International Institute of Information Technology, Bangalore

AI 825: Visual Recognition

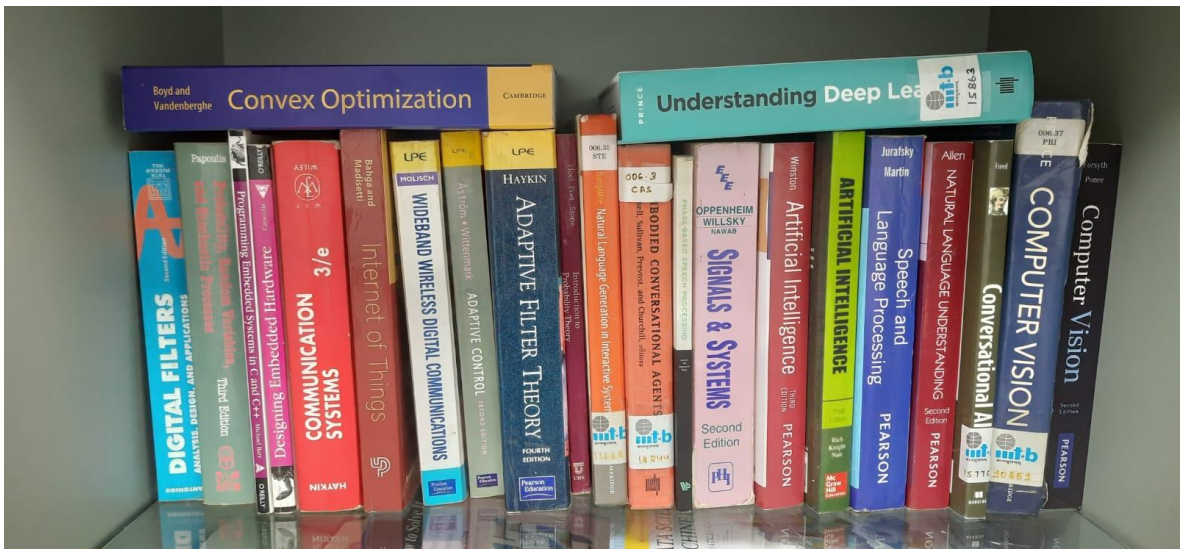
Report

On

Assignment 1

Task 1: Algo to count the number of books in the shelf

Given Image:



Initial Steps performed is given below:

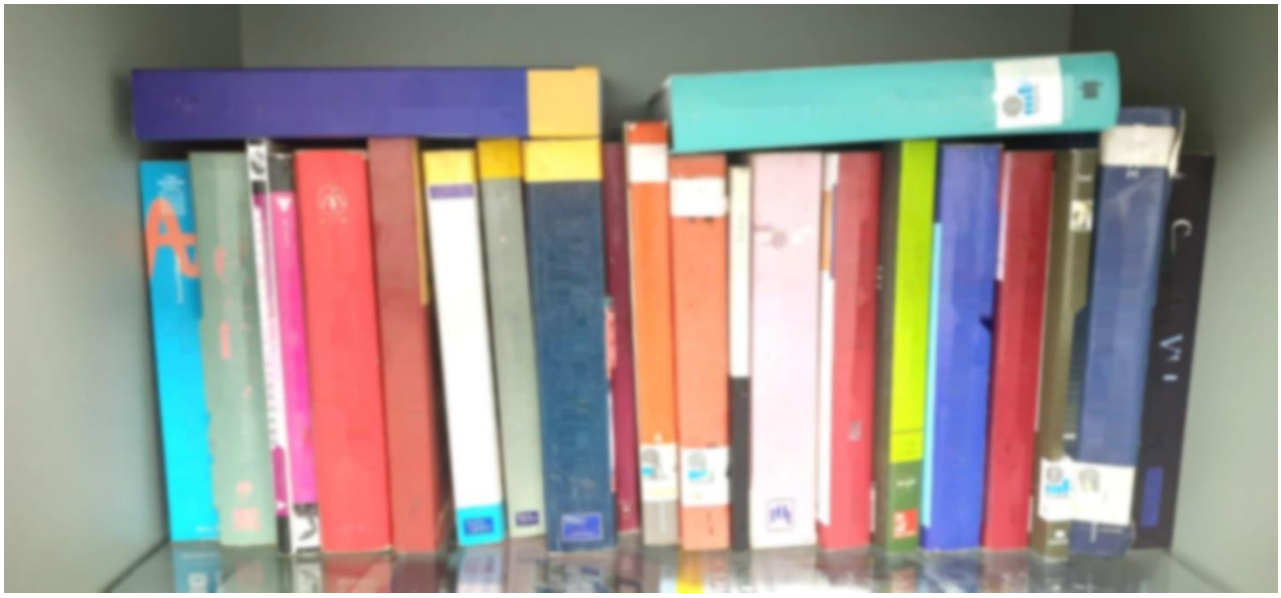
- Initially, I experimented with the Hough Lines Transform method in an attempt to detect the straight lines outlining the edges of the book. Despite exploring numerous parameter combinations, the obtained results were not satisfactory.
- Then, I turned to the Canny edge detection method, exploring various techniques within that approach to refine the edge detection process.

