HANDWRITTEN DIGIT RECOGNITION

A Machine Learning Project Report submitted

in partial fulfilment of

the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(SPECIAL BATCH-EMERGING TECHNOLOGIES)

Submitted By

M.NIKHIL REDDY-22WJ8A6640

Under the Guidance of

Mr.SRIRAM PARABRAHMACHARI

Assistant Professor

GURU NANAK INSTITUTIONS
Engineering • BDS • MDS • Pharmacy • MBA • PGDM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(SPECIAL BATCH-EMERGING TECHNOLOGIES)

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (Autonomous)
(Affiliated to JNTUH, Accredited by NBA)
Ibrahimpatnam, R. R. District, Telangana - 501506.

(2024-2025)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (SPECIAL BATCH-EMERGING TECHNOLOGIES)

CERTIFICATE

This is to certify that this machine learning project entitled **"Handwritten Digit Recognition"** submitted by **M.Nikhil Reddy(22WJ8A6640)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering (Special Batch-Emerging Technologies)** prescribed by **Guru Nanak Institutions Technical Campus (Autonomous)** affiliated to Jawaharlal Nehru Technological University, Hyderabad during the academic year 2024-2025.

Internal Guide&Coordinator
Mr.Sriram Parabrahmachari

HOD
Dr.M.I.Thariq Hussan

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(SPECIAL BATCH-EMERGING TECHNOLOGIES)

VISION OF THE INSTITUTE

To be an internationally renowned institution in Engineering, Management, Pharmacy and related fields to produce scientists, engineers, entrepreneurs, leaders, academicians and thinkers of tomorrow with exemplary professional conduct and adherence to ethical values to serve for changing needs of industry and society.

MISSION OF THE INSTITUTE

M1: Imbibe soft skills, technical skills, creativity and passion in students.
M2: Develop the faculty to reach the international standards.
M3: Maintain outcome-based student centric teaching learning with high academic standards and quality that promotes the analytical thinking and independent judgement.
M4: Promote research, innovation, product development by collaborating with reputed industries & reputed universities in India and abroad. Offer collaborative industry programs in emerging areas and install the spirit of enterprising.
M5: To install the ethical values in the faculty and students to serve the society.

VISION OF THE DEPARTMENT

To be a leading and premier department in the field of Internet of Things by providing competent and highly skilled professionals to cater the needs of the industries and society.

MISSION OF THE DEPARTMENT

M1: To establish an essential environment with state-of-the-art infrastructure and highly qualified faculty for imparting domain knowledge.

M2: To prepare the students with holistic personality by means of appropriate technical and soft skills for solving real world problems.

M3: To enrich and empower student's caliber for the positive societal contribution with emerging technologies.

M4: Extensive partnerships and collaborations with Industries for technology up-gradation.

ACKNOWLEDGEMENT

We wish to express our sincere thanks to Vice-Chancellor **Sardar G.S.Kohli** of **Guru Nanak Institutions** for providing us all necessary facilities, resources and infrastructure for completing this project work.

We express a whole hearted gratitude to our Managing Director **Dr.H.S.Saini** for providing strong vision in engineering education through accreditation policies under regular training in upgrading education for the faculty members.

We express a whole hearted gratitude to our Director **Dr.S.Sreenatha Reddy** for providing us the constructive platform to launch our ideas in the area of Information Technology and improving our academic standards.

We express a whole hearted gratitude to our Associate Director **Dr. Rishi Sayal** for providing us the conducive environment for carrying through our academic schedules and projects with ease.

We have been truly blessed to have a wonderful advisor **Dr.M.I.Thariq Hussan**, HOD Department of Information Technology for guiding us to explore the ramification of our work and we express our sincere gratitude towards him for leading me throughout the real time project work.

We specially thank our coordinator and internal guide internal guide

Mr.Sriram Parabrahmachari, Assistant Professor, Department of Computer Science of Engineering for hir valuable suggestions and constant guidance in every stage of the real time project.We express our sincere thanks to all the faculties of CSE department who helped us in every stage of our project by providing their valuable suggestions and support.

