

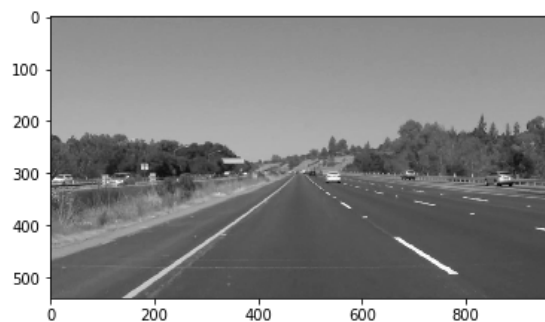
Project: Finding Lane Lines

Overview

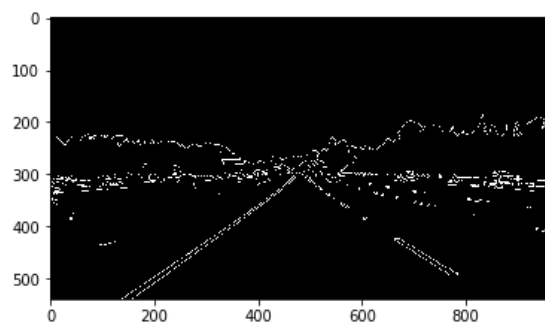
The main goal of this project is to uniquely identify two boarder lane lines on driving tract first on a set of images and then on a video stream. The main tools I use in this project are Python and OpenCV

Describe the current pipeline

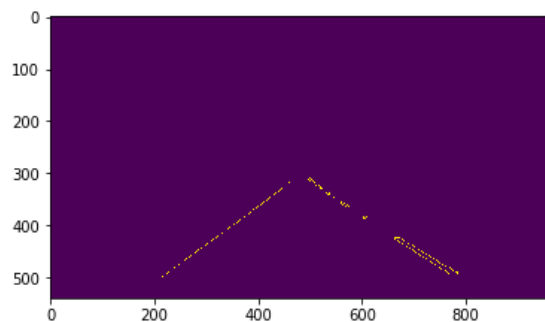
My pipeline consists of 3 steps. First, I converted the images to grayscale.



Then I apply Gaussian smoothing and Canny edge detection algorithm to the grayscaled image.



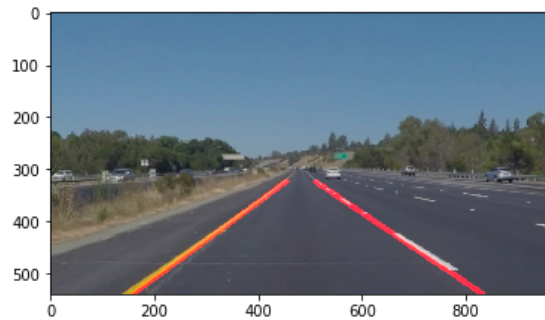
In the third step I use a quadrilateral mask to extract region of interest,



and run a Hough Transform on the edge-detected image. In order to draw a single line on the left and right lanes, I modify the `draw_lines()` function to three steps:

- 1) computing average values for slope(m) and intercept (b) for each of the lane lines
- 2) explicitly define y_1 , y_2 values for each lane and use averaged slope and intercept values to calculate x_1 and x_2
- 3) draw lines according to x_1 , x_2 , y_1 , y_2

Finally I got something like this



Identify potential shortcomings

One potential shortcoming is that the pipeline may fail to correctly mark out lane lines on a bendy road scenario.

Another shortcoming could be poor performance when the scene light changes

Suggest possible improvements

One possible improvement to make is iteratively increase the y-coordinate-values for the region of interest. This will reduce the height of the region of interest.

As for the second shortcoming, applying more robust edge detection method, such as descriptor or machine learning, could be a better way

Reference

1. Udacity Self-Driving Car Engineer Lesson 3: Computer Vision Fundamentals
2. <https://github.com/victoradipster> python_examples Repository