Final Steps performed is given below:

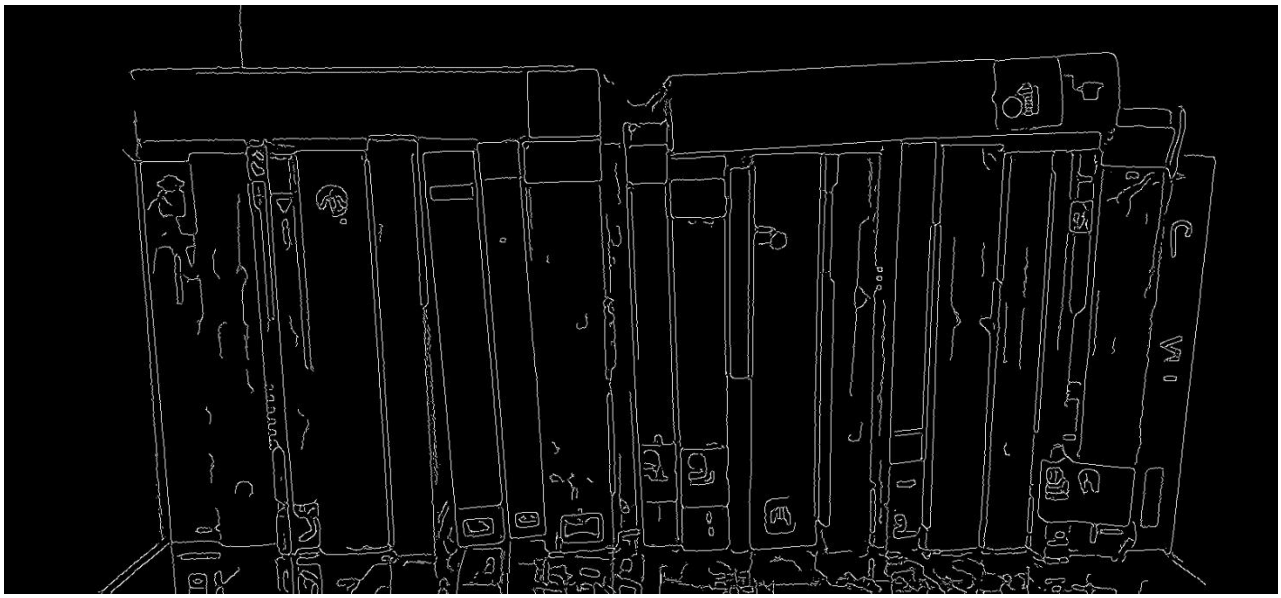
1. As while directly applying Canny Edge Detection method to the image, it is creating lots of unwanted edges due to the text in the imaged.
    a. By this observation it gave me a conclusion that at first, the text from the image had to be removed.
    b. To remove text from the image I used keras-ocr detection method to detect the text in the image and inpainted that area.