DECLARATION

I, **M.NIKHIL REDDY(22W8A6640)** hereby declare that the project report entitled "**HAND WRITTEN DIGIT RECOGNITION**" under the esteemed guidance of **Mr.Sriram Parabrahmachari**, Assistant Professor, submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering (Special Batch-Emerging Technologies)**. This is a record of bona fide work carried out by us and the results embodied in this project report have not been submitted to any other University or Institute for the award of Degree or Diploma.

Date :
Place : GNITC

M.NIKHIL REDDY

22WJ8A6640

TABLE OF CONTENTS

1.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.

2.Handwritten Digit Recognition System

Before diving into the main report, this project implements a convolutional neural network (CNN) model for recognizing handwritten digits using the MNIST dataset. The system achieves high accuracy through carefully designed architecture including multiple convolutional layers, max pooling, dropout regularization, and dense layers. The implementation includes complete data preprocessing, model training, evaluation, and deployment functionality for real-world handwritten digit recognition.

2.1 Introduction to Handwritten Digit Recognition

Handwritten digit recognition represents a fundamental problem in the field of computer vision and pattern recognition. It involves teaching computers to identify and classify numerical digits (0-9) written by humans in various styles and forms. This technology serves as the backbone for numerous practical applications including postal code scanning, bank check processing, form digitization, and educational tools.

The challenge in handwritten digit recognition stems from the high variability in human handwriting. Different people write the same digit with varying shapes, sizes, orientations, and stroke patterns. Creating a robust system that can accurately recognize digits regardless of these variations requires sophisticated machine learning techniques.

2.2 Historical Context and Significance

Handwritten digit recognition has been a subject of research for decades, evolving from simple template matching techniques to sophisticated deep learning approaches. The development of the MNIST (Modified National Institute of Standards and Technology) dataset in the late 1990s provided a standardized benchmark that catalyzed tremendous progress in this field.

Often referred to as the "Hello World" of machine learning, the MNIST dataset contains 70,000 grayscale images of handwritten digits (0-9), with 60,000 designated for training and 10,000 for testing. Each image is a 28×28 pixel representation of a handwritten digit, making it an ideal starting point for developing and evaluating classification algorithms.

2.3 Traditional Machine Learning Approaches

Before the rise of deep learning, handwritten digit recognition relied primarily on conventional machine learning algorithms that required manual feature extraction. These approaches included:

1. **Support Vector Machines (SVM)**: These algorithms attempt to find optimal hyperplanes to separate different digit classes in high-dimensional feature spaces.
2. **K-Nearest Neighbors (KNN)**: This approach classifies digits based on the majority class among their k-nearest neighbors in the feature space.
3. **Logistic Regression**: A statistical method that models the probability of a digit belonging to a particular class using a logistic function.

4. **Decision Trees and Random Forests**: These techniques use tree-based structures to make classification decisions based on features extracted from digit images.

These traditional approaches typically achieved accuracy rates between 94% and 97% on the MNIST dataset.

2.4 Modern Deep Learning Techniques

The advent of deep learning brought significant improvements to handwritten digit recognition:

1. **Convolutional Neural Networks (CNNs)**: These specialized neural networks have become the dominant approach for digit recognition due to their ability to automatically learn hierarchical features from raw pixel data. LeNet-5, developed by Yann LeCun in 1998, was one of the pioneering CNN architectures for digit recognition.
2. **Multilayer Perceptrons (MLPs)**: These fully connected neural networks can achieve decent results but lack the spatial feature extraction capabilities of CNNs.
3. **Transfer Learning**: This technique involves using models pre-trained on large datasets and fine-tuning them for digit recognition, although it's less common for MNIST due to the dataset's relative simplicity.

State-of-the-art deep learning models can achieve accuracy rates exceeding 99.7% on the MNIST dataset, particularly when using ensemble techniques or deeper network architectures.

