Sheet Stack Counter Application

Overall Approach

The Sheet Stack Counter application is designed to accurately count the number of sheets in an image of stacked papers using advanced image processing techniques. The approach involves several key steps:

1. **Image Upload**: Users upload an image of sheet stacks via a web interface.
2. **Preprocessing**: The image is converted to grayscale and sharpened to enhance details and make sheet detection more accurate.
3. **Foreground Segmentation**: Adaptive thresholding separates the sheets from the background, while background blurring enhances visual clarity.
4. **Sheet Detection**: Contour detection identifies individual sheets based on their shapes and sizes.
5. **Result Visualization**: The application displays the number of detected sheets and provides annotated images for user validation.

Frameworks/Libraries/Tools

- **OpenCV**: Used for image processing tasks including sharpening, blurring, thresholding, and contour detection.
- **NumPy**: Handles array operations and mathematical computations for image data.
- **Stream lit**: Creates a user-friendly web interface for image upload and result display.
- **Pillow**: Converts images between formats compatible with OpenCV.

Challenges and Solutions

1. **Challenge**: Ensuring accurate sheet detection while dealing with varying image quality and lighting conditions.

   o **Solution**: Implemented adaptive thresholding and sharpening techniques to improve contrast and detail, making sheet detection more robust.

2. **Challenge**: Balancing background blurring with maintaining the clarity of foreground sheets.

   o **Solution**: Applied Gaussian blur selectively to the background while preserving the sharpness of the foreground using bitwise operations.

3. **Challenge**: Filtering out noise and small contours to avoid counting irrelevant objects as sheets.

   o **Solution**: Introduced a contour area threshold to ignore small, irrelevant contours and focus on significant sheets.

Future Scope

1. **Enhanced Accuracy**: Implement machine learning models for more precise sheet detection and classification, especially for complex and varied sheet stacks.

2. **User Interface Improvements**: Add features such as drag-and-drop image upload, real-time processing feedback, and adjustable threshold settings for better user control.

3. **Batch Processing**: Extend the application to handle multiple images or video feeds for batch processing and automated sheet counting in larger volumes.

4. **Integration with Cloud Services**: Allow users to store and manage images and results in the cloud, enabling easier access and sharing of data.