1 description

my pipeline consisted of 5 steps. Color selection

First, I set a color threshold cause there are two color I need, that is yellow and white referenced by https://www.rapidtables.com/web/color/RGB_Color.html. Cv2.inRange() would be used to select color,cv2.bitwise_or would be used to apply mask which concludes yellow and white. cv2.bitwise_and() would be used to apply need image. Figure 1 shows region of interest by color selection.

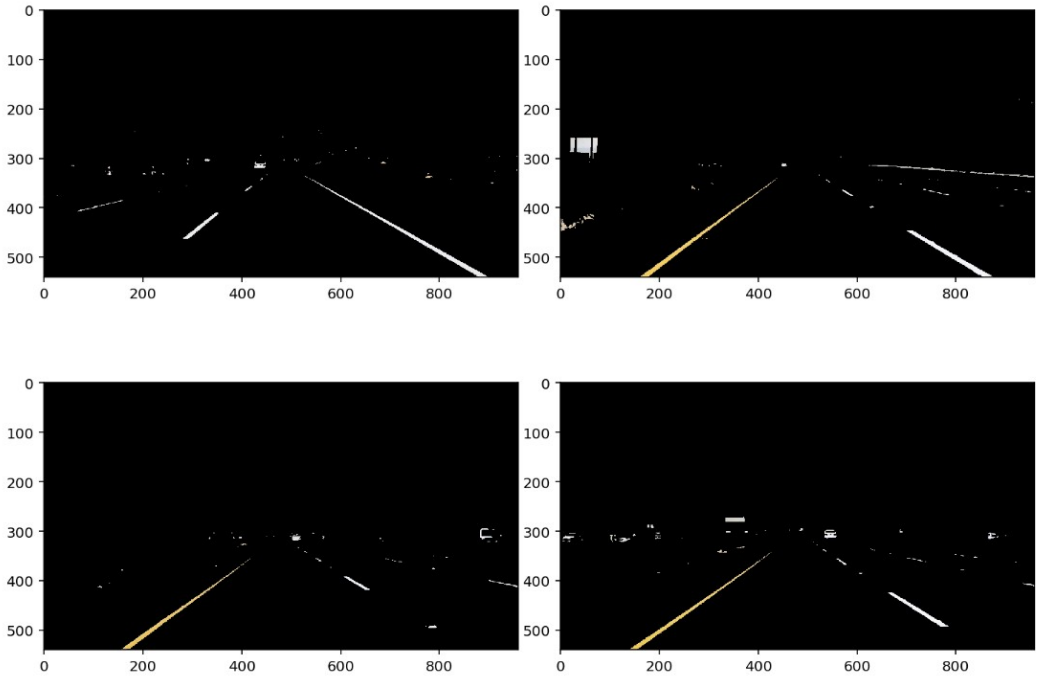

Figure 1 ROI color selection

The second step is grayscale to dowmthe image noise. Using cv2.cvtColor(image, cv2.COLOR_RGB2GRAY) to output image as seen in Figure 2.

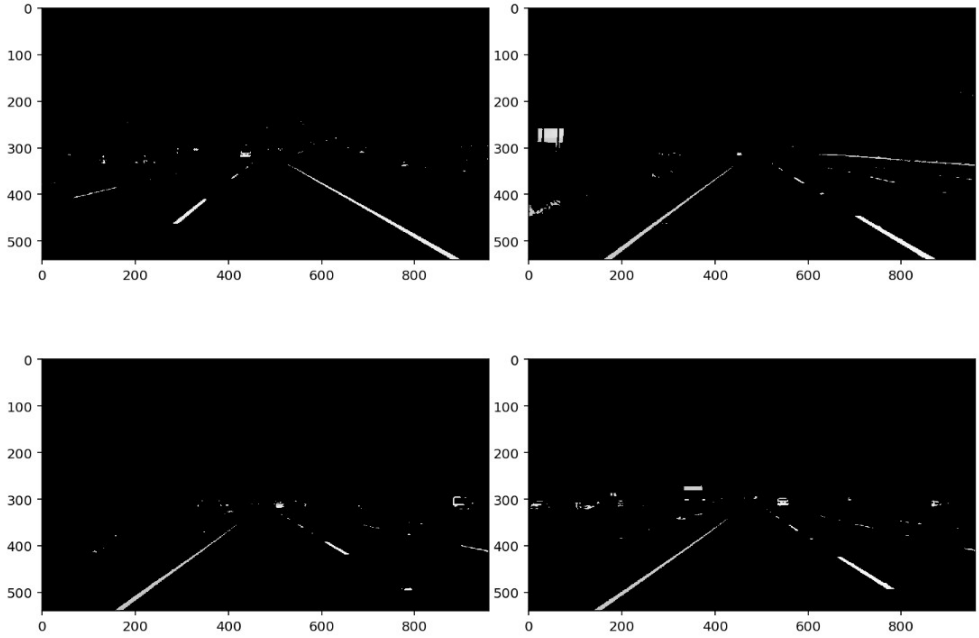

Figure 2 Grayscale

Step 3 must be Gaussian smoothing to smooth the image. I use kernel 5*5 to smooth image. As the figure 3 shows.

