Finding Lane Lines on the Road

Lane detection is one of the most important aspects of autonomous navigation, which can be achieved by using computer vision techniques.

The following techniques are used:

- Grayscale Transform
- Gaussian Smoothening
- Canny Edge Detection
- Region of Interest Selection
- Hough Transform Line Detection
- Merge original Image with lines

Description:

1. **Grayscale Transform:**

   The input images are converted into gray scale images, in order to detect edges in the image. This is because the Canny edge detection measures the gradients.

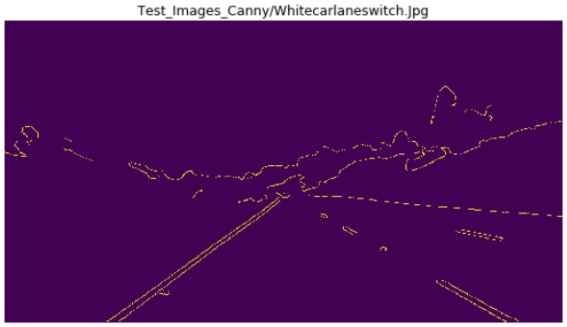
2. **Gaussian Smoothening:**

   When it comes to edge, the pixel intensity changes rapidly (from 0 to 255) which causes noisy edge detection, to avoid this we need to make edge smoother using Gaussian Smoothening.

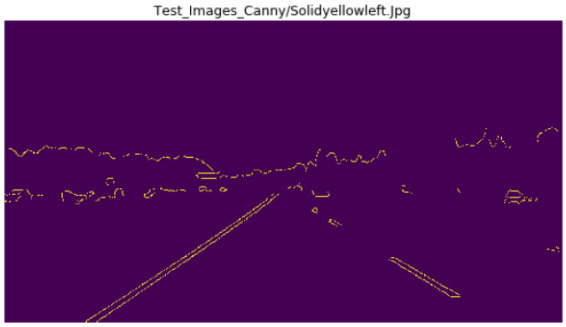
3. Canny Edge Detection:

The Canny Edge Detection use the fact that edges have high gradients. I.e., (how sharply image change from dark to white)

Test_Images_Canny/Whitecarlaneswitch.jpg



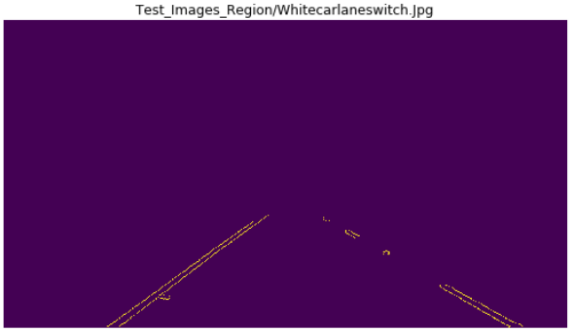
Test_Images_Canny/Solidyellowleft.jpg



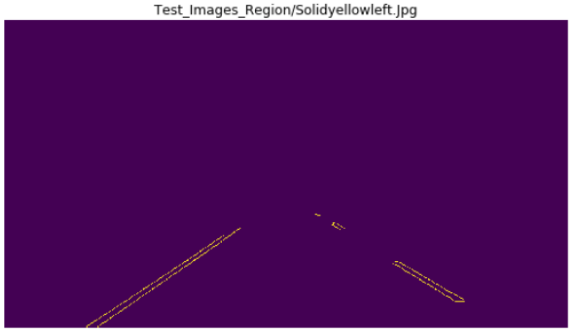
4. Region of Interest Selection:

When finding lane lines, we need not consider parts of image which is not part of the road.

Test_Images_Region/Whitecarlaneswitch.jpg



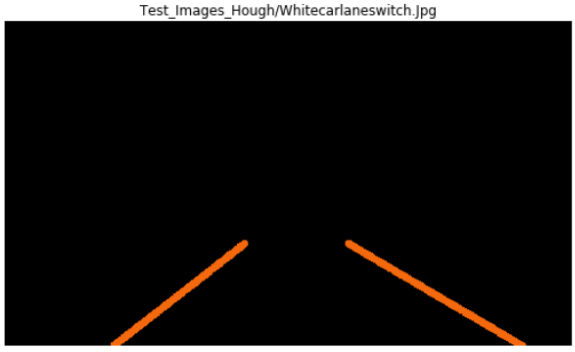
Test_Images_Region/Solidyellowleft.jpg



5. Hough Transform Line Detection:

The Hough Line Transform to detect lines in the edge images and draw lines on the lanes.

Test_Images_Hough/Whitecarlaneswitch.jpg



Test_Images_Hough/Solidyellowleft.jpg

Averaging and Extrapolating Lines:

- There are multiple lines detected for a coordinate (lane line). We should come up with average line for that.
- Also, some lane lines are only partially recognized. We should extrapolate the line to cover full lane line length.
- We want two lanes: one for the left and the other for the right, the left lane should have positive slope and the right lane should have a negative slope. Therefore, we'll collect positive slope and negative slope separately and take averages.
- Algorithm Logic:

```
#logic:
#   for each lines:
#      calculate Positive and Negative slopes, intercepts and store them in separate arrays.
#   find the minimum of all the y coordinates [y_min] and adjust it with line parameter.
#   find average of the positive-negative slopes and intercepts.
#   calculate x_min and x_max using equation x = (y-intercept)/slope
#   draw line with coordinates (x_min, y_min) and (x_max,y_max)
```

6. Merge original Image with lines:

Finally, drawing lane lines on the video clips. This is achieved by calculating the average lane lines across multiple video image frames to make it smoother.

Possible Improvements to Pipeline:

Improvements can be made to the draw_line algorithm to work better with curved lanes and other marks also from tree shadows.