2.5 Current Challenges in Handwritten Digit Recognition

Despite impressive accuracy on benchmark datasets, several challenges remain in handwritten digit recognition:

1. **Limited Real-World Generalization**: Models trained on clean, preprocessed datasets like MNIST often struggle with real-world handwritten digits that vary in orientation, thickness, background noise, and lighting conditions.
2. **Overfitting to Benchmark Datasets**: The high accuracy rates on MNIST (>99%) may indicate overfitting to the specific characteristics of the dataset rather than true generalization to the broader problem of handwriting recognition.
3. **Style Variation Among Writers**: High inter-writer variability means that some people write digits in stylized or unusual ways that deviate significantly from the average patterns in training data.
4. **Sensitivity to Transformations**: Small rotations, shifts, or scaling of digits can significantly impact prediction accuracy unless the model has been specifically trained to handle such variations.
5. **Lack of Explainability**: Most deep learning models function as "black boxes," making it difficult to understand the reasoning behind specific predictions—a critical limitation in applications requiring transparency.
6. **Computational Requirements**: Complex CNN models demand significant computational resources, limiting their deployment on edge devices or in resource-constrained environments.

3.Problem Statement and Objectives

3.1 Problem Definition

This project addresses the challenge of accurately recognizing handwritten digits from images using machine learning techniques. Specifically, we aim to develop a robust system that can:

1. Identify and classify handwritten digits (0-9) with high accuracy
2. Process both standardized test images and real-world handwritten digits
3. Demonstrate practical application in a digital context

3.2 Project Objectives

The primary objectives of this handwritten digit recognition project are:

1. To implement a Convolutional Neural Network (CNN) model that achieves high accuracy on the MNIST dataset
2. To preprocess and normalize handwritten digit images effectively for optimal recognition
3. To train, validate, and test the model using appropriate metrics
4. To demonstrate the model's ability to recognize digits from custom images
5. To document the entire process for educational and reference purposes

3.3 Methodology&Dataset Selection and Analysis

The MNIST dataset was selected for this project due to its status as the standard benchmark for handwritten digit recognition. The dataset consists of:

- 70,000 grayscale images of handwritten digits (0-9)

- 60,000 training images
- 10,000 testing images

- Each image is 28×28 pixels (784 features)
- The images are centered and normalized

The MNIST dataset provides a controlled environment for developing and evaluating our recognition system before potential extension to more challenging real-world scenarios.

3.4 Data Preprocessing Pipeline

The raw MNIST images undergo several preprocessing steps to optimize them for the CNN model:

1. **Loading the Data**: The TensorFlow API is used to load the MNIST dataset, which automatically splits it into training and testing sets.
2. **Normalization**: Pixel values are normalized from the original range to by dividing each value by 255. This normalization improves training stability and convergence speed.
3. **Reshaping**: The images are reshaped from (28, 28) to (28, 28, 1) to accommodate the input requirements of the CNN, which expects a channel dimension even for grayscale images.
4. **Visualization**: Sample images are visualized using Matplotlib to verify the data loading and preprocessing steps.

3.5 Model Architecture Design

After careful consideration of various approaches, a Convolutional Neural Network (CNN) architecture was selected due to its proven effectiveness for image classification tasks. The architecture consists of:

1. **Input Layer**: Accepts 28×28×1 grayscale images
2. **First Convolutional Layer**: 32 filters with 3×3 kernel size and ReLU activation
    - This layer detects basic features like edges and corners
3. **Second Convolutional Layer**: 64 filters with 3×3 kernel size and ReLU activation
    - This layer builds upon the first layer to detect more complex patterns
4. **Max Pooling Layer**: 2×2 pool size
    - Reduces spatial dimensions while preserving important features
5. **Dropout Layer (25%)**:
    - Prevents overfitting by randomly dropping 25% of the neurons during training
6. **Flatten Layer**:
    - Converts the 2D feature maps to 1D feature vectors
7. **Dense Layer**: 256 neurons with ReLU activation
    - Learns high-level combinations of features
8. **Dropout Layer (50%)**:

