OCR Application with Streamlit

Introduction:

The provided Python script implements an Optical Character Recognition (OCR) application using the EasyOCR library and Streamlit framework. The application allows users to upload an image file containing text, performs OCR to extract text from the image, preprocesses the extracted text, and displays both the extracted text and key features derived from it.

Code Overview:

1. Imports:
 The script imports necessary libraries such as os, numpy, easyocr, PIL, and streamlit.

2. Environment Variable: It sets the TESSDATA_PREFIX environment variable required for Tesseract OCR.

3. Image Conversion: Defines a function convert_to_jpg() to convert uploaded images to JPEG format using PIL.

4. OCR Function: Defines a function perform_ocr() to perform OCR on the uploaded image using EasyOCR.
 It extracts text from the image.

5. Text Preprocessing: Defines a function preprocess_text() for preprocessing the extracted text.
 Currently, no specific preprocessing steps are implemented.

6. Feature Extraction: Defines a function extract_features() to extract key features from the preprocessed text. The extracted features include document type, dates, parties involved, key terms, and action elements.

7. Image Processing: Defines a function process_image() to process the uploaded image. It performs OCR, preprocesses the extracted text, extracts features, and displays the results using Streamlit.

8. Main Function: The main() function is the entry point of the script. It creates a Streamlit application titled "OCR App ", allows users to upload an image file, and calls the process_image() function to perform OCR on the uploaded image.

Usage:

- Run the script.
- Access the Streamlit application in a web browser.
- Upload an image file containing text.
- The application will perform OCR on the uploaded image and display the extracted text along with key features.

How to run this code:

>pip install numpy easyocr pillow streamlit
>streamlit run app.py

Conclusion:

The provided Python script efficiently implements an OCR application using EasyOCR and Streamlit, allowing users to extract text from uploaded images and view key features derived from the extracted text. It provides a user-friendly interface for performing OCR tasks without the need for complex configurations. The application can be further enhanced by implementing additional preprocessing steps or improving feature extraction techniques based on specific requirements. Additionally, error handling mechanisms ensure smooth operation even in case of unexpected issues.