I want to use 1 free late day.

Q4_1)

In this part, we're supposed to write a function for Matching picturing using the extracted features using the FAST detector, building describers using the ComputeBRIEF and Matching points after that. I implemented this function and the output of the cv_desk and cv_cover is:

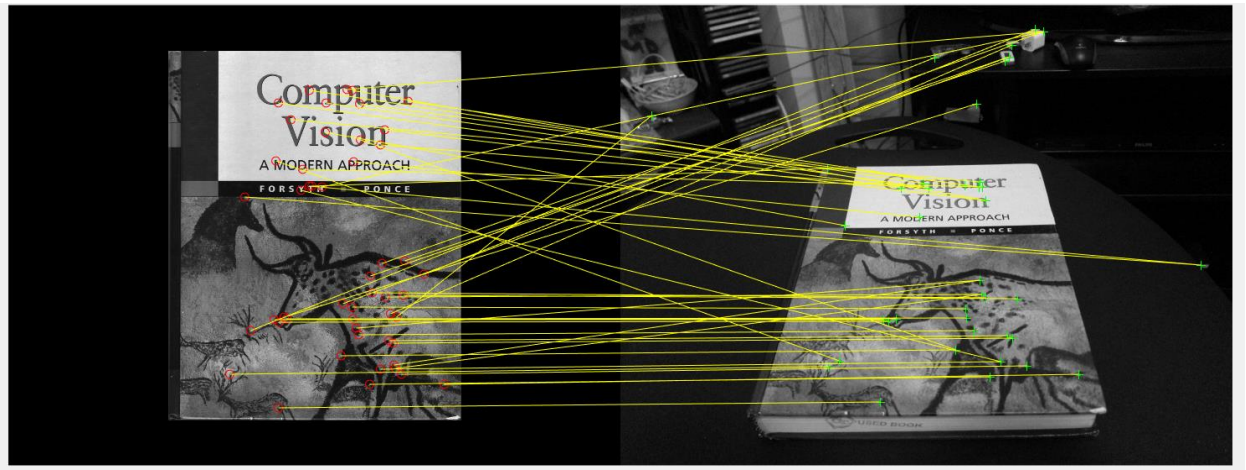


Figure 1: extracted pair of points in two images shows that some options in the function needs to be changed.

Q4_2)

In this part, pictures need to be rotate and then their features be detected using BRIEF and SURF descriptor. After that, describers will be built and the image will be matched with its rotated version. Then the number of matches in every orientation will be counted and the result for two method BRIEF and SURF is:

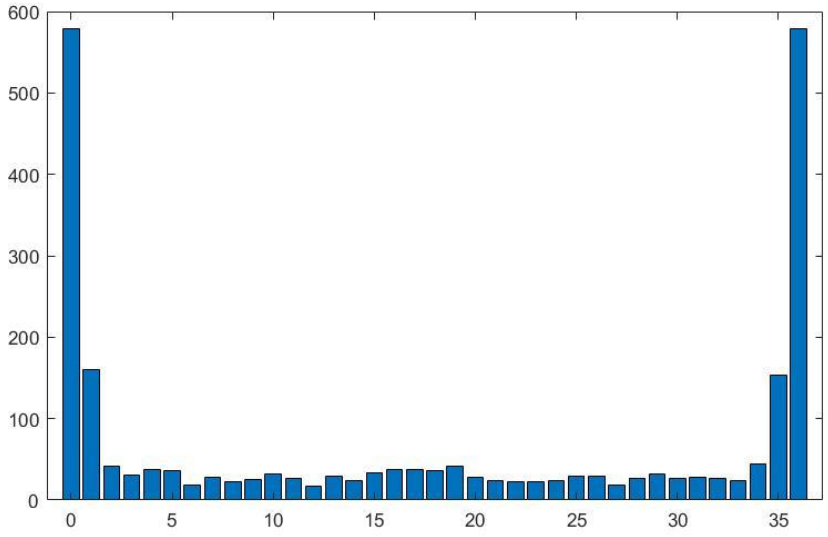


Figure 2: Histogram of number of matches using BRIEF.