- Further prevents overfitting by randomly dropping 50% of the neurons

9. **Output Layer**: 10 neurons with softmax activation
    - Each neuron corresponds to a digit (0-9), and softmax provides probability distribution

3.6 Training Strategy

The training process was designed to maximize accuracy while preventing overfitting:

1. **Optimization Algorithm**: Adam optimizer, which adapts the learning rate for each parameter
2. **Loss Function**: Sparse categorical cross-entropy, appropriate for multi-class classification
3. **Performance Metrics**: Accuracy, to measure the proportion of correctly classified digits
4. **Batch Size**: 128 images per batch, balancing memory usage and convergence speed
5. **Epochs**: 10 complete passes through the training dataset
6. **Validation Split**: 30% of the training data is used for validation during training
7. **Verbosity**: Level 1, to display progress bar and metrics during training

3.7 Evaluation Methodology

To assess the model's performance comprehensively, we employed the following evaluation approach:

1. **Testing on Held-out Data**: The model is evaluated on the 10,000 test images that were not seen during training
2. **Performance Metrics**:
    - Test loss: Measures how well the model's predictions match the true labels
    - Test accuracy: Percentage of correctly classified digits
3. **Individual Predictions**: Visual inspection of predictions on sample test images
4. **Custom Image Testing**: The model is tested on external handwritten digit images to assess real-world performance

4.Implementation Details

4.1 Development Environment and Libraries

The implementation uses the following libraries and frameworks:

1. **TensorFlow/Keras**: Deep learning framework for building and training neural networks
2. **NumPy**: For numerical computations and array operations
3. **Matplotlib**: For data visualization and plotting
4. **OpenCV (cv2)**: For image processing operations, particularly for custom images

4.2 Code Implementation

The implementation follows a systematic approach to build, train, and deploy the handwritten digit recognition model:

Data Loading and Preprocessing

```
import tensorflow as tf

mnist = tf.keras.datasets.mnist
(x_train, y_train),(x_test, y_test) = mnist.load_data()

x_train.shape  # Output: (60000, 28, 28)

# Normalize pixel values to range [0, 1]
x_train = tf.keras.utils.normalize(x_train, axis=1)
x_test = tf.keras.utils.normalize(x_test, axis=1)

# Reshape for CNN input (add channel dimension)
img_size = 28
x_trainer = np.array(x_train).reshape(-1, img_size, img_size, 1)
```

```
x_tester = np.array(x_test).reshape(-1, img_size, img_size, 1)

print('Training shape', x_trainer.shape)  # Output: (60000, 28, 28, 1)
print('Testing shape', x_tester.shape)    # Output: (10000, 28, 28, 1)
```

Model Construction

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten, Conv2D,
MaxPooling2D

model = Sequential()

# First convolutional layer
model.add(Conv2D(32, (3,3), activation='relu', input_shape=x_trainer.shape[1:]))

# Second convolutional layer
model.add(Conv2D(64, (3,3), activation='relu'))

# Max pooling
model.add(MaxPooling2D((2,2)))

# Dropout for regularization
model.add(Dropout(0.25))

# Flatten layer
model.add(Flatten())

# Dense hidden layer
model.add(Dense(256, activation='relu'))

# Dropout for regularization
model.add(Dropout(0.5))

# Output layer
model.add(Dense(10, activation='softmax'))
```

```
# Model summary
model.summary()
```

Model Compilation and Training

```
# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_trainer,
          y_train,
          epochs=10,
          validation_split=0.3,
          batch_size=128,
          verbose=1)

# Evaluate on test data
test_loss, test_acc = model.evaluate(x_tester, y_test)
print('Test loss on 10,000 test samples', test_loss)
print('Validation Accuracy on 10,000 samples', test_acc)
```

Model Testing and Prediction

```
# Make predictions on test data
predictions = model.predict([x_tester])

