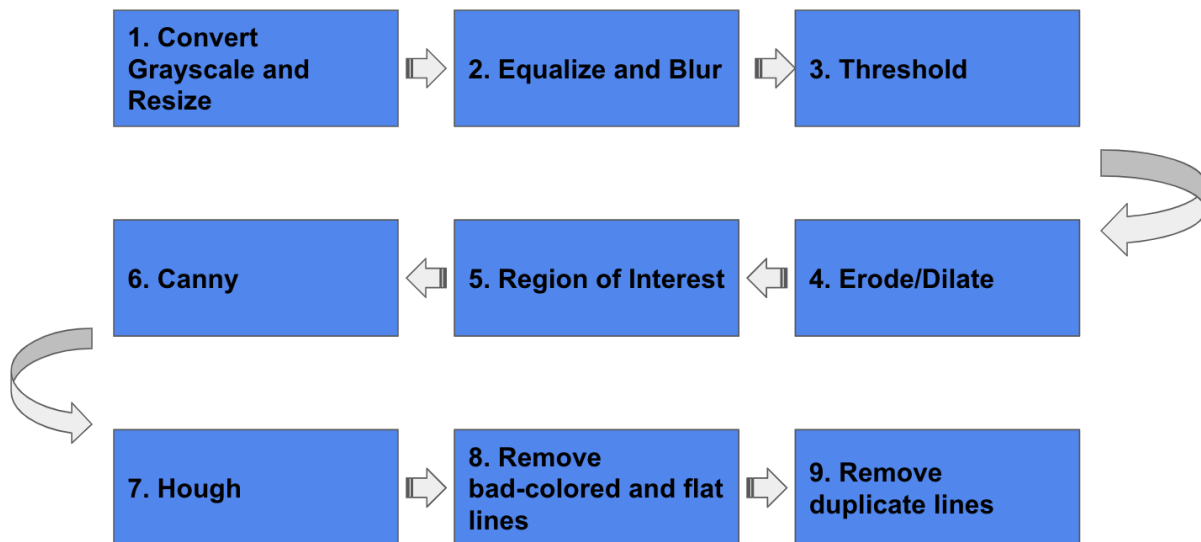


## Approach

The current steps the model uses to find the lane lines are as follows. You can also follow the code using these steps.



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

1- I resized all the images to 400x200 because this was approximately the size of some of the smallest images in the folder. While the aspect ratio will be off, the lines will still be found.

2- For example, reducing noise is accomplished using a small-kernel median blur. According to the documentation, this helps remove the salt and pepper

noise I was noticing. The kernel is 3x3 because I didn't want to lose too much info.

3- The program is incredibly sensitive to a bad threshold. So, I have chosen a threshold value that is not too aggressive. The histogram equalization performed in the previous step should help with this.

4- Erode/dilate helps remove further noise and bring the dashed segments a little closer.

5- The region of interest taken is toward the bottom of the image. We could have taken a larger ROI, but I wanted to avoid the sky/trees at all costs.

6/7- Canny and Hough is self explanatory. But, finding the right threshold value for Hough was a bit difficult. A value of 9 or so picks up a couple of extra lines, and past 20 was too selective. 12 was a good medium.

8- We then filter the lines based on three conditions. We want to avoid flat lines, because the lane lines are more likely to be vertical. We also want to filter based on color, because the lane lines should be white or yellow (red lines are ignored b/c they are a rarity).

9- Lastly, duplicate lines are those whose distance between starts plus distance between ends fall within a certain threshold.

## **Sample results**

You can see my results on the images below:



For a total of 11 lines found. Of those 11, 10 are correct, 1 is duplicate. At least 9 lane lines were missed. This is a score of  $30-9-1=20$  out of possible 57.

### **Limitations of this model**

The model is too conservative. It also does extremely poorly detecting doubled lines- it will just split the difference and take one of them.