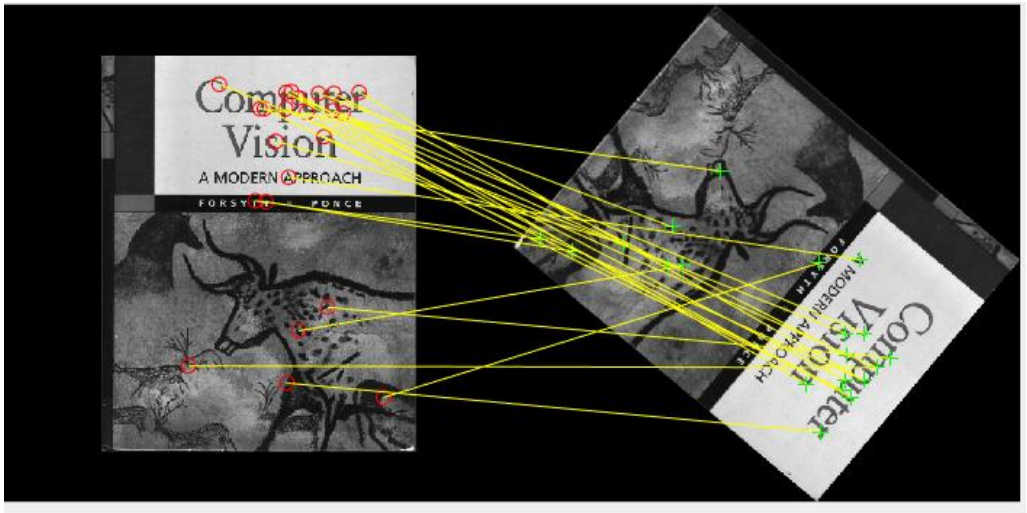


Figure 3: Extracted pair of points for the image and its rotated version using BRIEF.

