Introduction

The objective of this project is to write a code that helps to identify lane lines on the road. First lane detection will be made from different images of roads and next the code will be applied in a video stream which is a series of images to detect lane lines.
The pipeline or series of steps to achieve the above objective are list below

1. Convert the image into gray scale
2. Performing Gaussian blurring (smoothening) on the gray scale image
3. Canny edge detection on the blurred image
4. Define region of interest and mask out portion of the image
6. Apply Hough transform
7. Draw line on the original image

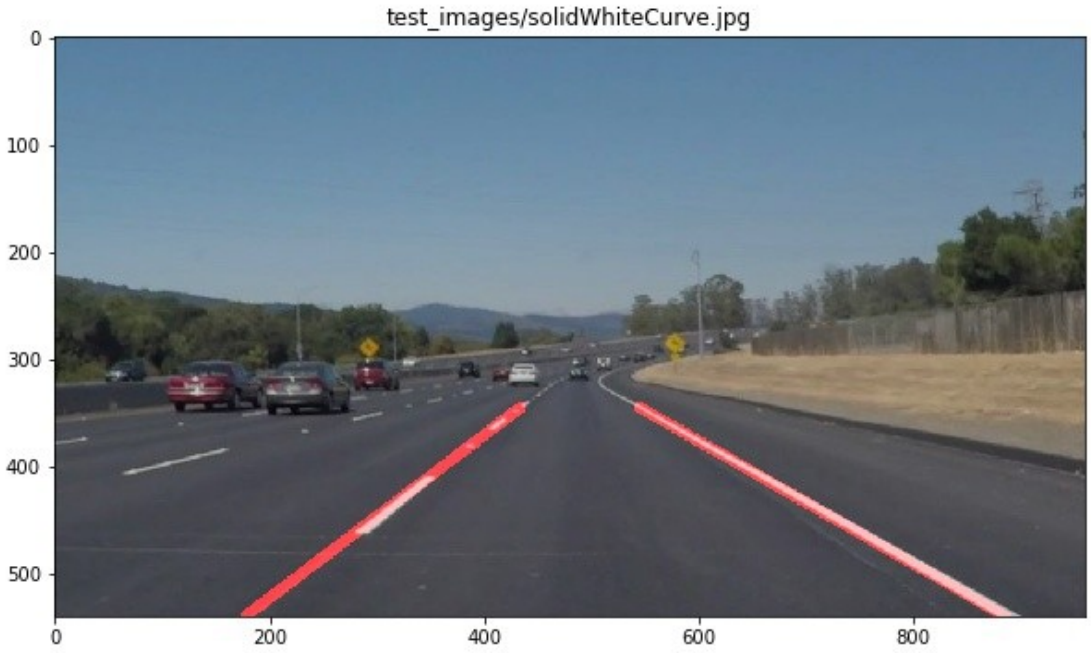
Reflection

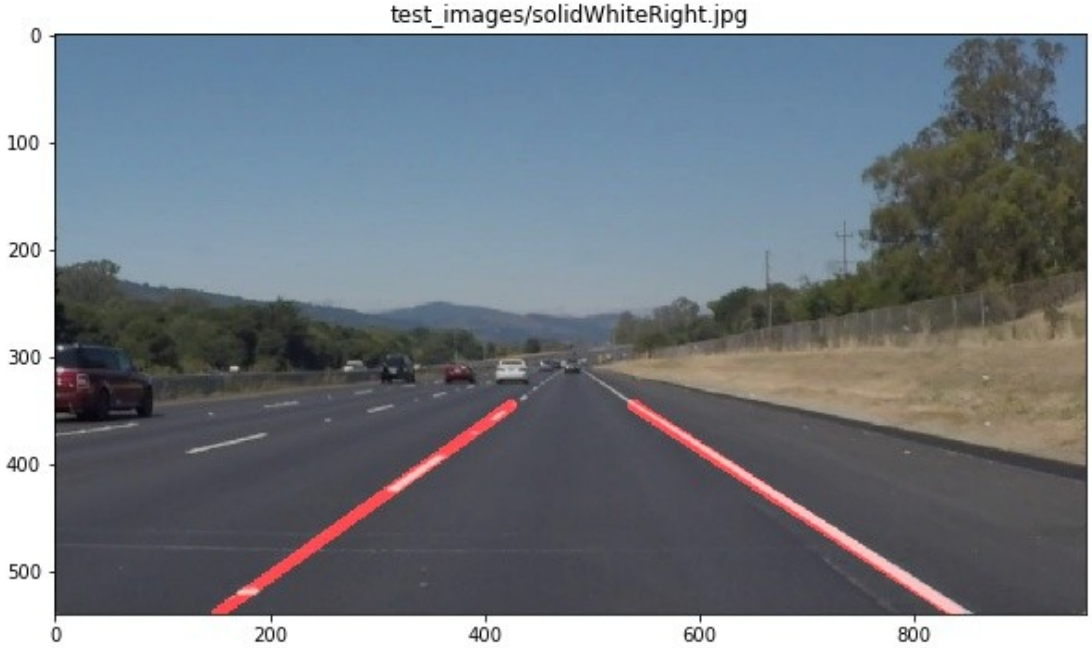
1. Description
The open source computer vision software (OpenCV) was mostly used to implement the above pipeline. The first step in the lane detection pipeline is to convert the color image with dimension of WxH*3 (where W is width and H is hight and 3 the color(RGB) or depth) into gray scale image of dimension W*H*1. This process helps us to identify edges based of difference in brightness of neighboring pixels(i.e by computing the gradient , (df/dx,df/dy)).

