Finding Lane Lines on the Road

Writeup-Matthew Yuan

Finding Lane Lines on the Road

The goals / steps of this project are the following:

* Make a pipeline that finds lane lines on the road

* Reflect on your work in a written report

1. Describe your pipeline.

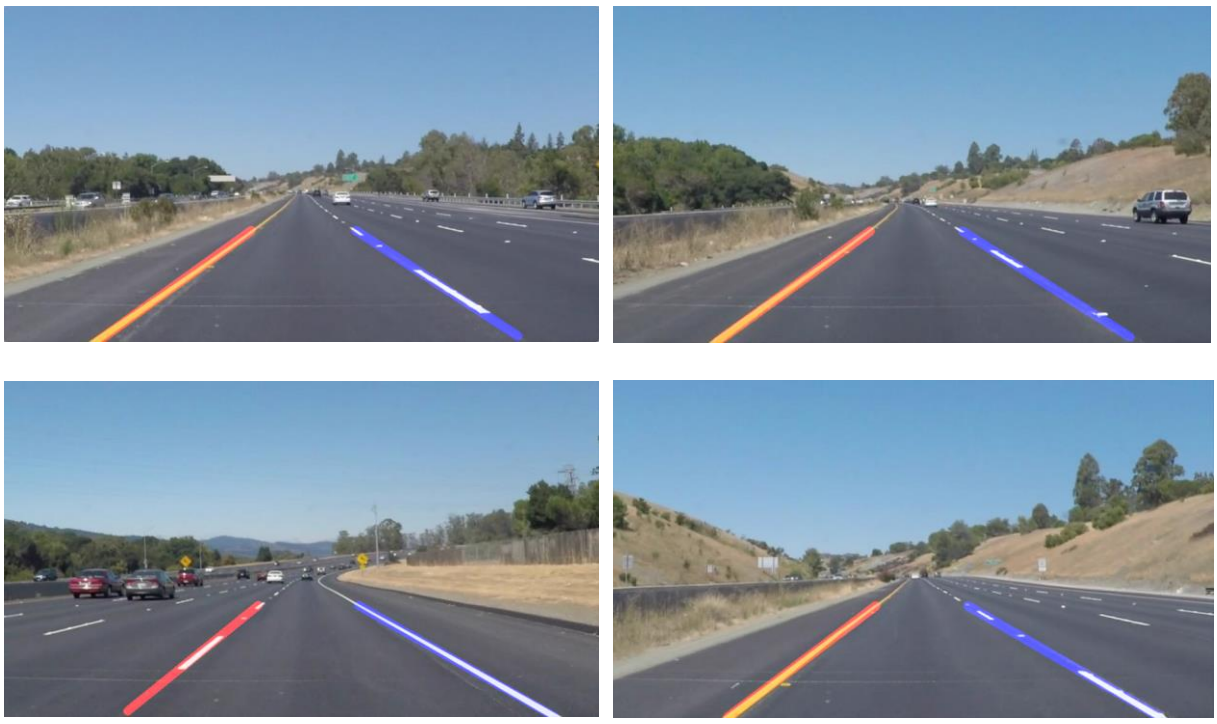
My pipeline consist standard steps learned in lessons.

1. Grayscale image
2. Gaussian image
3. Canny edge image and get part of image with ROI setting
4. HoughP transfer from Canny edge image then
5. Draw the pipelines—modified draw_lines function

The first 4 steps are kind of standard procedures which sample lines (points) for the step 5. It's hardest part to deal with the data. I choose Least Square Algorithm for the data processing in the first two videos. But it's failed with the challenge video.

My data processing is following steps:

- Separate the lines according to line slope: (y2-y1)/(x2-x1), into two lanes, namely left and right. Each consist sample points.
- Use Least Square Regression Line (LSRL) to train the data, and get the k & b with the left and right points
- Draw the lines, with using the k & b, from bottom to top which is around between 520-340 on y axis.



2. Identify potential shortcomings with your current pipeline

The picture is fine, but there are potential shortcomings with video.

1. When taking threshold higher which could cause no qualified line, and further lead to processing error. Need to add solution how to deal with such situation. Keep code more logical.
2. There are noise sample lines in challenge video, even in the second normal one.

3. Not sure if the self-car is following the straight lines in the picture/video. If it is not kind of long straight line road, like S road, the slope way may not work.

3. Suggest possible improvements to your pipeline

1. Code logic improvement to keep out of bugs.
2. The draw_line is totally disordered in challenge video. The algorithm may not lead to the disorder. The data need to be pre-filtered to keep rid of noises. Lack of digital signal processing tool, I cannot test and implement a certain one now. Maybe low-pass filter I learned long time ago.
3. Find proper Region of Interesting with real situation. May need to combine the speed and camera view to choose safety region.
4. Lower the threshold and other parameters for Hough line sample, pieces of short lines (points) would be used to closer the real line/shape.

A lots of thing need to learn: advanced python programming, digital data processing, neutral networking and etc.