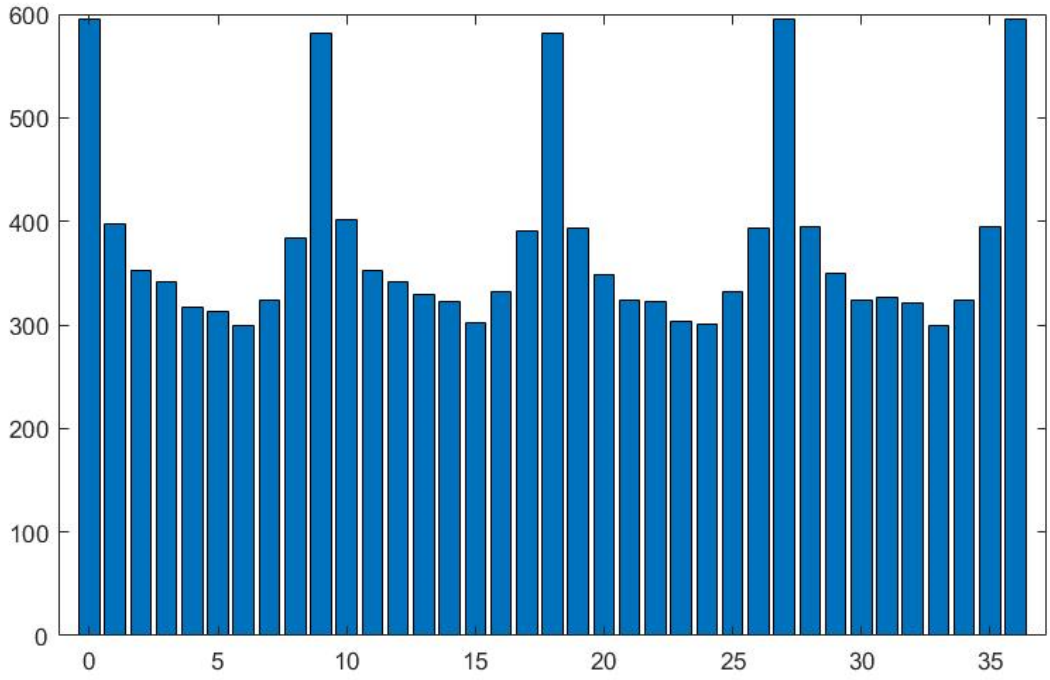


Figure 4: Histogram of number of matches using SURF.

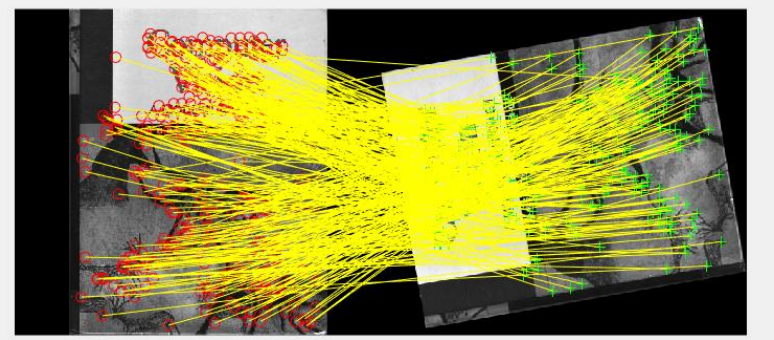


Figure 5: Extracted pair of points in image and its rotated version using SURF.

As it can be seen from the images, SURF is able to extract more accurate and better features and match point based on that.

Q4_3)

For getting output from function of this part and function of part 4, the RUNQ4_3_4 script has been written. For running Q4_4, the related lines, which has been commented, needed to be uncomment and run. In this part, only 10 points has used for computing H matrix. In addition, in these two parts, maybe there are some false points, which in RANSAC will be corrected. Also, it should be mentioned that for testing sections 3,4,5 the SURF features and descriptor has been used since it has been shown that leads to better results and extracted features.

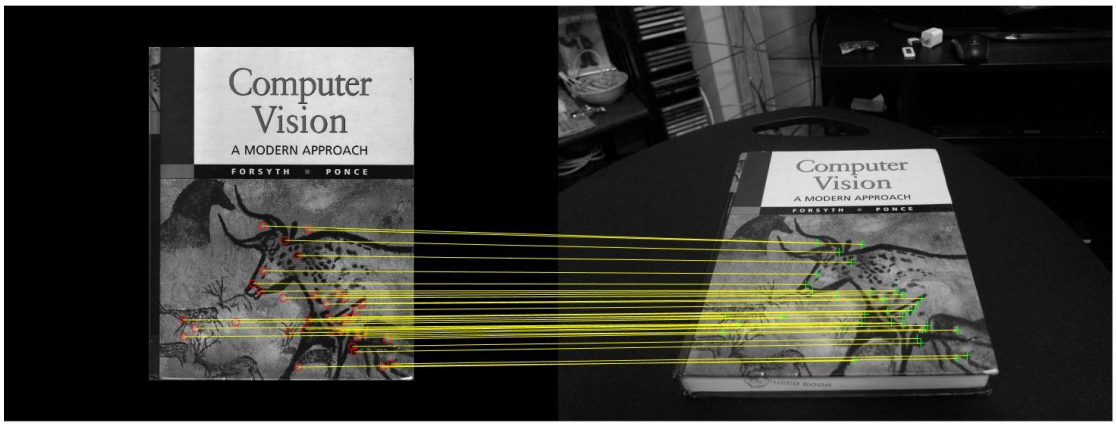


Figure 6: points and corresponding locations of them in the other image calculated by H.

