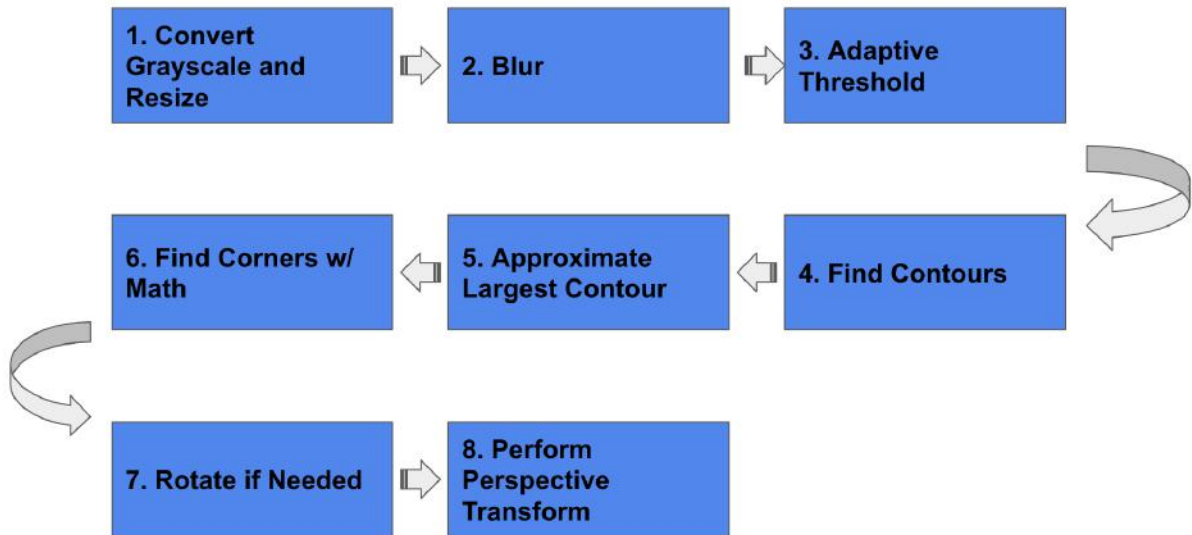


Approach

The current steps the model uses to transform the check are as follows.



I will go step by step and explain any notably different things I have done.

1- I resized all the images to 1290x790. There was no special reason for this size. While the aspect ratio will be changed in the image, the corners will still be found.

2- Gaussian blur is used with a smallish kernel. Setting sigma to larger values yielded poor results.

3- The lane line program was sensitive when using ordinary thresholding. So I chose a slightly more robust technique- adaptive thresholding.

4- Using library function, but the number of points returned is too much.

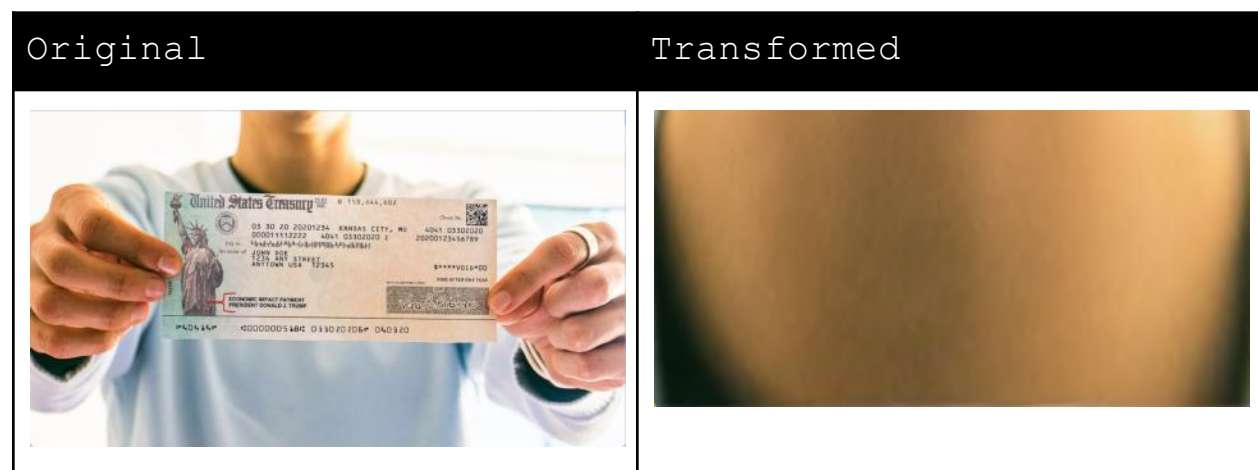
5- Approximation helps to remove some of the obvious outlier points. I still kept it kind of conservative with the epsilon value because if it removed too many points, I could lose corners. Anyway, next step will perform filtering.