2. Adjusted the brightness and the contrast of the image for better edge detection.

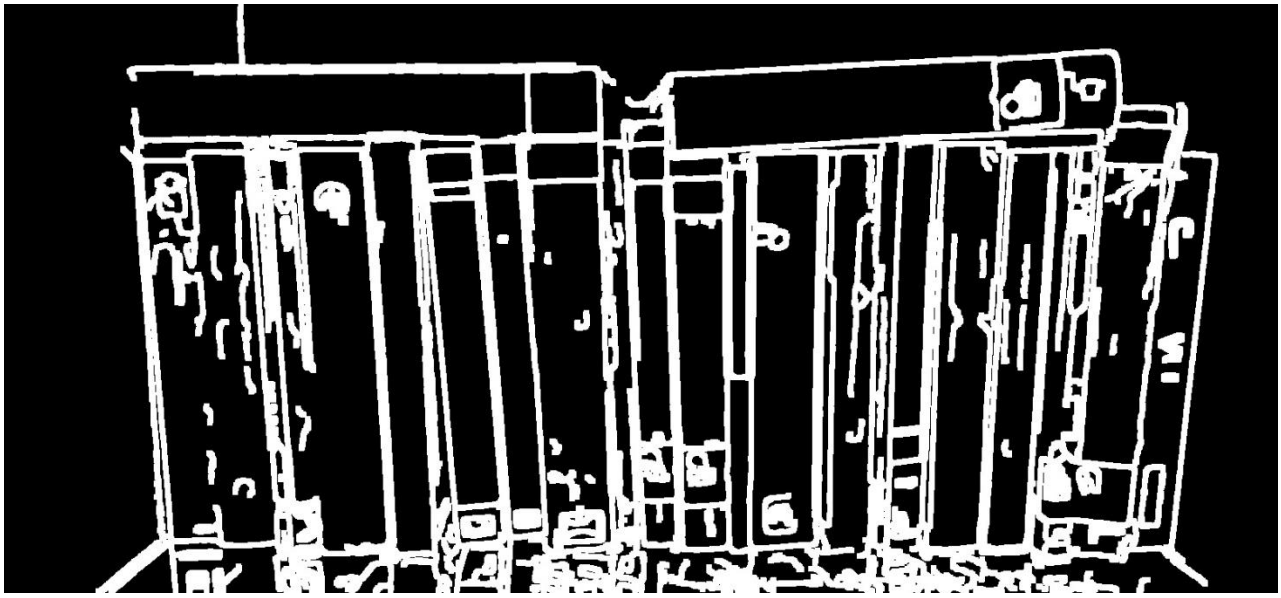
3. Then, applied Gaussian Blur in the above image using 15 X 15 kernel so that minor intensity changes get nullified.



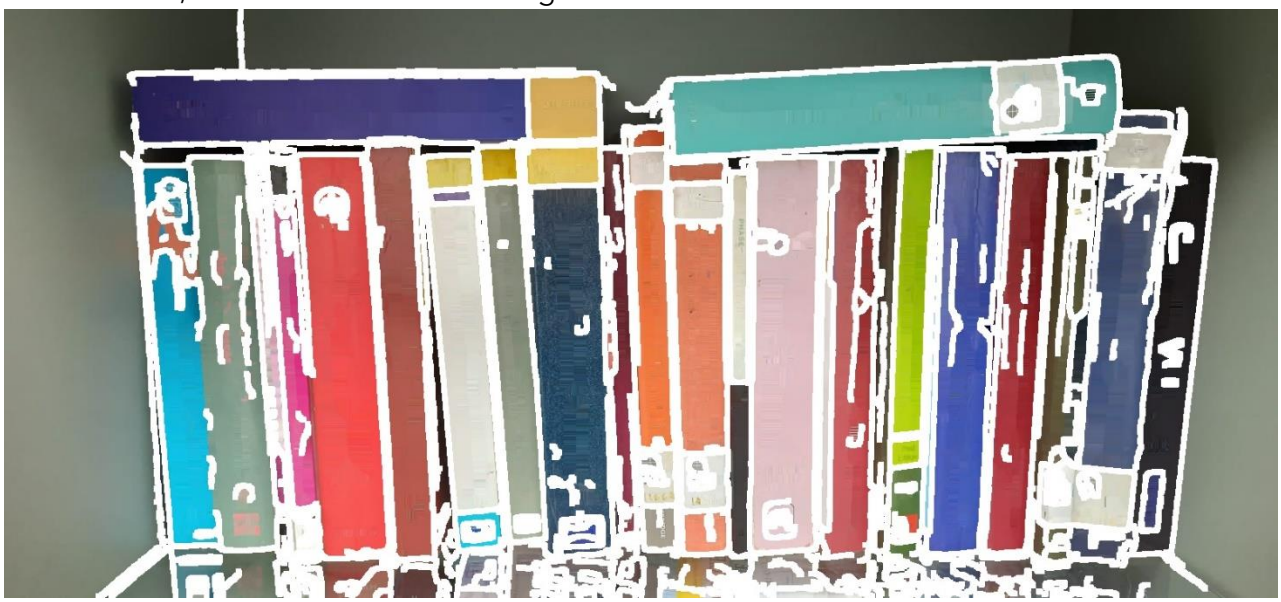
4. Converted the blurred image into grayscale and then applied Canny Edge Detection Method with 30,5 thresholds. After trying with multiple thresholds, finally found this pair of thresholds gave fairly good result.