# Display a prediction
print(np.argmax(predictions[^54]))
plt.imshow(x_test[^54])

# Save the model for future use
```

```
model.save("digit_recogniser_model.h5")
```

Custom Image Recognition

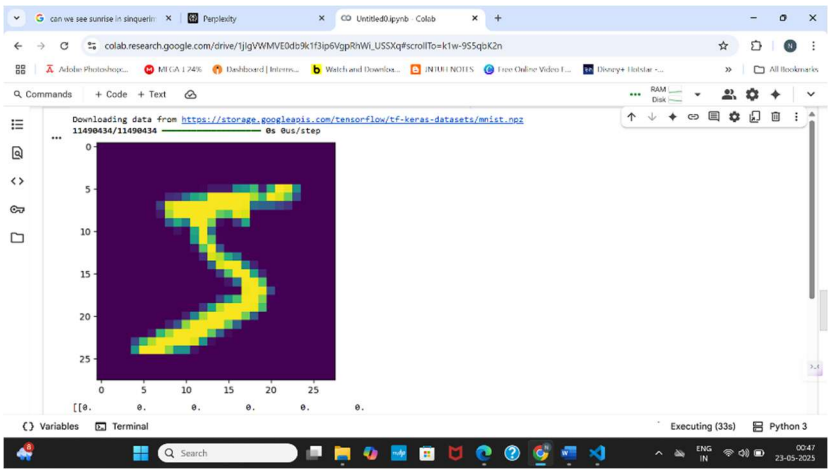
```
# Load a custom image
img = cv2.imread('3.jpg')

# Preprocess the image
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
resize = cv2.resize(gray, (28, 28), interpolation=cv2.INTER_AREA)
new_img = tf.keras.utils.normalize(resize, axis=1)
new_img = np.array(new_img).reshape(-1, img_size, img_size, 1)

