

Find Lane Lines on the Road

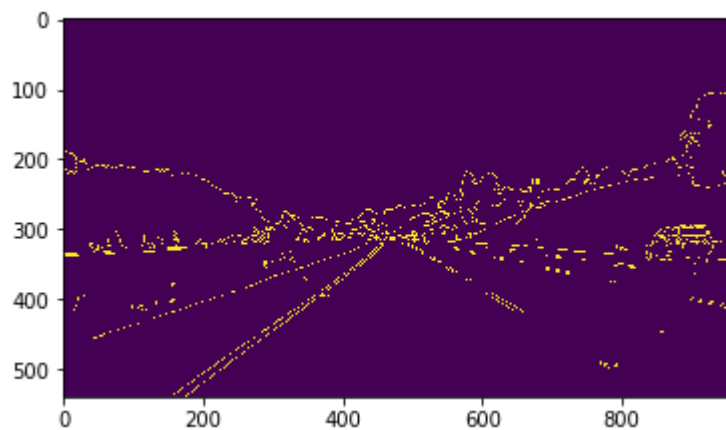
By Yaozhong Zhang

1. My pipeline description

There are basically two jupyter work in my submission. One for image, one for video. Their steps of lane finding processes are basically same, which has 4 parts:

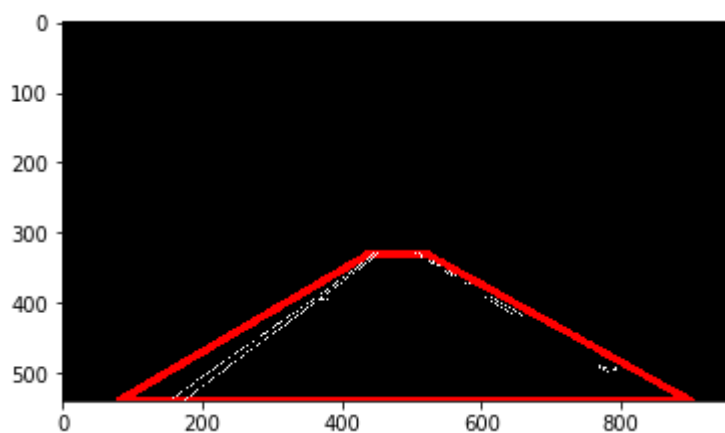
1). Canny edge processing

Make the image grayscale and use canny function



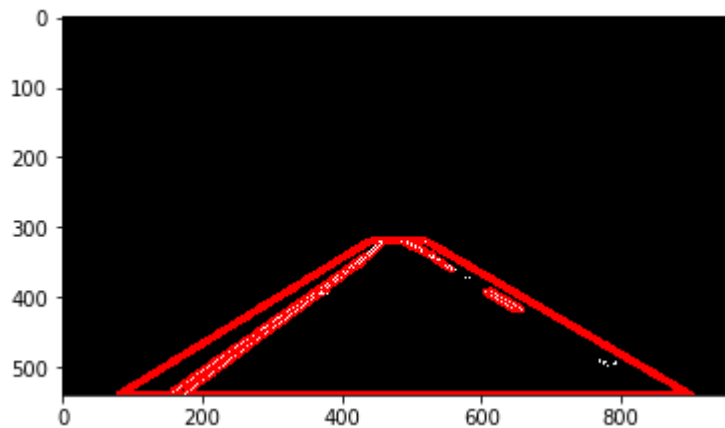
2). Masking

Find our interested area which has both left and right road lanes.



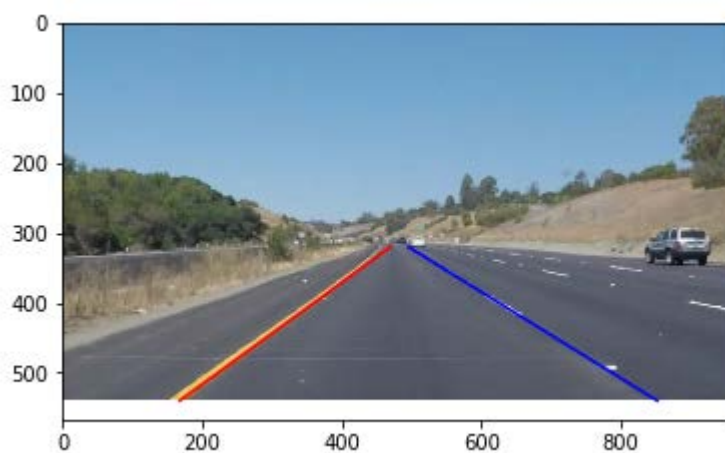
3). Hough transformation

Tuning the parameters until there are only lines we want



4). Extrapolate

This is the most interesting part since the approach is not introduced in courses. I have different ideas to process the image, but unfortunately I am not enough familiar with python's functions. Here I use a very straightforward approach that calculates the average value of the left/right line's slope and intercept. And I also use slope range to get rid of unreasonable lines. It works well. But it is still a little unstable, and I may apply Kalman filter later to reduce noise.



2. Identify potential shortcomings with current pipeline

As mentioned above, there is some noise in the lane detecting. Maybe I can

apply approaches like Kalman filter to make it more stable. Moreover, I think the masking area should not be fixed, since the lane area will change in condition of sharp turning. Also, this simple lane detector cannot perfectly deal with curve lanes. I also haven't finished the code for video "challenge". I will give my improved code later.

3. Suggest possible improvements to your pipeline

- 1) Kalman filter
- 2) If the result is unreasonable, we can save the previous frame's result to replace it.
- 3) Changes for video "challenge"