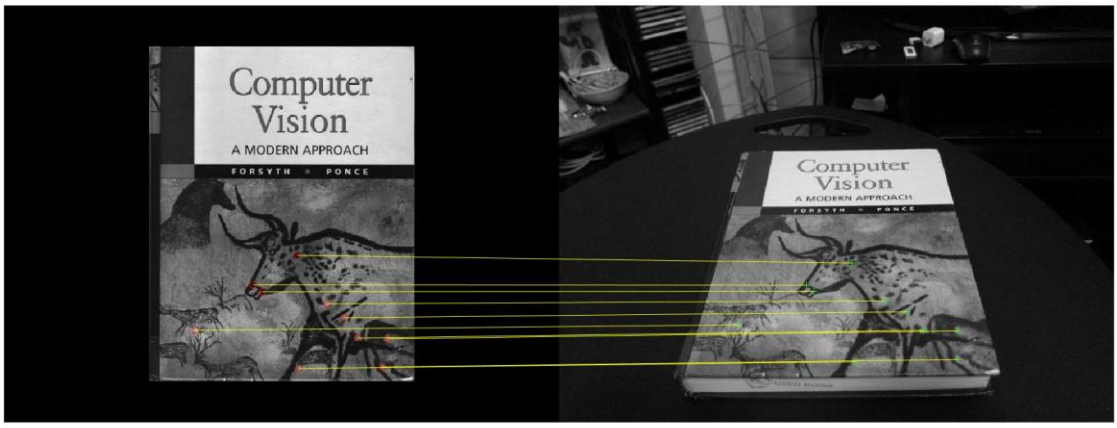


Figure 7: 10 pairs of points in both images.

Q4_4)

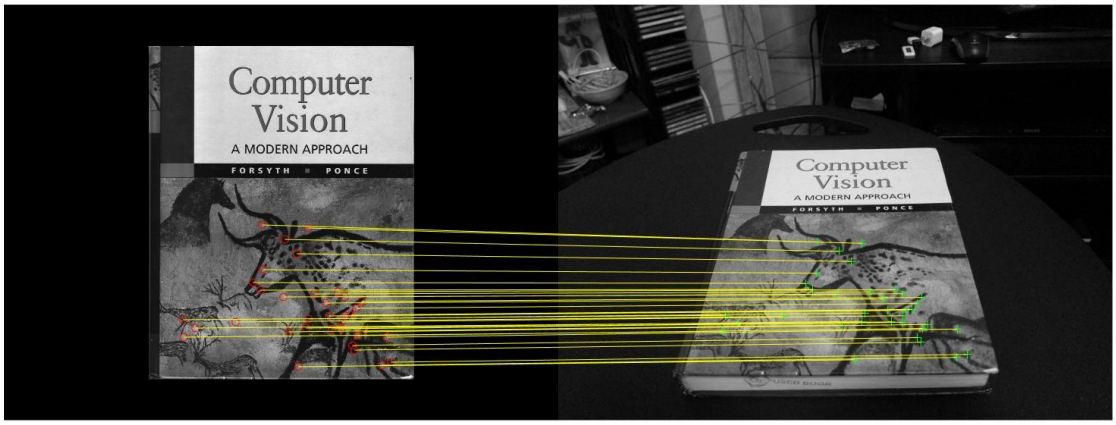


Figure 8: : points and corresponding locations of them in the other image calculated by H_norm.

