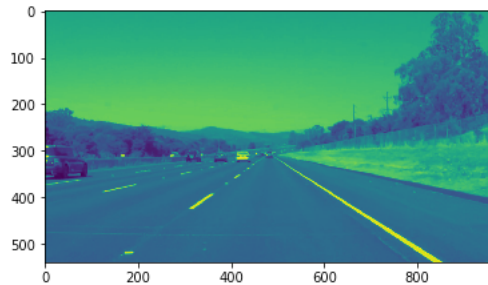


Project 1:

Overview

Summary of applied steps to test images and video for detecting lane lines:

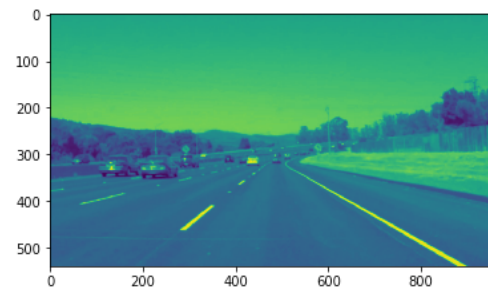
1) Converting test image to grayscale



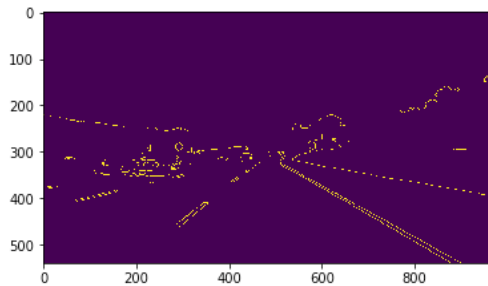
2) Apply gaussian blur to grayscale image to suppress noise and spurious gradients

```
plt.imshow(gaus_img)
```

<matplotlib.image.AxesImage at 0x1c2b67ca20>

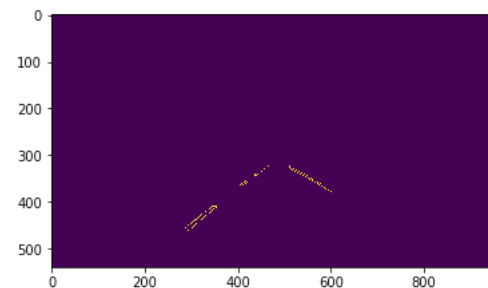


3) Apply canny edge detection to smoothed image from Step 2

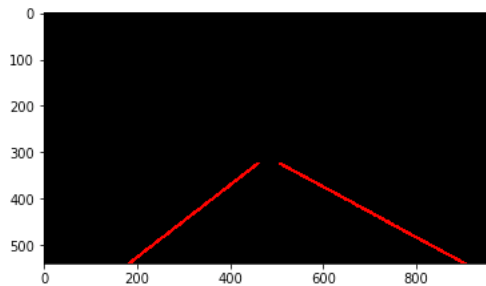


4) Defining vertices for applying region of interest

5) Apply region of interest on canny image from step 3(masking and extracting section of image which we perceive as consisting of the driving lane)



- 6) Apply Hough transform on image from step 5 for representing lines in parameter space and extrapolating the detected line segments to show solid lane lines (connecting spacing between intermittent lane lines)



- 7) Drawing the extrapolated solid lane lines onto the original image



Once we successfully detect the lane lines on test images, pipeline is also applied to the test video.

Output video files saved to folder path - ./laneFinding/test_videos_output

- solidWhiteRight.mp4
- solidYellowLeft.mp4

1. Describe your pipeline. As part of the description, explain how you modified the `draw_lines()` function.

The created pipeline has been described above along with corresponding output for each step.

In order to draw a single line on the left and right lanes, I added an extrapolate function which separates line segments by their slope $((y_2 - y_1) / (x_2 - x_1))$ to decide which segments are part of the left line vs. the right line i.e if slope is less than 0, then consider detected line segment as part of left lane, otherwise, consider detected line segment as part of right lane.

NOTE: As part of error handling, to avoid undefined value of slope ($m \sim \text{infinity}$) in line equation $m = (y_1 - y_0) / (x_1 - x_0)$, we skip the value when $x_0 = x_1$

2. Identify potential shortcomings with your current pipeline

While applying the pipeline to test video, there were certain frames where the drawn lines were not aligning with the actual lane lines.

When applying the pipeline to the optional challenge video, the drawn lines were not aligning with the video lane lines where the road had curves.

The pipeline would not work if for any test image/video frame, either of the left or right lanes don't have a consistent lane marking to be detected by canny function.

3. Suggest possible improvements to your pipeline

A possible improvement would be to fit a curve (parabola or ellipse) instead of using line equation for detecting lane lines in draw_lines/extrapolate functions.