Handwritten Digit Recognition and Camera Capture Documentation

Overview

This Python script combines two main functionalities:

1. Handwritten Digit Recognition:
    - Load and preprocess the MNIST dataset.
    - Build and train a Convolutional Neural Network (CNN) model for recognizing handwritten digits.
    - Visualize a few sample images from the MNIST dataset.
    - Evaluate the model's accuracy on the test data.
2. Camera Capture and Number Recognition:
    - Capture an image from the default camera (usually the built-in webcam).
    - Use Tesseract OCR to recognize digits in the captured image.
    - Extract and display recognized numbers.
    - Visualize the original image with recognized text.

Libraries Used

The script utilizes the following Python libraries:

- **numpy** (imported as **np**): For numerical operations.
- **keras**: For building and training the CNN model.
- **cv2** (OpenCV): For camera capture and image processing.
- **pytesseract**: For Optical Character Recognition (OCR) using Tesseract.
- **matplotlib.pyplot** (imported as **plt**): For data visualization.

MNIST Digit Recognition

Loading and Preprocessing Data

- The script loads the MNIST dataset, which contains hand-written digits.
- The data is preprocessed by normalizing pixel values to a range of [0, 1].

Building and Training the CNN Model

- A Convolutional Neural Network (CNN) model is constructed to recognize digits.
- It consists of convolutional layers, max-pooling layers, and fully connected layers.
- The model is compiled with categorical cross-entropy loss and the Adam optimizer.

- It is trained on the training data with a validation split for a specified number of epochs.

Data Visualization

- A function (**visualize_mnist_samples**) is provided to visualize a few sample images from the MNIST dataset.
- This helps users understand the dataset and model input.

Model Evaluation

- The trained model's performance is evaluated on the test data.
- The script displays the test loss and accuracy.

Camera Capture and Number Recognition

Capturing an Image

- The script captures an image from the default camera (usually the built-in webcam) using OpenCV (**cv2**).
- It checks for the availability of the camera and captures a single frame.

Recognizing Digits

- The captured image is processed and recognized using Tesseract OCR (**pytesseract**).
- The script extracts text (digits) from the recognized text.

Extracting Numbers

- Recognized text is processed to extract numbers using Python's string operations.

Displaying Results

- The script displays the recognized text and extracted numbers.
- It also shows the original captured image with the recognized text.

Usage

- Users are prompted to press Enter to capture an image from the camera.
- After capturing the image, the script performs digit recognition and displays results.

Dependencies and Setup

- Ensure that the required libraries (listed above) are installed in your Python environment.
- For Tesseract OCR on Windows, set the Tesseract executable path (only if needed).