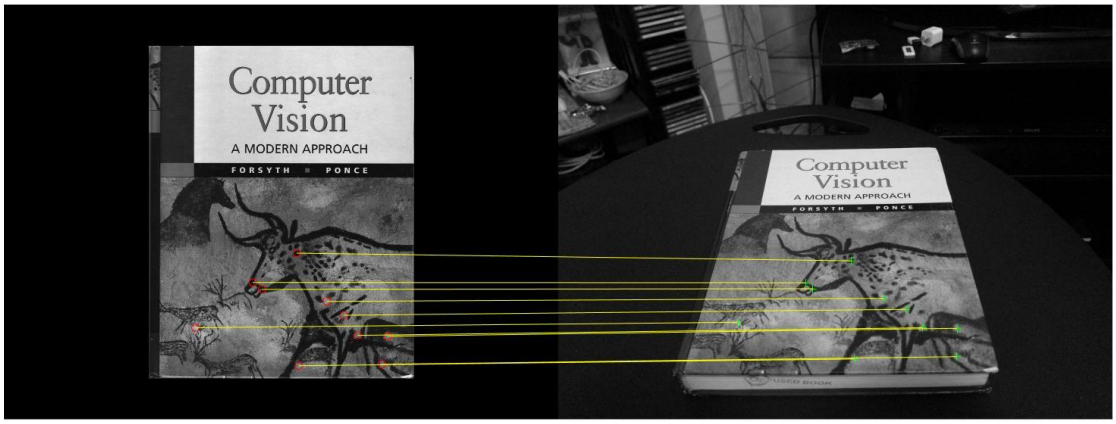


Figure 9: 10 pairs of points in both images.

Q4_5)

In this part, RANSAC algorithm has been implemented and tested by Q_4_5 script for two images.

For the Image 1:

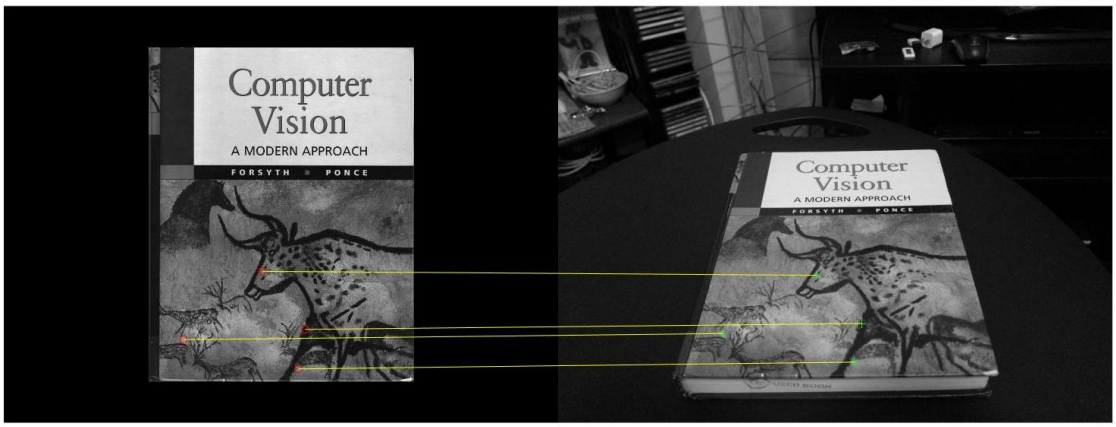


Figure 10: 4 pair points which make more number of inliers.