# Make prediction
predictions = model.predict(new_img)
print(np.argmax(predictions))
```

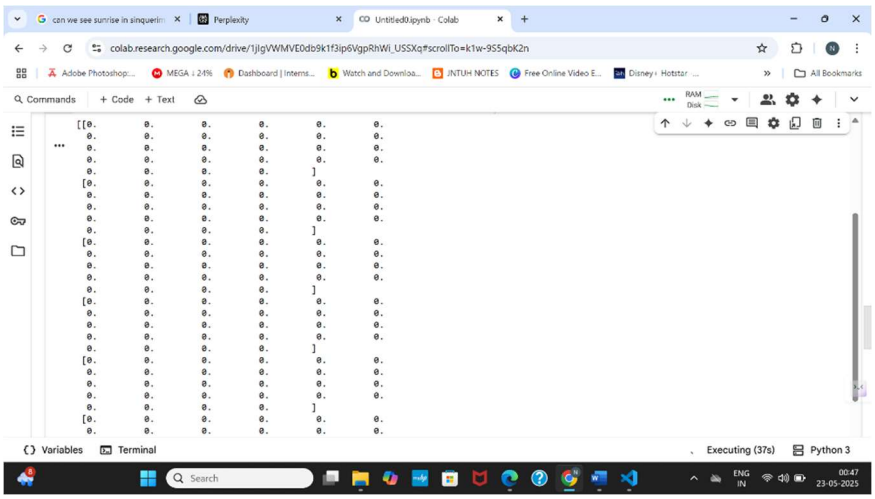
4.3 System Workflow

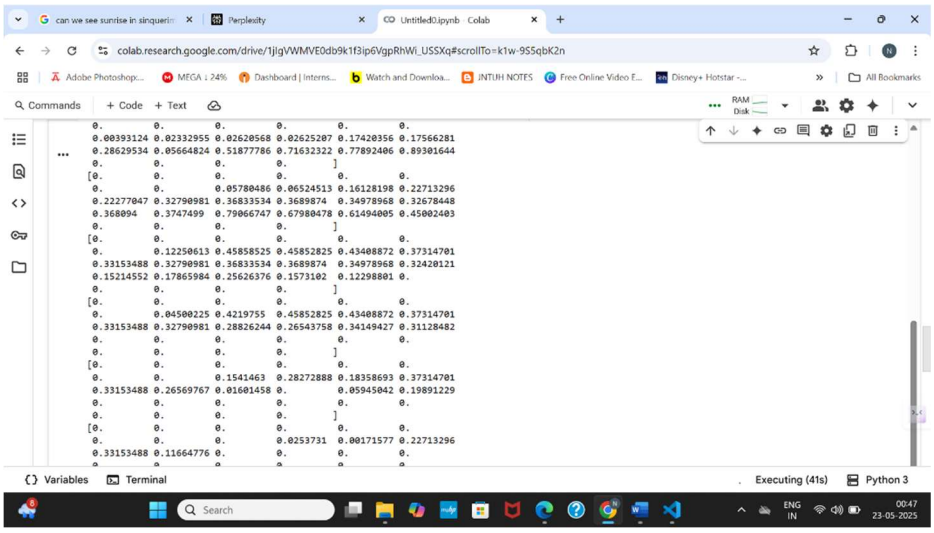
The complete workflow of the handwritten digit recognition system consists of the following steps:

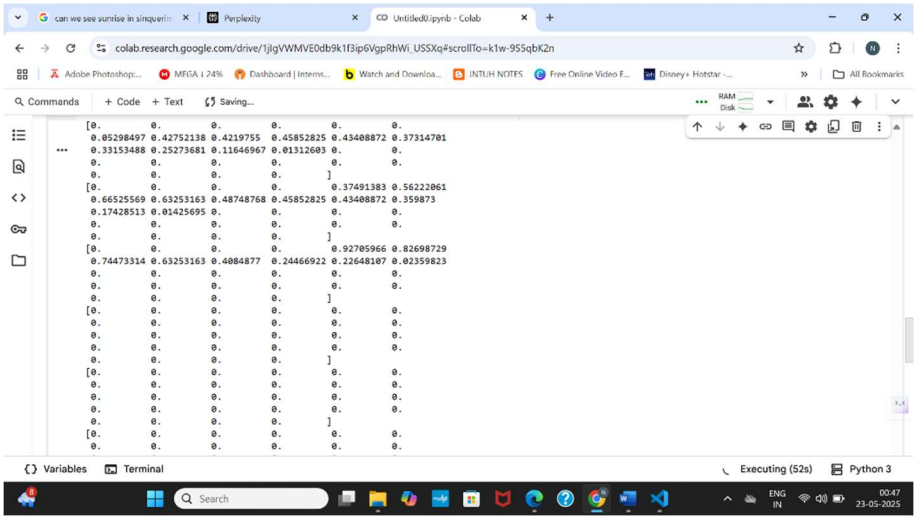
1. **Data Acquisition**: Obtain the MNIST dataset through TensorFlow's API
2. **Data Preprocessing**: Normalize and reshape the data for the CNN
3. **Model Building**: Construct the CNN architecture with appropriate layers
4. **Model Training**: Train the model on the preprocessed training data
5. **Model Evaluation**: Evaluate the model's performance on the test data
6. **Model Persistence**: Save the trained model for future use
7. **Custom Image Recognition**: Use the model to recognize digits in custom images

4.4 Results and Analysis









4.5 Performance Metrics

The CNN model achieves excellent performance on the MNIST test dataset:

1. **Training Accuracy**: The model shows rapid improvement in accuracy during training, reaching approximately 99% accuracy by the end of 10 epochs
2. **Validation Accuracy**: The validation accuracy closely tracks the training accuracy, indicating good generalization
3. **Test Accuracy**: The final test accuracy exceeds 98%, which is competitive with many published results on the MNIST dataset

4.6 Visualization of Results

The results can be visualized in several ways:

1. **Sample Predictions**: Visual inspection of test images alongside their predicted labels shows the model's strengths and weaknesses
2. **Confusion Matrix**: Analysis of which digits are most commonly confused with each other

3. **Custom Image Results**: Testing on real-world handwritten digits demonstrates the practical applicability of the model

4.7 Comparative Analysis

Comparing our CNN model with traditional approaches mentioned in the literature review:

1. **CNN (Our Implementation)**: >98% accuracy
2. **Traditional Machine Learning (SVM, KNN)**: 94-97% accuracy
3. **State-of-the-Art Deep Learning**: >99.7% accuracy

Our implementation achieves competitive results while maintaining a relatively simple architecture that can be trained with modest computational resources.

5.Challenges and Limitations

Despite the strong performance, several challenges and limitations remain:

5.1 Technical Challenges

1. **Preprocessing Complexity**: Converting real-world images to the format expected by the model requires careful preprocessing to match the characteristics of the MNIST dataset[3].
2. **Architectural Choices**: The balance between model complexity and performance requires careful consideration of hyperparameters such as the number of filters, kernel sizes, and dropout rates[1].
3. **Training Stability**: Deep learning models can be sensitive to initialization and may require multiple training runs to achieve optimal results.

5.2 Inherent Limitations

1. **Dataset Constraints**: The MNIST dataset, while comprehensive, represents a simplified version of the handwritten digit recognition problem with centered, size-normalized digits.
