**MNIST Digit Recognition using Convolutional Neural Networks (CNN)**

**Project Overview**

The MNIST Digit Recognition project leverages deep learning techniques to classify handwritten digits (0-9) from the MNIST dataset. This project implements a Convolutional Neural Network (CNN) using PyTorch, a popular machine learning framework, to achieve high accuracy in digit recognition. The CNN model is trained on the MNIST dataset, which consists of 60,000 training images and 10,000 test images of handwritten digits.

**Objectives**

* **Implement a CNN model** to classify handwritten digits accurately.
* **Utilize PyTorch** for building, training, and evaluating the model.
* **Visualize results** to understand the model's performance, including correct and incorrect predictions.

**Methodology**

1. **Data Preprocessing**:
   * The MNIST dataset is loaded and transformed to tensors. Each image is normalized to have pixel values in the range of [-1, 1].
   * Data loaders are created for both training and test datasets to facilitate batch processing.
2. **Model Architecture**:
   * A simple CNN architecture is designed with two convolutional layers, followed by max-pooling layers, and two fully connected layers. The model's architecture is as follows:
     + **Conv Layer 1**: 1 input channel (grayscale), 16 output channels, 3x3 kernel.
     + **Conv Layer 2**: 16 input channels, 32 output channels, 3x3 kernel.
     + **Max-Pooling Layer**: 2x2 window for down-sampling.
     + **Fully Connected Layer 1**: 128 outputs.
     + **Fully Connected Layer 2**: 10 outputs (for 10 digit classes).
     + ReLU activation functions are applied after each convolutional and fully connected layer to introduce non-linearity.
3. **Training**:
   * The model is trained using the Adam optimizer and Cross-Entropy loss function for classification.
   * The training process involves multiple epochs where the model learns to minimize the loss through backpropagation and gradient descent.
4. **Evaluation**:
   * After training, the model's accuracy is evaluated on the test dataset.
   * Predictions are made on test images, and visualization is performed to showcase the model's performance.
5. **Custom Image Prediction**:
   * The project includes functionality to load and preprocess custom images for prediction. Users can test the model's capability to recognize digits from their own handwritten images.

**Key Features**

* **High Accuracy**: Achieved significant accuracy on the MNIST test dataset.
* **Visualization of Results**: Displayed both correct and incorrect predictions for better insight into model performance.
* **Custom Image Support**: Users can input their own handwritten images for digit classification.

**Tools and Technologies**

* **Programming Language**: Python
* **Framework**: PyTorch
* **Libraries**:
  + torch for building and training the model.
  + torchvision for dataset handling and transformations.
  + matplotlib for data visualization and result presentation.
  + PIL (Python Imaging Library) for image processing.

**Conclusion**

The MNIST Digit Recognition project demonstrates the capabilities of CNNs for image classification tasks. Through the use of deep learning techniques and the PyTorch framework, the model is able to learn complex patterns in handwritten digits, resulting in high accuracy and efficiency in classification. This project serves as a foundational example for anyone interested in exploring deep learning and computer vision.