Project Report on the Computer Hub AI Intern Assignment: Sheet Counter

Soumyadeep Bose

Introduction:

Sheet Counter is a project designed to count the number of sheets in a given video or image. The app uses computer vision techniques to detect and count horizontal lines, which represent sheets in the input media. There are two versions of the app available: one built using Streamlit and the other using Tkinter. The Streamlit app is more user-friendly and provides a better user experience, while the Tkinter app is more lightweight and can be used for quick testing. The Tkinter app was developed first, and then the Streamlit app.

Objective:

Developing a user-friendly interface to receive a video/image and count the number of sheets in it.

Overall Approach:

The code first reads the input video frame by frame and increases contrast to make the horizontal lines more visible. The frame is then passed to the Canny Edge Detector to detect the edges, and then Probabilistic Hough Line Transform is used to get the lines representing the edges. The code then filters out the non-horizontal lines and then using the y coordinates of mid-points of the lines, DBSCAN performs clustering to group lines on similar y coordinates as distinct sheets. The number of clusters is then counted to get the number of sheets in the frame.

Libraries Used:

1. **Streamlit**: For building the web-based interactive interface.
2. **OpenCV**: To process the video/image file.
3. **scikit-learn**: To perform clustering on edges of common sheets.

Approach Details:

1. Pre-processing the frames: When a video is uploaded, all frames are processed in an iteration. Each frame is converted to BGR and then resized to 600, 400 dimensions, and then passed to the `get_sheet_count` function. Within the function, the frame's contrast is increased by a factor of 1.25 so as to aid in the successive steps of edge detection.

2. Canny Edge Detector: Popular edge detection algorithm Canny Edge Detection is used to perform edge detection on the frame to detect the edges in it both horizontal and non-horizontal, i.e. angled slightly with respect to the horizontal. The upper and lower thresholds are 100 and 25 respectively, with an aperture size of 3. This step is crucial as it gives the base number of sheets there are potentially in the frame/image. However, it is very noisy.

3. Probabilistic Hough Transform: The edges detected by Canny Edge Detector were of various length and orientation, on top of being noisy. To get proper lines out of the detected edges, this step is performed.

4. Filtering out non-horizontal lines: In quite a lot of frames/images, after performing PHT it was observed that somehow there were many highly angled lines, overlapping the

horizontal lines that represented the edges of sheets. To ignore all such lines that are not edges, they were filtered out. Now, not all angled lines can be filtered out as some actual edges are slightly angled. So, there is a tiny threshold of 1 to account for the slight orientation resulting from the bending of the sheets.

6. Clustering using DBSCAN: For each frame or for an image, after obtaining all the potential lines that represent edges of the sheets, their mid-points are calculated, and since the stack of sheets is assumed to be horizontal, the lines are clustered on the basis of "similarity" of the y coordinate of their mid-points. This has been done because it was observed that for one edge multiple discontinuous lines were being counted. This step reduces that by effectively clustering the lines that represent the same edge. The parameter `eps` was tuned to the current value since it increased the accuracy greatly.

7. Displaying the number: The number of clusters in each frame/image is approximately the number of sheets in it. However, since a video has more than one frame, all frames are processed, number of clusters are calculated and stored, and then the maximum number is displayed to be the number of sheets in the video. For an image however, one simple call to the `get_sheet_count` function is sufficient.

Conclusion and Future Scope:

Sheet Counter is an application that counts the number of sheets in a video/image. However, there are many shortcomings. It has been observed that the reported number of sheets in the demo video is somewhat lesser than the actual ground truth. This is because many edges are either extremely near to each other or the lighting condition and/or contrasts are not suitable. Besides basic contrast adjustment, adaptive contrast enhancement can help too. Furthermore, implementing methods to automatically tune parameters (e.g., Canny thresholds, Probabilistic Hough Transform parameters, DBSCAN eps, etc.) based on the frame's characteristics can help in significantly increasing the accuracy of the system. Perhaps even dilating and distorting the frame/image in ways other than what was experimented during development, can help. Lastly, opting for more advanced clustering algorithms can also increase accuracy.

Thanks!