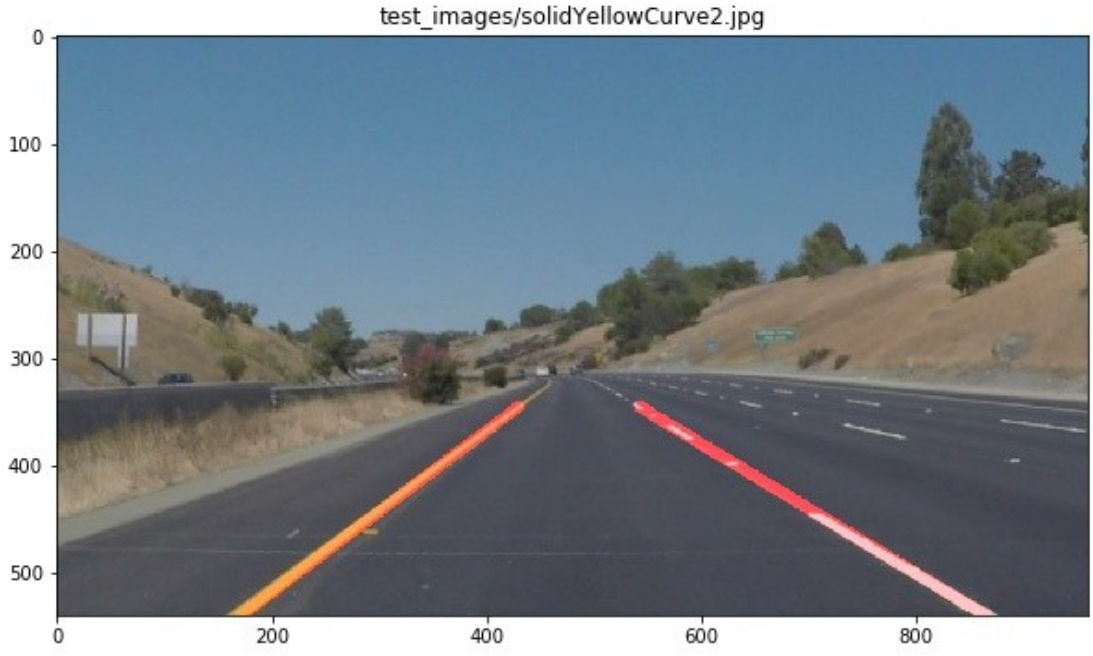
We don't need to do the gradient calculation by ourselves, the OpenCV library called Canny() can help us to identify edges on the gray image. But before performing the canny edge detection we need to smooth the gray image using the OpenCV library (Gaussian_blur()), the function can take n*n pixel and then average the pixel values which gives us blurred gray image, in the process the places with a strong gradient standout which make the edge detection simple. After canny edge detection is performed masking out the undesired area was done by specifies vertices on the image.