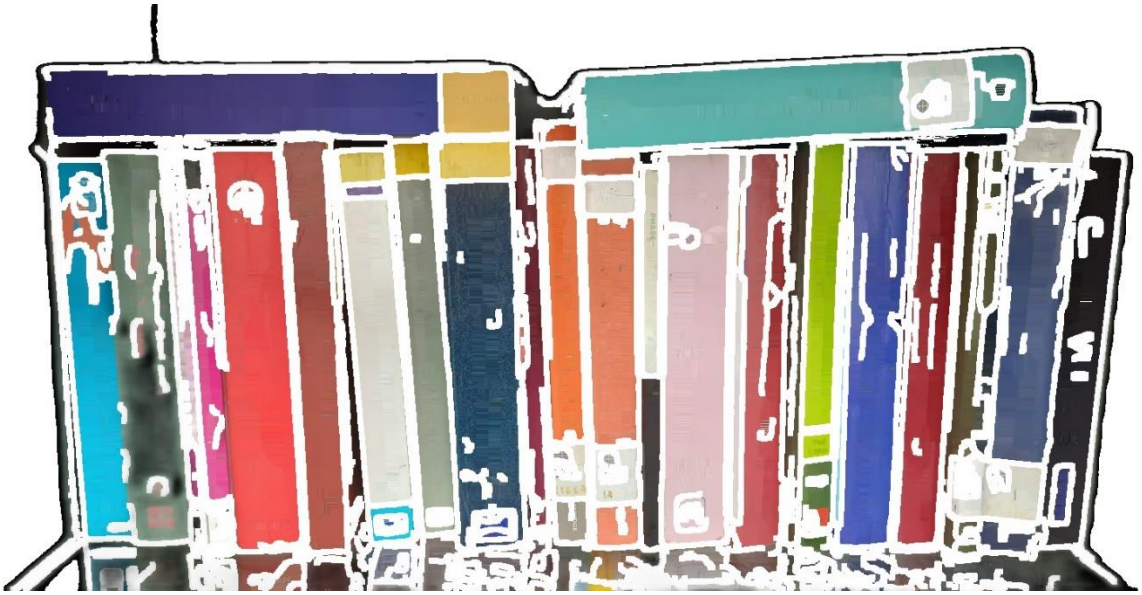
5. When I was trying to find contours directly on the original image, it is seen that due to similar colours of books it was unable to find the contours properly. Hence number of bounding boxes formed were very poor.
    a. So, when canny detected edges of books fairly enough I got an idea that lets superimpose the detected edges on the above edge.
    b. Then dilated the canny edges with 7 X 7 kernel to make the edges prominent.



