**DATA SCIENCE**

**Student Name** : M.Reemas

**Register Number:** 613023243041

**Institution:** Vivekanandha College of Technology For Women

**Department:** AI&DS

**Date of Submission:** 03.05.2025

**Github Repository Link:** <https://github.com/Reemasgif04/Data-science.git>

* **Problem Statement**

Recognizing handwritten digits is a classical computer vision task with applications in automation, postal address recognition, check processing, and smart forms. The challenge lies in accurately identifying digits (0–9) written in various styles by different individuals. This project addresses the need for reliable and scalable handwriting recognition systems using deep learning techniques. It is a multi-class classification problem where the objective is to predict a digit (0–9) from an image input.

* **Abstract**

The project focuses on building a deep learning model capable of recognizing handwritten digits from images, using the MNIST dataset. The goal is to automate digit recognition for use in real-world AI applications such as smart document processing and digital form reading. We utilize Convolutional Neural Networks (CNNs) due to their excellent performance in image-related tasks. After preprocessing and feature scaling, multiple models are trained and evaluated. The final CNN model achieves high accuracy, and is deployed as an interactive web application using Streamlit. The outcome is a functional prototype that demonstrates AI’s ability to interpret handwritten input, contributing to more intelligent automation systems.

* **System Requirements**

Hardware:

Minimum: 4 GB RAM, Intel i3 Processor

Recommended: 8+ GB RAM, GPU (NVIDIA CUDA) for faster training

Software:

Python: 3.7+

IDE: Google Colab / Jupyter Notebook

Libraries:

numpy, pandas, matplotlib, seaborn

tensorflow / keras, scikit-learn

streamlit for deployment

* **Objectives**

Build a deep learning model to classify handwritten digits (0–9).

Achieve over 98% accuracy on the MNIST test dataset.

Deploy the trained model as a web application for real-time digit recognition.

Demonstrate the feasibility of using deep learning in smart AI systems that interact with human handwriting inputs.

* **Flowchart of Project Workflow**

Data Collection → Preprocessing → EDA → Feature Engineering → Modeling → Evaluation → Deployment

* **Dataset Description**

Source: Kaggle MNIST Dataset

Type: Public, Real-world data

Size:

60,000 training samples

10,000 test samples

784 features per sample (28x28 pixels flattened) + 1 target label

* **Data Preprocessing**

Missing values: None

Duplicates: Removed if any

Outliers: Not applicable (image data)

Feature Scaling: Pixel values normalized (0–255 scaled to 0–1)

Label Encoding: Not needed (digits 0–9 already numeric)

* **Exploratory Data Analysis (EDA)**

Visualized class distribution to ensure balanced data

Plotted average digit images per class

Used heatmaps and pixel value histograms to identify common patterns

Key Takeaways:

Data is well-balanced across all digit classes

Some digits like ‘1’ and ‘7’ show more variance in handwriting

Central pixel regions carry most of the discriminative information

* **Feature Engineering**

Reshaped images to 28x28 for CNN input

Created additional channels for grayscale image (1 channel)

No manual feature creation, CNNs learn spatial features automatically

* **Model Building**

Baseline Model: Logistic Regression, Multilayer Perceptron (MLP)

Advanced Model: Convolutional Neural Network (CNN)

Layers: Conv2D → MaxPooling → Dropout → Dense

Optimizer: Adam

Loss: Categorical Crossentropy

Epochs: 10–20

* **Model Evaluation**

Metrics Used:

Accuracy

Precision, Recall, F1-Score

Confusion Matrix

ROC AUC (for multi-class)

**Final Results:**

CNN Test Accuracy: ~99.2%

CNN outperforms MLP and Logistic Regression significantly

* **Deployment**

Method: Streamlit + Hugging Face Spaces

Platform: Streamlit Cloud

Public URL: [Insert link here]

Functionality: Upload image or draw digit, get prediction instantly

* **Source code**

Includes:

Data loading and preprocessing scripts

Model training notebooks

Streamlit app code for deployment

* **Future scope**

Expand to Alphabet Recognition:

Extend model to recognize handwritten letters (A–Z) using EMNIST.

Real-time Integration:

Integrate model into OCR pipelines and mobile scanning apps.

Model Compression:

Optimize model for edge devices using quantization or pruning techniques.

**13. Team Members and Roles**

B.Pavitha

Team Lead

S.Punitham

Coordinator

M.Rithicka

Developer