Sheet Stack Counter App Documentation

Overall Approach

The goal of this project was to develop an application that automates the counting of sheet stacks in a manufacturing plant using image processing techniques. The approach involved several key steps:

1. Image Preprocessing: Convert the uploaded image to grayscale and apply Gaussian blur to reduce noise. Use Canny edge detection to identify edges in the image.
2. Line Detection: Utilize the Probabilistic Hough Line Transform to detect lines in the edge-detected image.
3. Line Merging: Merge lines that are close to each other to reduce redundancy and improve accuracy.
4. Midpoint Calculation: Calculate the midpoints of the merged lines to identify the positions of the sheet stacks.
5. Sheet Counting: Count the number of midpoints to determine the number of sheets in the stack.

Frameworks/Libraries/Tools

- Streamlit: Used for building the interactive web application.
- OpenCV: Utilized for image processing tasks such as converting to grayscale, applying Gaussian blur, and edge detection.
- NumPy: Employed for numerical operations and array manipulations.
- Pillow: Used for image enhancement tasks like rotation, resizing, brightness, and contrast adjustments.

- streamlit-cropper: Provided cropping functionality within the Streamlit application.

Challenges and Solutions

1. Image Noise: Noise in the uploaded images affected edge detection accuracy.

- Solution: Applied Gaussian blur to smooth the image and reduce noise before performing edge detection.

2. Line Detection Accuracy: The initial line detection results included many short and irrelevant lines.

- Solution: Used a combination of minimum line length and maximum line gap parameters in the Hough Line Transform to filter out short and irrelevant lines. Additionally, merged lines that were close to each other to improve accuracy.

3. Dynamic Image Adjustments: Users needed flexibility to adjust the image to improve detection accuracy.

- Solution: Provided image editing options such as crop, rotate, resize, brightness, and contrast adjustments in the sidebar to allow users to fine-tune the image before processing.

Future Scope

1. Real-Time Processing: Implement real-time video processing to count sheets in a continuous manufacturing process.

2. Improved Line Merging Algorithm: Enhance the line merging algorithm to handle more complex cases and improve detection accuracy.

3. User Feedback Loop: Incorporate a feedback loop where users can manually adjust detected lines and midpoints to improve the accuracy of sheet counting.

4. Additional Image Formats: Support additional image formats and improve the robustness of image uploading and processing.

5. Mobile Compatibility: Develop a mobile-friendly version of the application to allow users to capture and process images directly from their mobile devices.

6. Advanced Image Editing: Add more advanced image editing options, such as color adjustments, sharpening, and noise reduction filters.