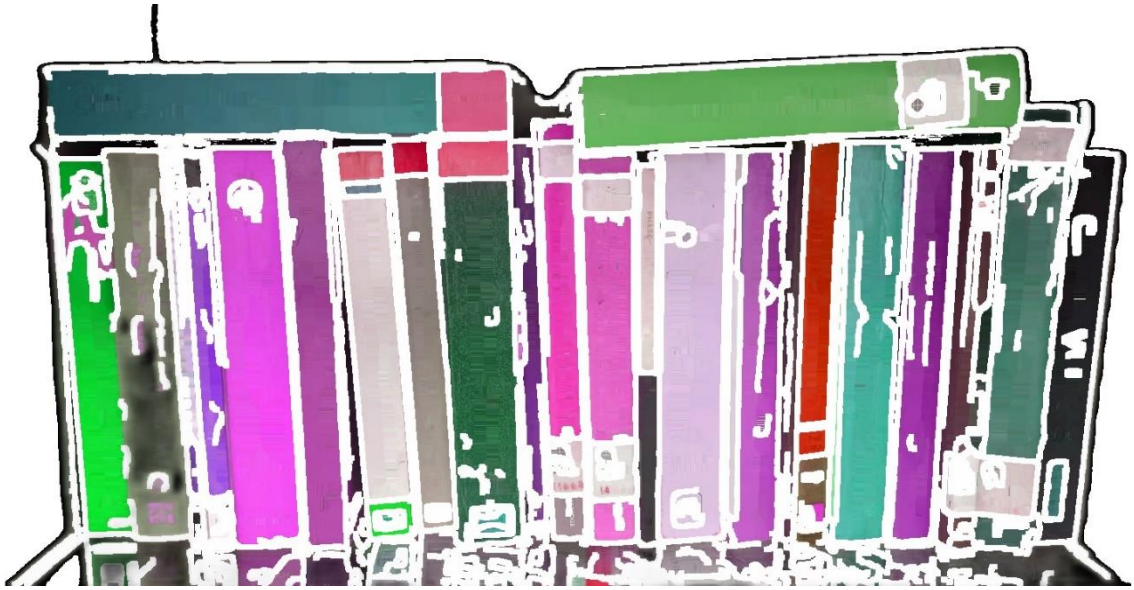
6. Then superimpose the original image with the dilated image to get a better idea about the edges of each book, so that contour detection gets better



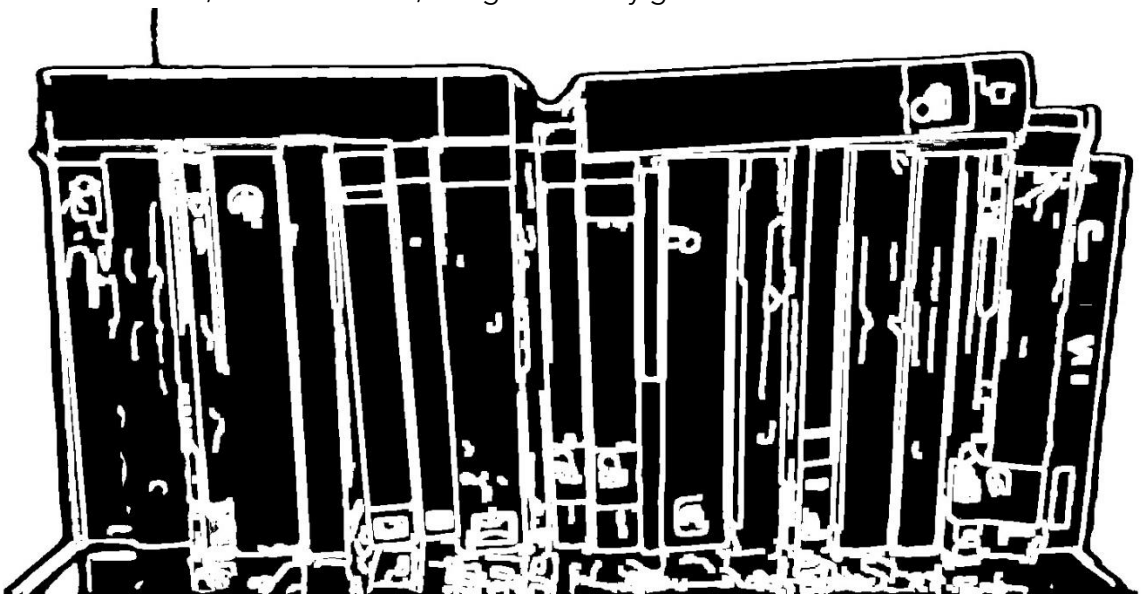
7. Then remove the background image, so that contour detection method focuses only on the foreground object



