Pattern Recognition & Machine Learning
Handwritten Digit Recognition

TEAM DESCRIPTION

Navneet Meena (B21CS051)
Vishal Jharwal (B21EE077)
Harsh Nawal (B21AI015)
Rohan Kumar Duddeda(B21AI031)
Tushar Saini(B21EE071)

INTRODUCTION

The project aims to create a system that can accurately recognize handwritten digits using advanced pattern recognition and machine learning techniques. It involves gathering handwritten digit samples, processing them for analysis, and then developing and fine-tuning machine learning models.

METHODOLOGY OVERVIEW

- **Data Preprocessing:** The MNIST dataset was loaded and preprocessed to prepare it for training. This includes normalization and reshaping of the images.
- **Model Architecture:** The ResNet-18 architecture was selected for its proven effectiveness in image classification tasks. The structure of the ResNet-18 model was defined using PyTorch, including the convolutional layers, residual blocks, and fully connected layers.
- **Training:** The model was trained using a specified optimizer and loss function. Training parameters such as batch size, learning rate, and number of epochs are set to optimize the model's performance.
- **Evaluation:** The trained model was evaluated on a separate test set to assess its accuracy and performance in digit recognition. Metrics such as accuracy, precision, recall, and F1-score may be calculated to evaluate the model's performance.
- **Visualization:** Training metrics such as loss and accuracy were visualized using plots to analyze the model's learning progress over epochs.

DATASET

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.

It is a collection of 28x28 pixel grayscale images of handwritten digits (0-9).

Random image samples from the dataset

MODEL ARCHITECTURE

ResNet architecture was employed for the recognition task. It is a deep neural network architecture renowned for its ability to train very deep networks effectively. Its key innovation is the introduction of skip connections, which allow information to flow directly to deeper layers. This helps combat the vanishing gradient problem encountered in training deep networks. ResNet comprises residual blocks, each containing convolutional layers and shortcut connections. These shortcuts enable the network to learn residual mappings, making it easier to optimize even extremely deep models.

The ResNet-18 architecture was utilized, comprising 18 layers organized into convolutional blocks, each containing convolutional layers, batch normalization, rectified linear unit (ReLU) activation functions, and residual connections.

At its core were residual blocks, introducing skip connections to enable the model to bypass certain layers and learn residual mappings, rather than absolute transformations, making training deeper networks easier without encountering vanishing gradient problems.

The architecture began with a convolutional layer followed by a max-pooling layer to downsample the input. It contained four groups of residual blocks, each group gradually increasing the number of filters to capture more complex features. The residual blocks within each group shared the same number of filters to maintain consistency. Following the final group of residual blocks, global average pooling was applied to reduce the spatial dimensions of the feature maps, followed by a fully connected layer and a softmax activation function to produce class probabilities.

Overall, ResNet-18 combined deep convolutional layers with skip connections to effectively learn hierarchical representations of images, making it well-suited for tasks like digit recognition on datasets such as MNIST.

ResNet architecture was chosen for the handwritten digit recognition due to its effectiveness in learning complex image features, mitigating the vanishing gradient problem through residual connections, facilitating better training dynamics, and serving as a benchmark with demonstrated state-of-the-art performance in image classification tasks.

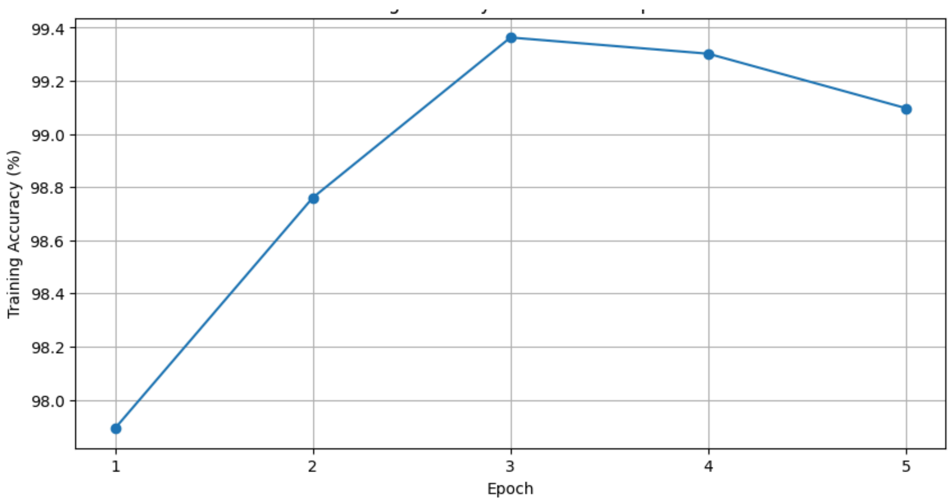
The complete model architecture visualization can be found **here**.

IMPLEMENTATION

The model was implemented in Python using the **PyTorch** library.

TRAINING

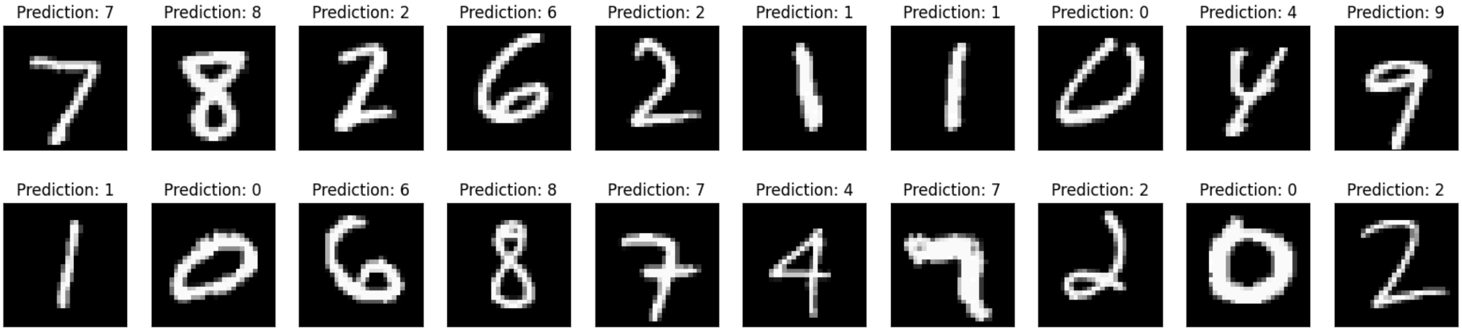
The model was trained using Adam optimizer to minimize the cross-entropy loss.



Training accuracy vs number of epochs curve

PREDICTION RESULTS

The model achieved an accuracy of **98.82%** on the test set.



Predictions for random image samples from the test set

REFERENCES

MNIST Dataset
https://en.wikipedia.org/wiki/MNIST_database

PyTorch Documentation
https://pytorch.org/docs/stable/index.html
https://pytorch.org/docs/stable/generated/torch.optim.Adam.html
https://pytorch.org/docs/stable/generated/torch.nn.functional.cross_entropy.html

ResNet Architecture
https://blog.paperspace.com/writing-resnet-from-scratch-in-pytorch/