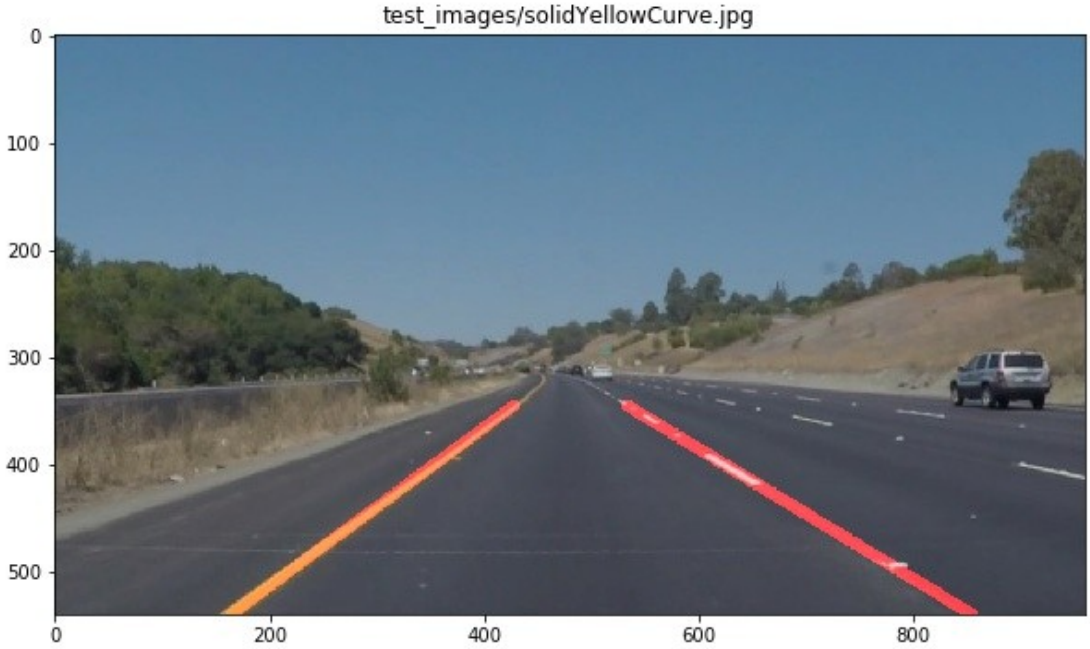
The last steps are most important in detection lanes. The Hough transform takes in the edge detected image above and converts them into many line segment which instead are composed of points. By manipulating the parameter of Hough transform we can identify line /lanes of our interest. However the Hough

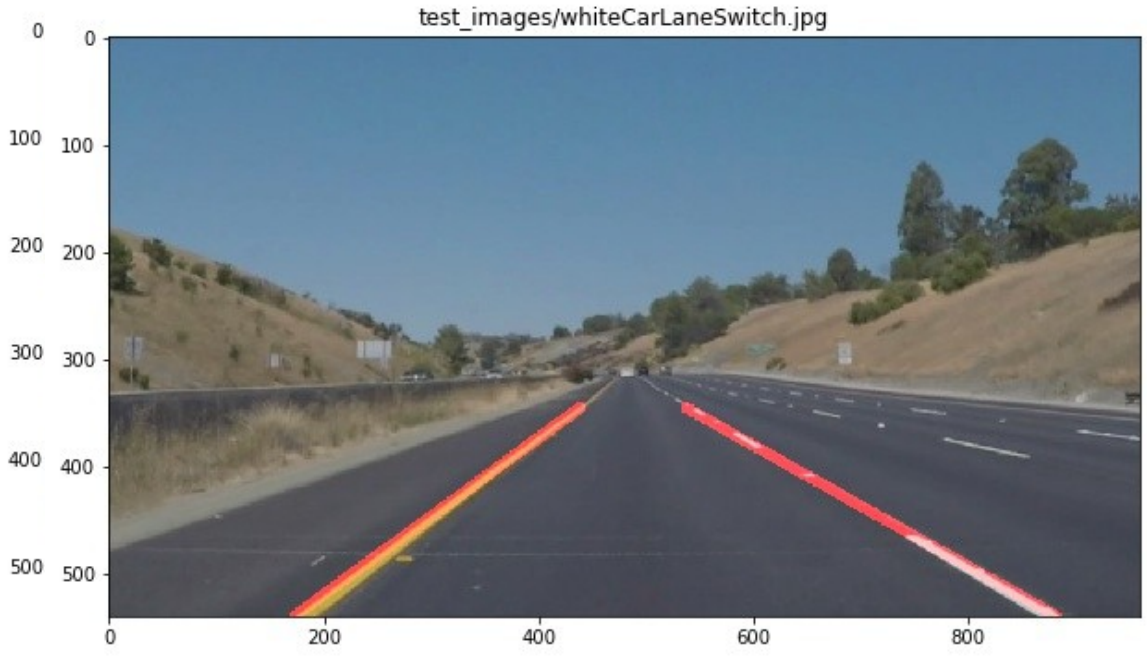
transform alone couldn't help us to detect the left line and right line of a lane. To do that we need a help function which takes all the lines detected by the Hough transform and identify the right and left line of a lane by based on their slope and finally extrapolate the line. This step in the pipeline is the most time consume and import step in the lane detection process. The final output for these process of lane line detection gives the following images.






test_images/solidYellowCurve2.jpg
0
100
200
300
400
500
0
200
400
600
800


test_images/solidYellowCurve.jpg
0
100
200
300
400
500
0
200
400
600
800



2. Potential Shortcomings

One of the most important thing in self driving car is the Lane line detection as the car used it to run forwards, to turn left or right and to change direction. So lane detection in both situation and condition must be taken into consideration while processing lane detection. All the lane line detection done in this project are performed on road images mostly taken on day time and the lane color are also visible clearly in most of the images.

3. Suggestions

I recommended to include road images which are taken in during night time,rainy condition and also images that consisted of faded lane line to see how lane detection pipeline above performs on this situation and conditions. I also would how the self driving car works on roads with out lane at all such as roads which are not concrete asphalt roads or in rural area roads etc.