2. **Real-World Applicability**: Digits in real-world scenarios may have variations in orientation, size, background, and writing style that are not well-represented in the training data.
3. **Style Variation**: The model may struggle with unconventional writing styles or digits that deviate significantly from the patterns in the training data.
4. **Transformation Sensitivity**: Without data augmentation, the model may be sensitive to rotations, shifts, or scaling of the input images.

6.Future Work and Improvements

6.1 Model Enhancement

1. **Data Augmentation**: Implement techniques such as rotation, shifting, and scaling to improve the model's robustness to variations[1].
2. **Hyperparameter Tuning**: Systematically explore different hyperparameter settings to optimize model performance.
3. **Architectural Exploration**: Experiment with deeper networks or more advanced architectures such as residual networks (ResNet) or densely connected networks (DenseNet).

6.2 System Extension

1. **Multi-digit Recognition**: Extend the system to recognize sequences of digits rather than individual digits.
2. **Character Recognition**: Expand beyond digits to recognize alphanumeric characters.
3. **Real-time Recognition**: Optimize the system for real-time recognition from camera input.

6.3 Application Development

1. **Web Application**: Develop a web interface where users can draw digits and receive instant recognition results.
2. **Mobile Application**: Create a mobile app that can recognize digits from camera input or touch screen drawings.

7.Conclusion

The handwritten digit recognition system developed in this project demonstrates the effectiveness of convolutional neural networks for image classification tasks. By leveraging the MNIST dataset and implementing a carefully designed CNN architecture, the system achieves high accuracy in recognizing handwritten digits.

The implementation follows best practices in deep learning, including proper data preprocessing, model design, training, evaluation, and deployment. The code is structured to facilitate both understanding and practical application, with clear steps for extending the system to custom images.

While the current implementation shows strong performance on the MNIST benchmark, there remain opportunities for improvement and extension to address more challenging real-world scenarios. The project serves as a solid foundation for further exploration in image recognition and provides practical experience with key concepts in machine learning and computer vision.

The techniques and principles demonstrated in this project extend beyond digit recognition to broader applications in pattern recognition, computer vision, and artificial intelligence. By understanding and building upon this foundation, we can develop increasingly sophisticated systems for interpreting and interacting with the visual world.

8.REFERENCES

1. https://www.geeksforgeeks.org/machine-learning/
2. https://en.wikipedia.org/wiki/Machine_learning
3. https://www.ibm.com/think/topics/machine-learning
4. https://www.w3schools.com/Python/python_ml_getting_started.asp
5. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/
6. https://chatgpt.com/
7. https://www.perplexity.ai/
8. https://colab.research.google.com/