OCR PDF to Text Converter

Overview

This Python script extracts text from PDF or image files (JPG, PNG, JPEG) using Optical Character Recognition (OCR). It utilizes the Tesseract OCR engine and supports preprocessing of images to enhance text extraction accuracy.

Pipeline Explanation

1. **Input File Handling**:
   - Accepts input as either a `.pdf` or image file (`.png`, `.jpg`, `.jpeg`).
   - Converts PDF files to images using `pdf2image`.

2. **Image Preprocessing**:
   - Converts the input image to grayscale.
   - Applies adaptive thresholding for binarization.
   - Denoises the image using a median blur filter.
   - Displays the preprocessed image for debugging purposes.

3. **OCR Text Extraction**:
   - Uses Tesseract OCR to extract text from the preprocessed image.
   - Supports customization via Tesseract configuration options (`--oem` and `--psm`).

4. **Output Handling**:

   - Saves the extracted text into `.txt` files in the specified output directory.

   - Each page of a PDF is saved as a separate `.txt` file.

How to Run the Script

Prerequisites

- Python 3.x

- Installed libraries:

  - `opencv-python`

  - `pytesseract`

  - `pdf2image`

  - `Pillow`

  - `matplotlib`

- Tesseract OCR installed on your system:

  - [Download Tesseract OCR](https://github.com/tesseract-ocr/tesseract)

Installation

1. Clone or download the script to your local system.

2. Install required Python libraries:

   ```bash
   pip install opencv-python pytesseract pdf2image Pillow matplotlib
   ```

3. Install Tesseract OCR and set the executable path in the script:

   ```python
   pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
   ```

Running the Script

1. Open a terminal or command prompt.

2. Execute the script:

   ```bash
   python ocr_pdf_to_text.py
   ```

3. Enter the file path for the input PDF or image when prompted.

4. The extracted text will be saved in the `extracted_text` directory.

Sample Input and Output

Input:

- A PDF file with handwritten or printed text.

- An image file (e.g., `.png`, `.jpg`).

Output:

- Extracted text as `.txt` files saved in `extracted_text` directory.

Debugging and Logs

- Preprocessed images are displayed during execution for verification.

- Debugging information (e.g., file type and processed pages) is printed to the console.

Future Enhancements

- Support for custom-trained OCR models (e.g., CNN + LSTM).

- Add error rate evaluation metrics (CER, WER).

- Batch processing for multiple files.