**<HANDWRITTEN DIGIT CLASSIFIER >**

**Submitted for**

**CSET211 - Statistical Machine Learning**

Submitted by:

**(<E23CSEU1203>) YASH PRATAP SINGH**

**(<E23CSEU1219>) ARNAV AGRAWAL**

**(<E23CSEU1213>) ARNAV AHLAWAT**

Submitted to:

**( Sir. Prashant Kapil)**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**Abstract:**

1. **Introduction**

A handwritten digit classifier is a system designed to recognize and classify handwritten digits (0–9) using machine learning. This project leverages a Convolutional Neural Network (CNN) model to perform digit recognition based on the popular MNIST dataset, which contains labeled images of handwritten digits. The classifier is trained to identify digits from input images and is capable of achieving high accuracy due to its powerful deep-learning architecture.

1. **Related Survey**

Handwritten digit recognition has been a prominent topic in machine learning and computer vision. Some key advancements include:

* **LeNet-5 (1998):** One of the first CNN architectures, proposed by Yann LeCun, specifically designed for handwritten digit recognition using the MNIST dataset.
* **Deep Learning:** Advances in deep neural networks, like CNNs, have significantly improved the performance of digit recognition models.
* **Applications:** Handwritten digit classifiers have found applications in postal address digitization, check processing in banks, and form digitization tasks.

The MNIST dataset has become the benchmark for evaluating these models, offering a large corpus of 70,000 labeled images of digits for training and testing.

1. **Datasets**

The dataset used is the **MNIST dataset**, which contains:

* **Training set:** 60,000 grayscale images of handwritten digits (28x28 pixels each).
* **Testing set:** 10,000 grayscale images for evaluation.

#### 3.1 Data Preprocessing

Steps for preprocessing:

1. **Normalization:** The pixel values (0–255) are scaled to a range of 0 to 1.
2. **Reshaping:** Images are reshaped to include a channel dimension (28x28x1) for compatibility with CNNs.
3. **One-hot Encoding:** Labels are converted into a categorical format with 10 classes (0–9).
4. **Methodology**

The model employs a Convolutional Neural Network (CNN) with the following architecture:

1. **Convolutional Layers:** Extract spatial features from input images.
2. **Max-Pooling Layers:** Reduce dimensionality and computation.
3. **Dropout:** Prevent overfitting during training.
4. **Dense Layers:** Classify features into the appropriate digit class.

#### 4.1 Hardware and Software Requirements

* **Hardware:** A system with a GPU is recommended for faster training.
* **Software:**
  + TensorFlow/Keras for model development.
  + Python for coding.
  + Libraries: NumPy, Matplotlib, PIL for preprocessing and visualization.
  1. **Performance Metrics**

1.  **Accuracy:** Percentage of correctly classified images.
2.  **Loss:** Quantifies the difference between predictions and actual label
3. **Results and Analysis**
4.  **Training Accuracy:** Consistently increased over epochs, indicating effective learning.
5.  **Validation Accuracy:** Approximately 99.2% after 10 epochs, suggesting excellent generalization.
6.  **Test Accuracy:** Achieved 99.2%, indicating strong performance on unseen data.
7.  **Visualization:** Example predictions of test images demonstrate accurate recognition with high confidence.

### 6. Conclusions and Future Work

#### Conclusions

* The CNN-based handwritten digit classifier achieved a high test accuracy of 99.2%.
* The MNIST dataset proved effective for training and testing the model.

#### Future Work

* **Enhanced Architectures:** Explore advanced models like ResNet or MobileNet for even higher accuracy.
* **Dataset Expansion:** Incorporate additional datasets (e.g., EMNIST) to make the model more robust.
* **Real-World Deployment:** Integrate the model into web or mobile applications for digit recognition.
* **Edge Computing:** Optimize the model for low-power devices for on-device digit recognition.
* This project demonstrates the potential of CNNs in practical handwritten digit classification and lays the groundwork for further exploration.

**\*\* Include the GitHub link as the last sentence of the Abstract for easy access to supplementary materials.**

<https://github.com/yashthakur234/SMLPROJECT/tree/main>

\*Above link contain code of my project/Model and Screenshots of output.