

# **Sheet Count from Image - Streamlit App**

- **Veera Karthick V**

## **1. Overall Approach**

The goal of this project is to develop a Streamlit application that processes images of stacked sheets and counts the number of sheets using edge detection techniques. The approach involves:

1. **Image Upload and Processing:** Allow users to upload a JPEG image. The uploaded image is processed to detect edges using Canny edge detection.
2. **Edge Detection:** Convert the image to grayscale, apply Gaussian Blur, and then use Canny edge detection to highlight the edges of sheets.
3. **Contour Detection:** Identify contours in the edge-detected image to count the number of sheets based on detected contours.
4. **Display Results:** Show the original and processed images side by side, display the total count of sheets, and indicate the time taken to process the image.
5. **User Interface:** Provide a user-friendly interface with clear instructions and an expandable section to display default edge detection thresholds.

## **2. Frameworks/Libraries/Tools**

- **Streamlit:** A framework for creating interactive web applications with Python. It is used to build the user interface of the application.
- **OpenCV:** A library for computer vision tasks. It is used for image processing tasks such as grayscale conversion, Gaussian Blur, Canny edge detection, and contour finding.
- **NumPy:** A library for numerical operations. It is used for handling image data as arrays and performing numerical operations on the data.
- **Matplotlib :** A library for plotting images. It is used to display images in the app, though Streamlit's built-in image display functions are used in the final app.

### 3. Challenges and Solutions

#### Challenges:

##### 1. Edge Detection Accuracy:

- **Challenge:** Ensuring accurate edge detection for varying image conditions and sheet configurations.
- **Solution:** Used Gaussian Blur to reduce noise and adjusted Canny edge detection thresholds to balance sensitivity and specificity.

##### 2. Contour Detection:

- **Challenge:** Differentiating between individual sheets and ensuring that contours are accurately counted.
- **Solution:** Applied morphological operations (dilation and erosion) to enhance contours and reduce noise, improving the accuracy of contour detection.

##### 3. User Interface Design:

- **Challenge:** Creating a user-friendly interface that is intuitive and visually appealing.
- **Solution:** Used Streamlit's layout features to organize images and results, and added expandable sections for additional information, ensuring a clean and interactive UI.

### 4. Future Scope

##### 1. Improved Detection Algorithms:

- **Enhancement:** Explore advanced image processing and machine learning techniques to improve detection accuracy for different sheet configurations and lighting conditions.

##### 2. Real-Time Processing:

- **Enhancement:** Implement real-time image processing for live feed scenarios, allowing users to count sheets from a camera feed.

##### 3. Performance Optimization:

- **Enhancement:** Optimize processing time and memory usage for handling larger images or batch processing.

##### 4. Additional Features:

- **Enhancement:** Add features such as integrating with databases for record-keeping, and providing detailed analysis reports.