

DIGITS PREDICTING HANDWRITTEN PREDICTION

In partial fulfillment of the Requirements for the completion of VI

Semester of

EMERGING OF TECHNOLOGY

In

CSE (AI&ML)

Submitted By

M. GAYATHRI DEVI (22WJ8A6637)

M. BHAVANI (22WJ8A6638)

M. SAIKIRAN(22WJ8A6639)

M. NIKHIL REDDY(22WJ8A6640)



DEPARTMENT OF CSE (AI&ML)

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS

ABSTRACT

Handwritten digit prediction is an important task in machine learning that focuses on recognizing numbers written by hand, typically from images. This project uses the MNIST dataset, a well-known collection of 28x28 pixel grayscale images of digits ranging from 0 to 9. The primary goal is to train a model that can accurately identify which digit is present in a given image. To achieve this, machine learning techniques such as Convolutional Neural Networks (CNNs) are used, as they are highly effective for image classification tasks. The model learns from thousands of labeled examples, extracting features and patterns to make predictions on new, unseen data. This technology has practical applications in fields like postal code recognition, bank check processing, and form automation. The project demonstrates the process of preprocessing image data, training the model, testing its accuracy, and deploying it for real-world usage.

Literature Survey

2.1 Existing Work on Digits predicting handwritten prediction:

1. MNIST Dataset (The Benchmark)

The **MNIST (Modified National Institute of Standards and Technology)** dataset is the most widely used dataset for digit recognition. It contains:

- 70,000 grayscale images of handwritten digits (0–9)
 - 60,000 for training
 - 10,000 for testing
- Each image is 28x28 pixels

The MNIST dataset is often considered the “Hello World” of deep learning.

Key Approaches & Algorithms Used:

A. Traditional Machine Learning Models

Before deep learning took over, these algorithms were popular:

- **Support Vector Machines (SVM)**
- **K-Nearest Neighbors (KNN)**
- **Logistic Regression**
- **Decision Trees and Random Forests**

These models require **feature extraction** (like Histogram of Oriented Gradients, zoning, projection histograms, etc.).

B. Deep Learning Models (More Recent)

Deep learning models automatically learn features from data:

1. Convolutional Neural Networks (CNNs)

- Most successful model for digit recognition.
- Architecture usually includes convolutional layers, pooling layers, fully connected layers, and softmax for classification.
- **LeNet-5** by Yann LeCun (1998) is one of the earliest CNNs designed for digit recognition.

2. Multilayer Perceptrons (MLPs)

- A basic feedforward neural network with fully connected layers.
- Less effective than CNNs due to no spatial feature extraction.

3. Transfer Learning

- Though less common for MNIST due to its simplicity, models pre-trained on larger datasets can be fine-tuned.

Accuracy Benchmarks

- Traditional ML (SVM, KNN): ~94%–97% accuracy
- CNNs: ~99.3% or higher with proper tuning and data augmentation
- State-of-the-art models using ensemble techniques or deeper networks can go beyond 99.7%

2.2 Challenges in Current Approaches:

1. Limited Real-World Generalization

- **MNIST is too clean** – it's been preprocessed and centered.
- Real-world handwritten digits (e.g., from forms or touchscreens) vary significantly in:
 - Orientation
 - Thickness
 - Background noise
 - Lighting conditions
- Models trained only on MNIST may fail on more realistic datasets like **EMNIST** or **SVHN**.

2. Overfitting on Benchmark Datasets

- Many models achieve **>99% on MNIST**, but this leads to **overfitting to the benchmark** rather than solving broader handwriting recognition.
- Models may not generalize to:
 - Digits in different languages or writing styles (e.g., Devanagari, Arabic)
 - Handwritten characters beyond digits.

3. Style Variation Among Writers

- **High inter-writer variability**: Some people write digits in very stylized or unusual ways.
- CNNs sometimes fail to classify when stroke patterns deviate from the average.

4. Rotation, Scale, and Shift Sensitivity

- Small **rotations** or **shifts** in digits can throw off prediction.
- CNNs aren't naturally invariant to transformations unless trained with **data augmentation**.

5. Lack of Explainability

- Deep learning models (especially CNNs) are **black boxes**.
- For applications in education, security, or healthcare, **explainable AI** is crucial but still limited.

6. Computational Requirements

- Complex models (like deep CNNs or ensembles) demand:
 - High computational power (GPU/TPU)
 - Long training times
 - Energy resources
- Not ideal for edge devices (e.g., handheld scanners, low-power devices)

7. Handling Noisy or Incomplete Data

- In real-world scenarios, digits may be:
 - Partially visible
 - Blurry
 - Written over
- Models need to be more robust to such issues.

8. Bias in Training Data

- If the training dataset lacks diversity in handwriting styles (e.g., mostly English, adult handwriting), the model becomes **biased**.
- This affects performance across age groups, cultures, or different educational backgrounds.

2.3 Motivation for the Project

The growing integration of artificial intelligence in day-to-day applications has sparked immense interest in the field of pattern recognition, particularly in the area of **handwritten digit recognition**. With the digital transformation of many industries, there is a growing need for intelligent systems that can **automate manual data entry**, improve **document digitization**, and enhance **human-computer interaction**.

Despite the availability of well-known datasets like MNIST and considerable progress in accuracy levels using machine learning and deep learning models, several **real-world challenges still remain**—including handwriting variability, noise, and generalization to unseen data. These issues highlight the **need for more robust, scalable, and adaptive models** that can perform well in diverse and dynamic environments.

This project is motivated by the following key factors:

- **Practical Use Cases:** Applications like postal code recognition, bank cheque verification, form processing, and mobile digit input systems require efficient and accurate digit recognition models.
- **Educational Value:** Developing a digit recognition system provides an excellent foundation for understanding key concepts in machine learning, image processing, and deep learning, especially CNNs.
- **Bridging the Gap:** While high performance has been achieved on standardized datasets, there remains a gap when deploying models in the wild. This project aims to explore this gap and contribute toward **generalizable solutions**.
- **Skill Development:** The project offers hands-on experience with tools like Python, TensorFlow/Keras or PyTorch, and real-world problem-solving through data preprocessing, model building, and evaluation.

In essence, the motivation behind this project is not only to replicate existing success but also to deepen understanding and explore **how these techniques can be adapted, improved, and scaled for practical applications**.