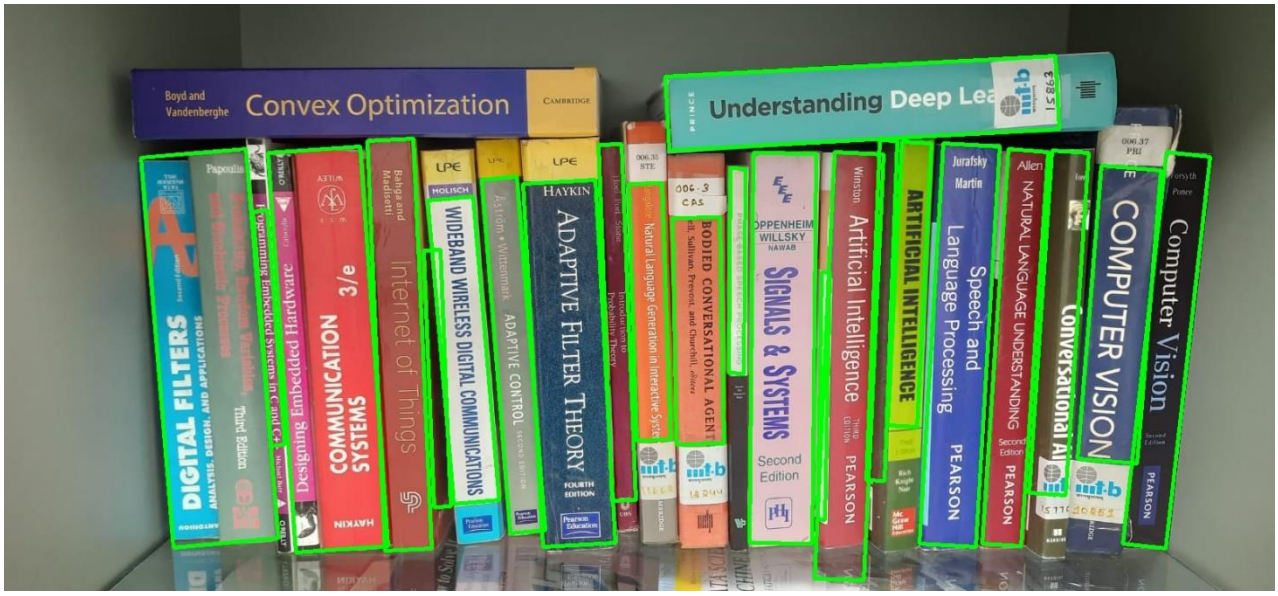
8. The contour of the blue book at the left top corner was not detected.
    a. So, I tried to shifted the hue of the image by 30 degrees so that each book has a contrasting and distinguishing colour.
    b. But unfortunately, after shifting the hue also its contour was not detected (the result we will see later in this report).



9. This is the last step before calculating the number of bounding boxes.
    a. converted the above image into grayscale.
    b. Applied thresholding on the grey image to create Binary image, after trying with multiple the thresholds, found that 240,255 gave a fairly good result.



10. Then I performed hierarchy-based contour detection and created the bounding boxes.
    Note: Filtered the bounding boxes based on the perimeters of each box.
    a. There are multiple unwanted small and very large bounding boxes were formed. That gave me idea to filter bounding boxes based on its perimeter.