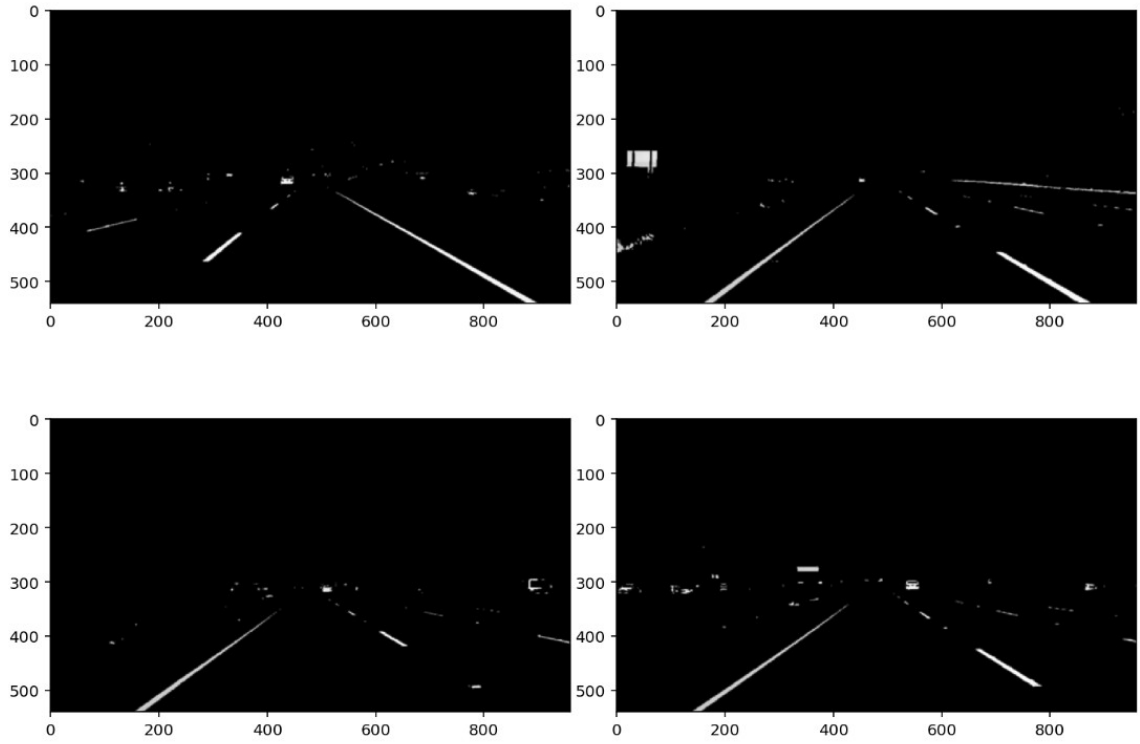

Figure 3 Gaussian smoothing

The next step is to perform edge detection on the output of the preprocessing. It was basically on the canny edge detection. I set the low_threshold 50 and high_threshold 150 due to the former leason.