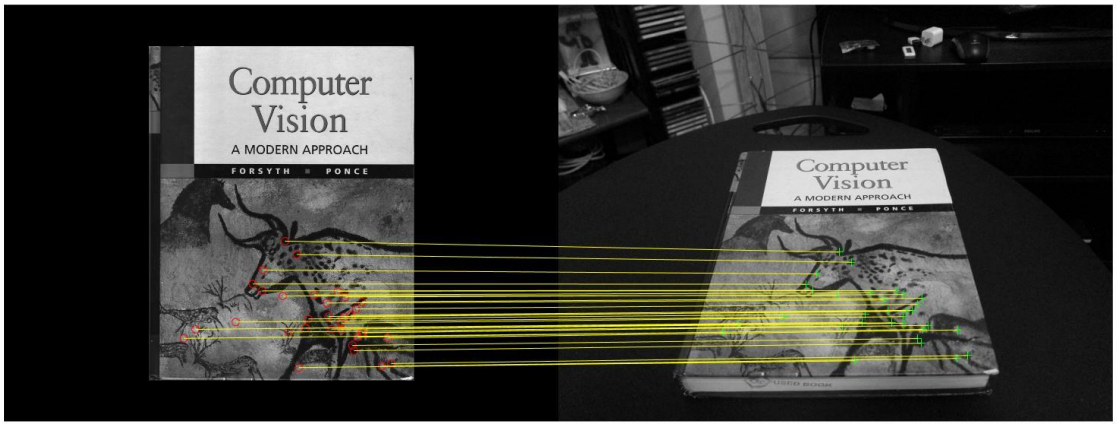
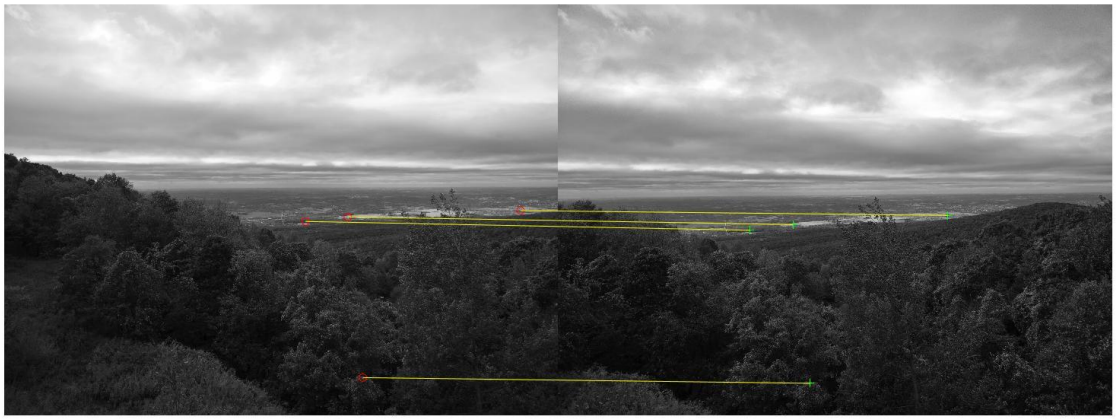
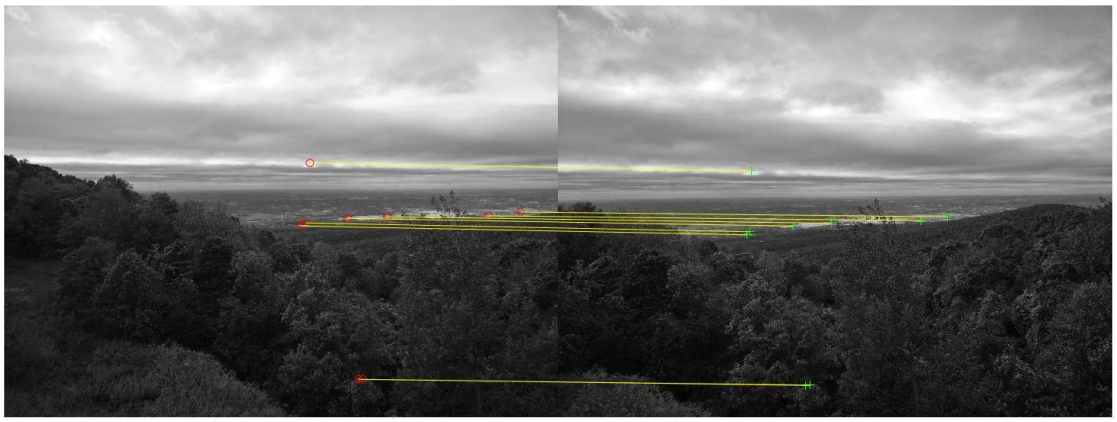


Figure 11: Selected Inliers by RANSAC.

Image 2:

4 pair_points that produces the most number of Inliers.





