

Counting Stack of Sheets Application

Overall Approach

The goal of this project is to develop a web application that counts the number of sheets in a stack from an uploaded image. The approach involves using computer vision techniques to detect the edges and lines in the image, which correspond to the edges of the sheets. The application is built using Streamlit for the web interface and OpenCV for image processing.

The steps are as follows:

1. **Image Upload:** Users can upload an image of a sheet stack.
2. **Image Processing:**
 - Convert the image to grayscale.
 - Apply Gaussian blur to reduce noise.
 - Detect edges using the Canny edge detection algorithm.
 - Detect lines using the Hough Line Transform.
3. **Line Counting:** Count the detected lines, which correspond to the sheet edges.
4. **Display Results:** Show the original image, the image with detected lines, and the edge-detected image, along with the count of sheets.

Frameworks/Libraries/Tools

1. **Streamlit:** Used to create the web interface for uploading images and displaying results.
2. **OpenCV:** Utilized for image processing, including grayscale conversion, Gaussian blur, edge detection, and line detection.
3. **Pillow:** Used to handle image uploads and convert images to a format suitable for OpenCV.
4. **Numpy:** Provides support for handling arrays and image data.
5. **Tensorflow:** Open-source software library for machine learning and artificial intelligence.
6. **Matplotlib:** Used for visualizing images during development.

Challenges and Solutions

Challenge 1: Noise in Images

- **Problem:** Noise in images can lead to false edge detections.
- **Solution:** Applied Gaussian blur to the grayscale image to reduce noise before edge detection.

Challenge 2: Accurate Edge Detection

- **Problem:** The Canny edge detector can sometimes miss edges or detect too many edges.
- **Solution:** Tuned the parameters of the Canny edge detector to balance sensitivity and specificity for edge detection.

Challenge 3: Line Detection and Counting

- **Problem:** The Hough Line Transform can detect multiple lines for a single sheet edge, leading to overcounting.
- **Solution:** Set appropriate thresholds for the Hough Line Transform and applied post-processing to merge closely spaced lines.

Future Scope

Improved Edge Detection

- **Propose:** Experiment with other edge detection algorithms or enhance pre-processing steps to improve accuracy.

Robust Line Merging

- **Propose:** Implement more sophisticated algorithms to merge detected lines that represent the same sheet edge.

User Interface Enhancements

- **Propose:** Add more interactive features in the Streamlit interface, such as adjustable parameters for edge detection and line detection.

Support for Various Image Types

- **Propose:** Expand the application to handle different types of images, including various lighting conditions and perspectives.

Machine Learning Integration

- **Propose:** Incorporate machine learning models to automatically learn and improve edge and line detection based on a dataset of labeled images.