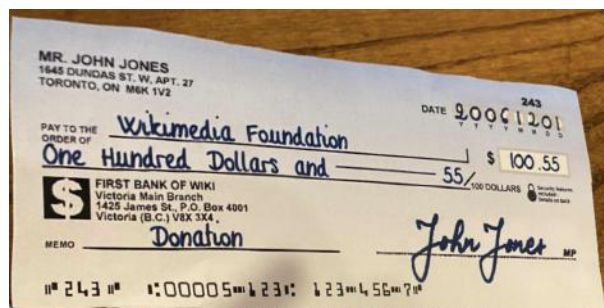
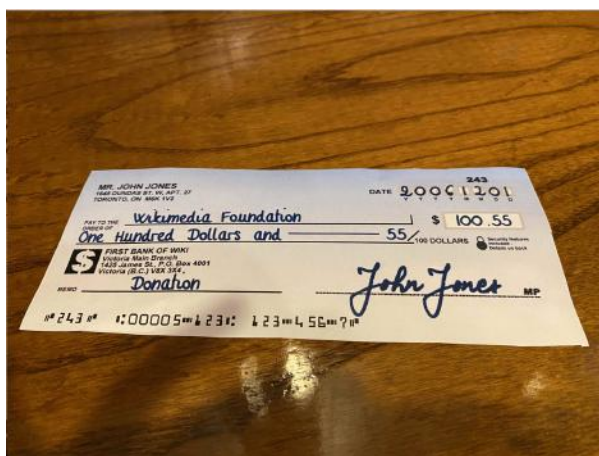
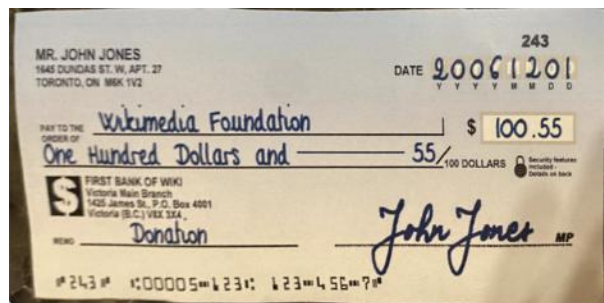
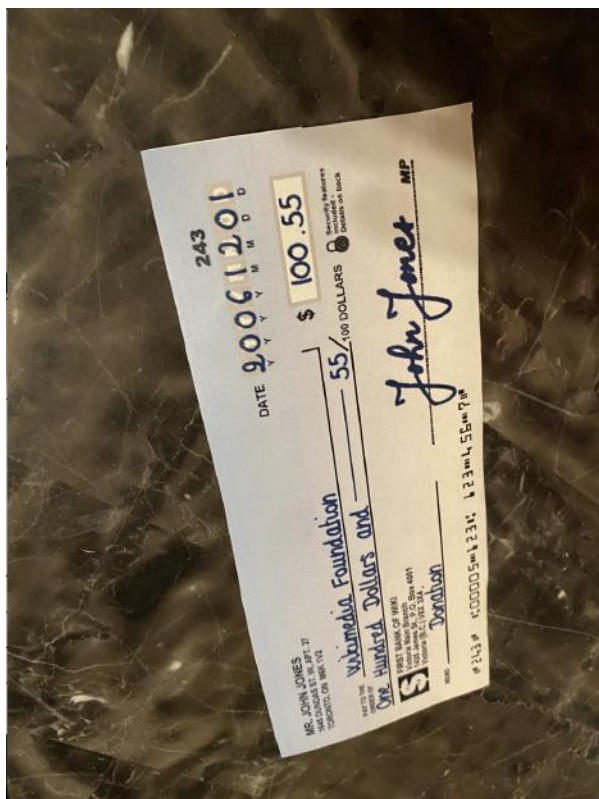
6- One can find the corners of the approximated contour using Euclidean distance. Sometimes the corners are obviously wrong (i.e. two points are on top of each other). In that case, we revert to using cv2.boxPoints.

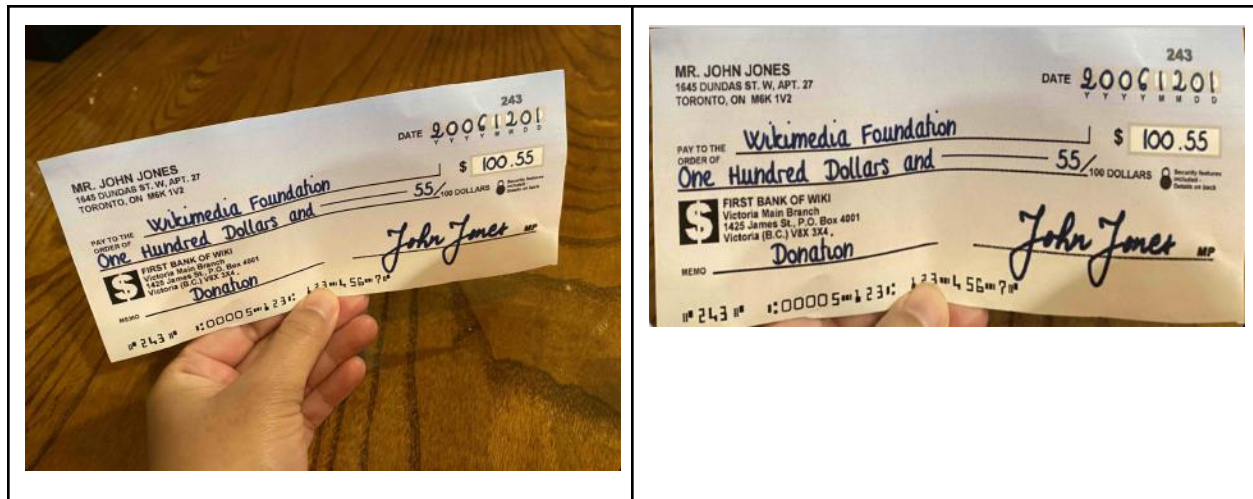
7,8 - No comment, done using standard libraries.

Sample results

You can see my results on the images below:







Limitations of this model

The model has a hard time if the contrast is not clear. The contour will not close. Also, the threshold is still a make or break step.