Summarizing:

## Overall Approach

I firstly preprocessed the image by setting the accurate brightness and contrast of the image in which we can clearly see the edges of the sheets.
Then I converted it to **grayscale** and applied **Gaussian blur** to reduce noise with **Canny edge detection** to find edges. It then passed to **adaptive thresholding technique** to carefully detect the edges.
Further I calculated the **edge profile** which is a 1D array where each element represents the sum of edge pixels in the corresponding row of the original image, and there I used **SciPy's find_peaks function** to detect peaks in the edge profile. It can handle variations in sheet thickness and spacing by adjusting the distance and prominence parameters.
By **counting the number of peaks**, we get an estimate of the number of sheets in the stack. The **accuracy** of this method can be fine-tuned by adjusting the edge detection parameters, as well as the **min_distance** and **prominence values** in the peak finding step.
We add a **visualization function** to show the original image, edge detection result, and the edge profile with detected peaks.
For an interface this time (In last assignment I used Flask), I used **streamlit library** to create a good user experience and deployed it on **hugging face spaces**(https://huggingface.co/spaces/hshivhare/Sheet-Counter-Application).

## Frameworks/Libraries/Tools

I used basic libraries such as:

- Open CV
- Numpy
- Scipy
- Matplotlib
- Streamlit

## Challenges and Solutions:

I basically faced issue in maintaining the accuracy of the counts of sheet stack like it should not detect any another line in an image that is not related to sheet for it I used adaptive thresholding technique to ignore the background with good processing of image like adjusting the brightness and contrast of the image.

## Future Scope

- Firstly we can be more accurate by training it like a CNN model if we know the correct count of sheets.
- Can click more good quality of images and avoid any obstacle in the image.
- Allow users to download results as CSV, PDF, or Excel files
- Add support for live camera feed to count sheets in real-time
- Predict future sheet needs using machine learning
- Provide automatic stock updates based on sheet counts
- Allow users to define custom sheet types or categories