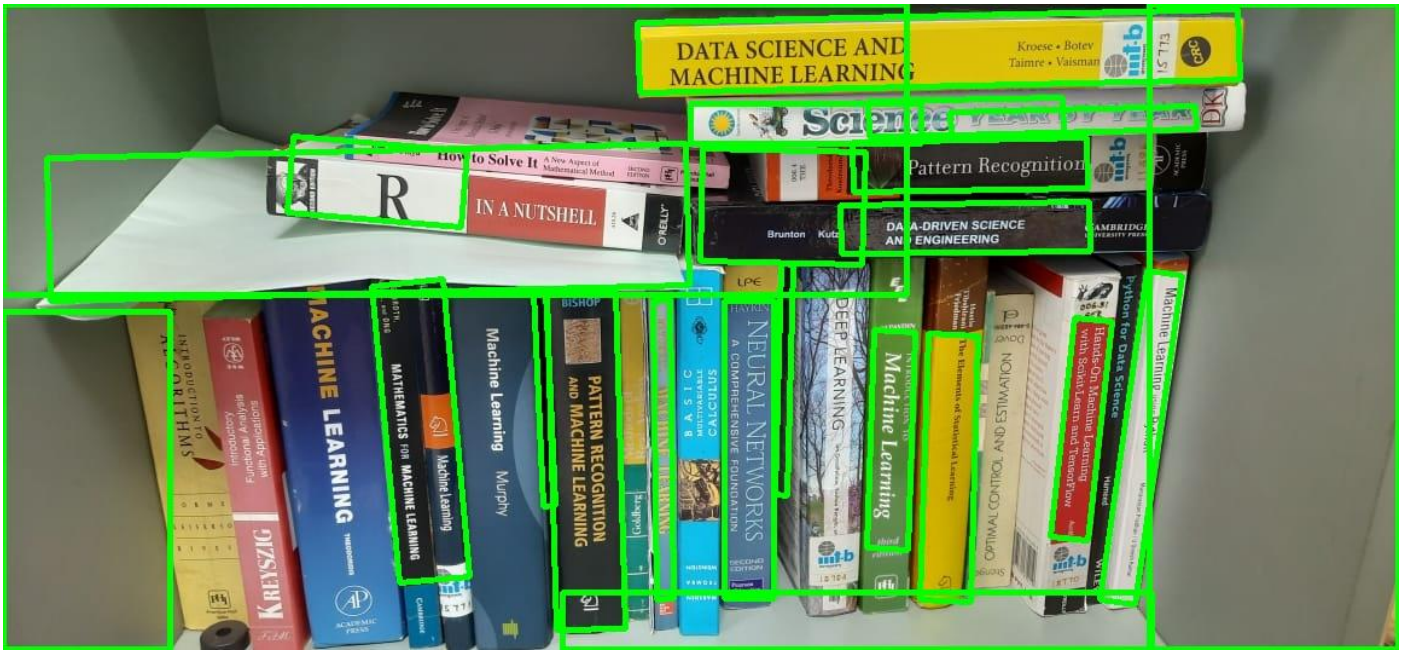


Observation: The bounding boxes clearly detect the majority of the books very precisely. Hence, we can easily count the number of bounding boxes to roughly get the count of number of books present in the self.

- Number of Books: 23
- Number of Bounding Boxes: 23
- Number of True Positive cases: 20
- Number of False positive cases: 3
- Number of False Negative cases: 3

Result of the test image

I performed the process with same parameters in the test image and got this Result.



Observation: It can be seen in the test image that few of the books are detected clearly while few others are not. The reason behind this problem is that the threshold for canny edge detection and contour detection was specific for the train image and for the test image it needs a different threshold.

Task 2: Detect shadow regions on the road and inpaint it i.e. create an image without shadow on the road region

Given Image:

- I have initially experimented with K-means segmentation algorithm to segregate the shadow region. Tried with various 'k' to efficiently segregate the shadow portion but the algorithms did not perform well.
- Then I tried to segregate the shadow portion using LAB colour Space. I segmented the shadowed region in LAB colour space by upper limit and lower limit of the pixel values. The segmentation using LAB colour space performs better than k-means clustering but the result were not satisfactory.
- Then I move to Region of Interest (ROI) based technique to select the road area and created mask over the shadowed region. This approach gave a better result (shown below).

Final steps performed is given below:

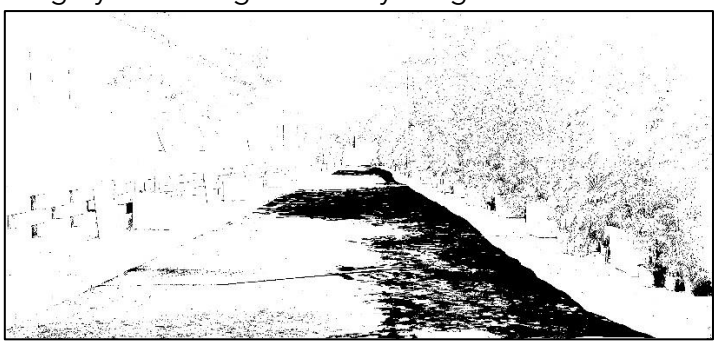
1. ROI (Region of Interest) based masking of the image
    a. Manually selected the Region of Interest (i.e. road)
    b. Converted that selected region into grayscale
    c. Applied thresholding on the grey region to create Binary image, after trying with multiple the thresholds, found that 13,255 gave a fairly good result.

2. Filling of masked pixels with value collected from nearby pixel of shadow in the road
    a. The process involved the meticulous gathering of pixel values from adjacent areas of shadows on the road manually.
    b. Through a systematic evaluation, the optimal pixel value was identified, resulting in an enhanced visual appearance.
    c. Ultimately, to achieve a uniform and pleasing outcome, all the masked pixels were uniformly filled with the selected value [149, 161, 171].

Result of the test image

I followed the similar process that I performed for train image. But only changed two parameters, i.e.

a. While converting from grayscale image to binary image here I used a threshold of 9,255.



b. And for filling the pixels I used a different value, i.e. [192, 200, 207]

Given Test Image

Removed shadow from Test Image