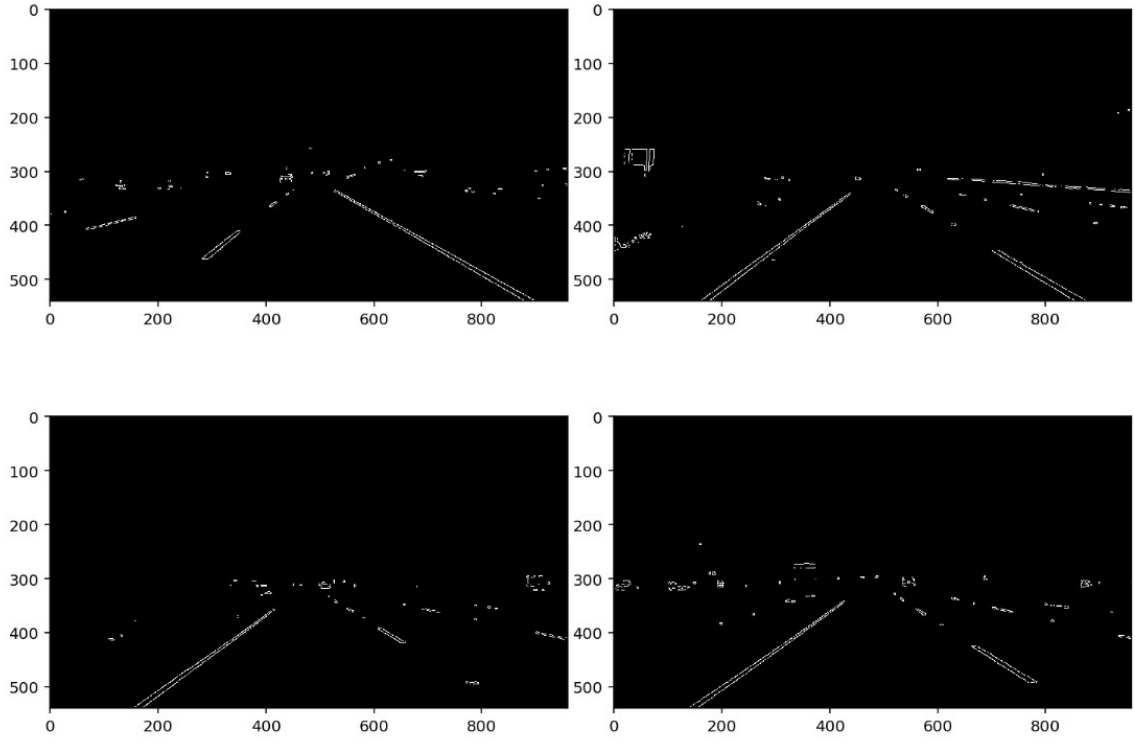

Figure 4 canny edge detection

The nest step I use the region of interest again to reduce needed range. There I used a ratio instead of real value.