As we expected, since in the second image only left and right has been changed, the transport Matrix is sth like [0 0 0…0 1;0……0 1 0;…] and linier only transport everything from the right-side to left-side.

Q4_6)

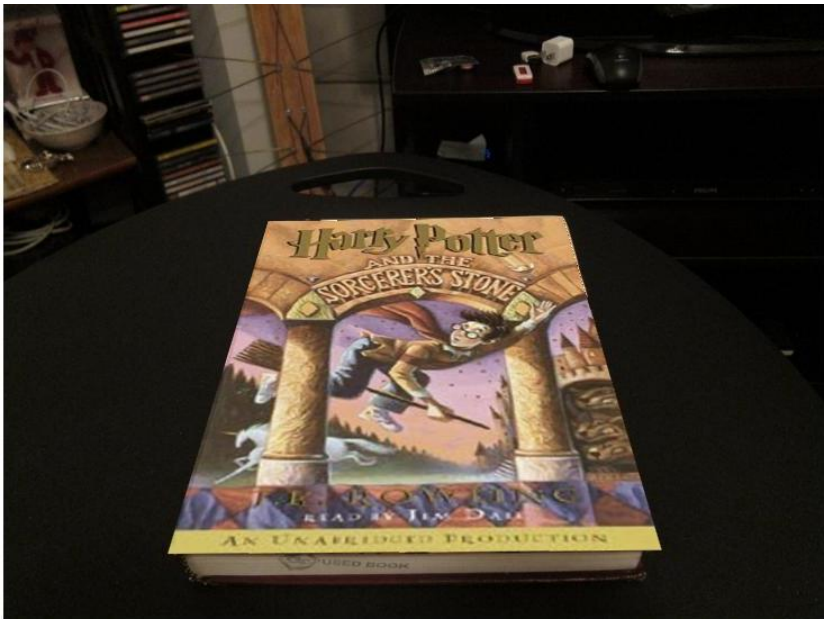


Figure 12: HarryPtterized CV book image.

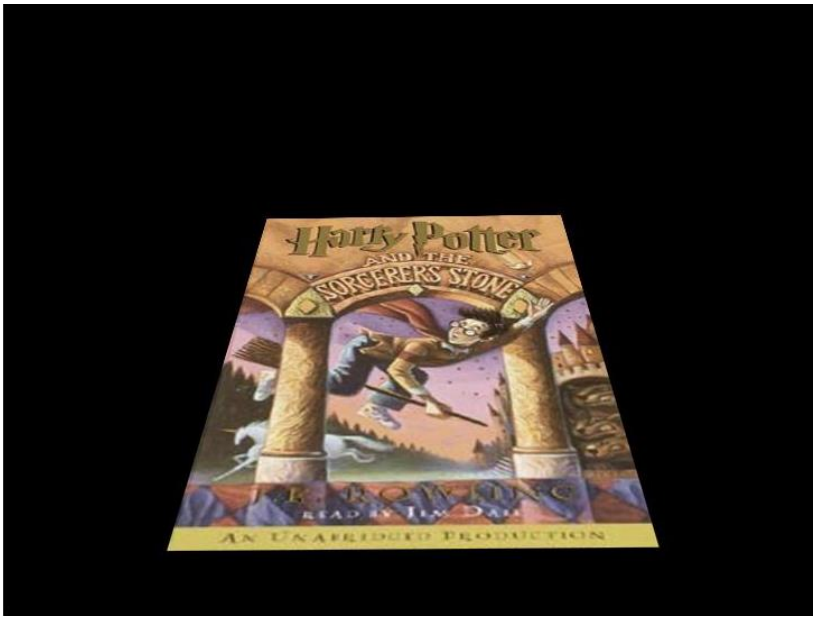


Figure 13: HarryPotterize before adding the surrounding space.

Q5) Result has saved as a result_ar.mov in the result folder. To keep result consistent, in every frame, the frame of the video has been cropped to have the same aspect ratio of the “computer vision” book image.