

preprocessing code explanation.

1. Importing Libraries

- `os` : To handle file and directory operations.
 - `cv2` (**OpenCV**): For image processing tasks.
 - `numpy` : To handle numerical operations and arrays.
-

2. Feature Extraction (Unused in This Context)

- **FeatureExtractor class**:
 - Implements Local Binary Pattern (LBP) feature extraction, used for image texture analysis.
 - Currently, the class is defined but not used in the script. You can use this later for advanced processing.
-

3. Preprocessing Function

The core of the script lies in the `preprocess_images()` function.

- **Input and Output Directories:**
- `input_dir` : The folder containing original handwriting images, organized into subfolders for each author.
- `output_dir` : The folder where processed images are saved, mirroring the input directory structure.
- **Steps:**

1. Directory Creation:

- If the output directory or subfolders don't exist, they are created.

2. Image Loading:

- Reads `.jpg` images from the input directory in grayscale using OpenCV.

3. Image Preprocessing:

- Applies binary thresholding to the grayscale image to create a binary image.
- This converts pixel intensities into black and white, enhancing clarity for further analysis.

4. Save Processed Images:

- Saves the binary image in the corresponding subfolder of the output directory.
 - **Key Operations:**
 - `cv.threshold`: Performs binary thresholding.
 - `cv.imwrite`: Saves the processed image.
-

4. Function Call

The `preprocess_images()` function is called with:

- `input_directory`: "Handwriting"
 - `output_directory`: "Processed_Handwriting"
-

Goal of the Script

- **Input:** Handwriting images organized by authors in a folder.
- **Processing:** Convert grayscale images to binary (black & white) for better analysis and storage.
- **Output:** Save the processed binary images in a mirrored folder structure.

This setup is useful for tasks like handwriting recognition, where clean, binary images are easier to analyze with machine learning or feature extraction algorithms.