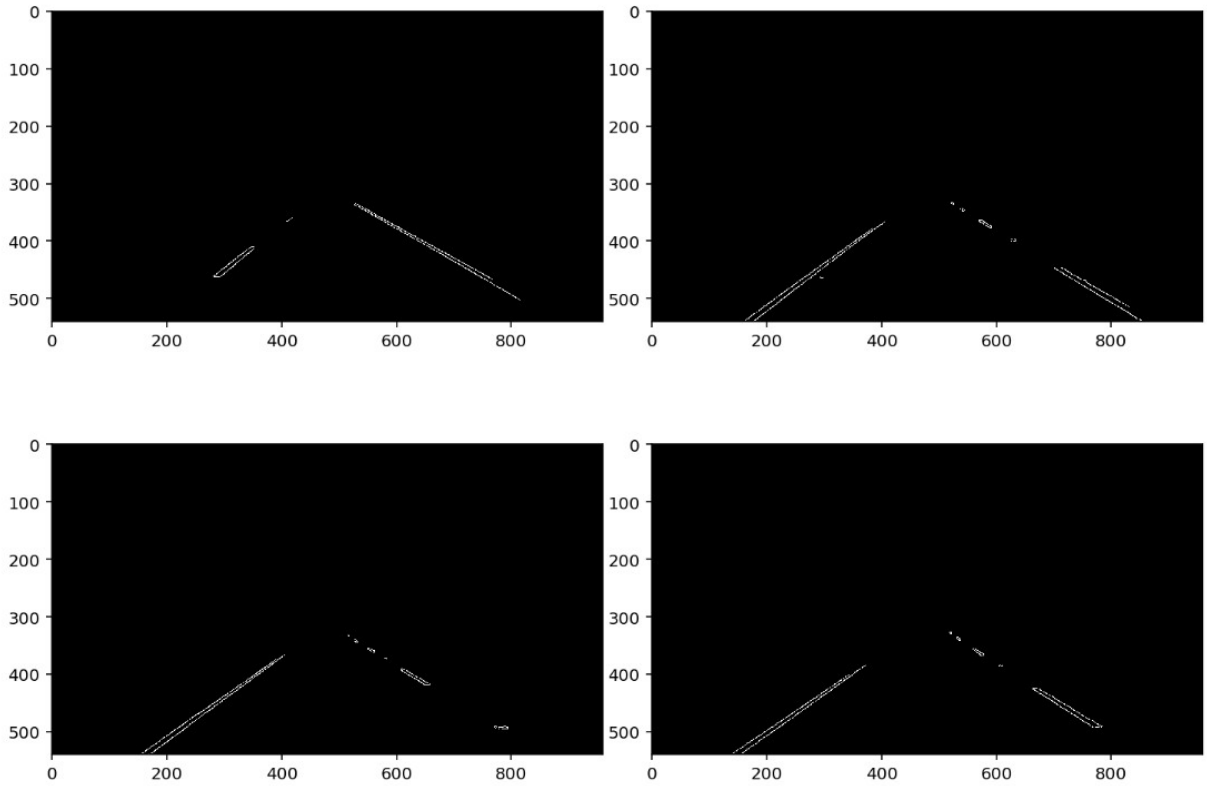

Figure 5 ROI region selection

The nest step is hough transform to convert dots to lines and draw lines in initial image.

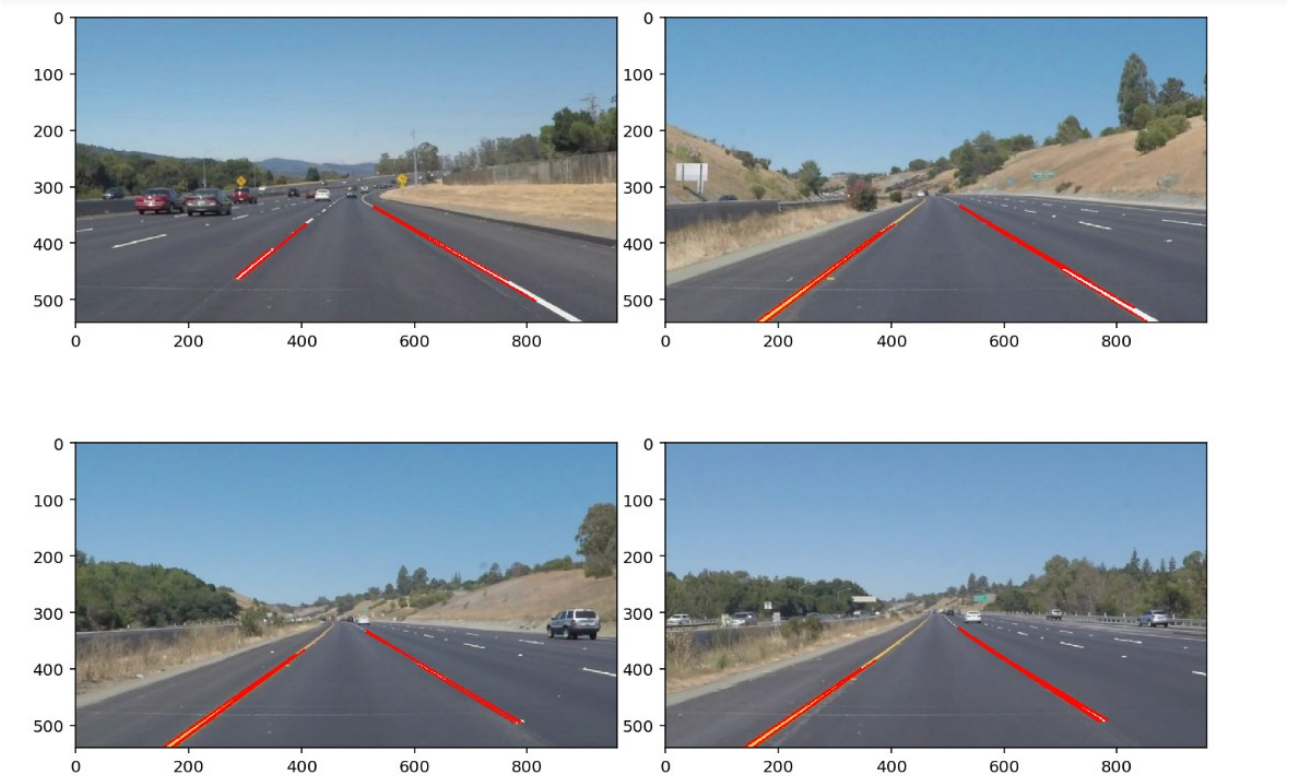

Figure 6 hough transform

2.potential shortcomings
One obvious shortcoming is that it is not good job for curved line.
3.reference
[1]https://github.com/naokishibuya/car-finding-lane-lines.git
[2]https://www.rapidtables.com/web/color/RGB_Color.html
[3]https://zhuanlan.zhihu.com/p/25354571utm_source=wechat_session&utm_medium=social