplotlib,seaborn,scikit-learn

# 11. Team Members and Contributions

1. *D.Thanuja - Data cleaning and EDA*
2. *S.Pooja - Feature engineering and Model development*
3. *M.Hamsalekha - Documentation and reporting*