Project Title: Image Classifier Using Convolutional Neural Networks

---

Introduction

The **Image Classifier** project utilizes deep learning techniques to classify handwritten digits from the MNIST dataset. It demonstrates the application of Convolutional Neural Networks (CNNs) to extract features and perform accurate image classification tasks. This project serves as a foundational implementation for exploring image recognition and pattern detection.

Objective

To develop a robust CNN model capable of recognizing handwritten digits with high accuracy and demonstrating the practicality of deep learning in solving image classification problems.

Key Features

- **Dataset:** MNIST, a standard benchmark dataset for image classification.
- **Data Preprocessing:**
    - Normalized pixel values to the range [0, 1].
    - Reshaped images to include a single color channel (grayscale).
    - One-hot encoded class labels.
- **CNN Architecture:**
    - **Convolutional Layers**: Extract spatial features.
    - **Pooling Layers**: Reduce dimensions while preserving important features.
    - **Dense Layers**: Perform classification.
- **Output:** Predicts the class (digit) of each input image.

- **Performance Metrics:** Accuracy, loss evaluation on training and test datasets.

Technical Details

- **Programming Language:** Python 3
- **Libraries Used:** TensorFlow, Keras, NumPy, Matplotlib
- **System Requirements:**
    - Python 3.8 or higher
    - Compatible with Windows, macOS, and Linux
    - At least 4GB RAM (recommended for efficient processing)

Model Summary

- **Input Shape:** (28, 28, 1) for grayscale images.
- **Layers:**
    1. **Conv2D (32 filters, 3x3 kernel, ReLU activation)**
    2. **MaxPooling2D (2x2 pool size)**
    3. **Conv2D (64 filters, 3x3 kernel, ReLU activation)**
    4. **MaxPooling2D (2x2 pool size)**
    5. **Conv2D (64 filters, 3x3 kernel, ReLU activation)**
    6. **Flatten Layer**
    7. **Dense Layer (64 units, ReLU activation)**
    8. **Output Layer (10 units, softmax activation)**
- **Optimizer:** Adam
- **Loss Function:** Categorical Cross-Entropy
- **Metrics:** Accuracy

Results

- **Training Accuracy:** ~99%
- **Test Accuracy:** ~98%

- **Model Evaluation:** Achieved high accuracy on unseen test data, demonstrating the effectiveness of the CNN architecture.

Use Cases

- Digit recognition systems in applications like postal code recognition, check processing, and digital form submissions.
- Foundation for more complex image classification projects (e.g., multi-class object detection).

Conclusion

This project highlights the effectiveness of CNNs in solving image classification challenges. By training on the MNIST dataset, the model achieved exceptional accuracy, making it a reliable tool for digit recognition tasks.

Contact Information

- **Name:** Junaid Umar
- **GitHub Username:** umaralp
- **GitHub Repository:** https://github.com/Umaralp/ImageClassifier
- **Email:** junaid.umar.ju@gmail.com