

# Finding Lane Lines on the Road

---

## Writeup

---

### Finding Lane Lines on the Road

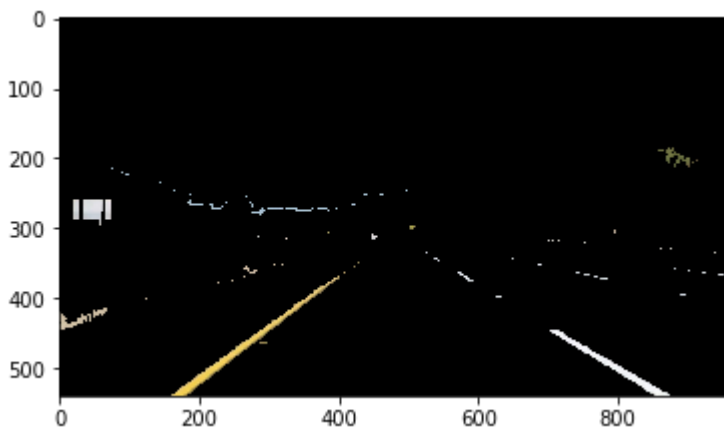
The goals / steps of this project are the following:

- Make a pipeline that finds lane lines on the road
  - Reflect on your work in a written report
- 

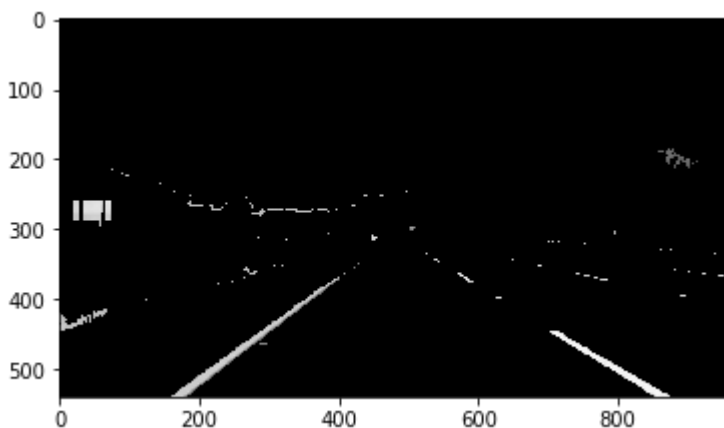
### Reflection

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

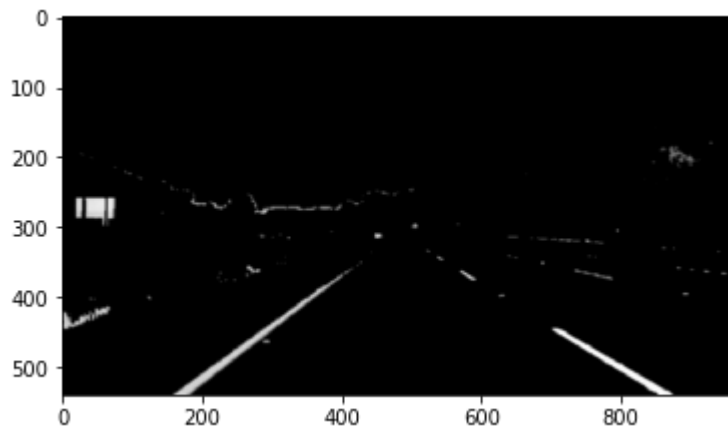
My pipeline consisted of 7 steps. First, I did a thresholding on the original image to get places (possible line locations) on the image with shades of yellow and white.



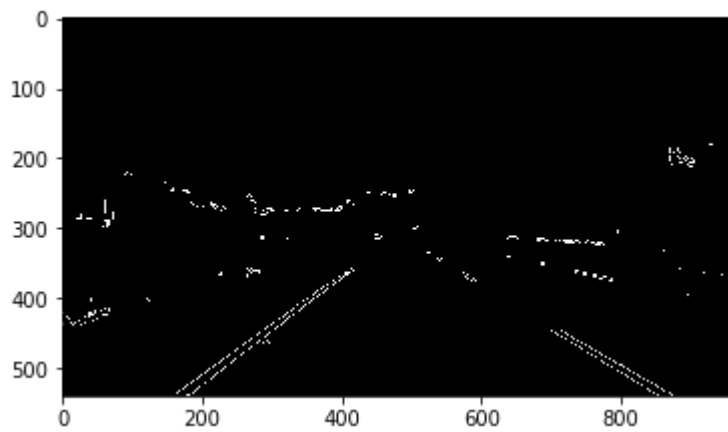
Then, I grayscaled the resulting image.



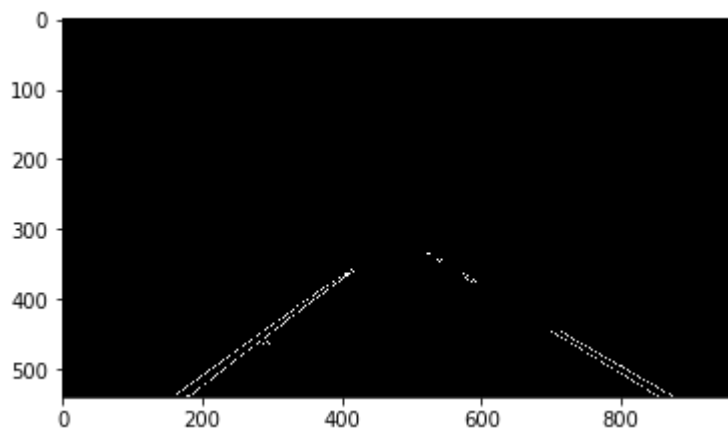
Then, I used gaussian blur to remove some high-frequency components on the image.



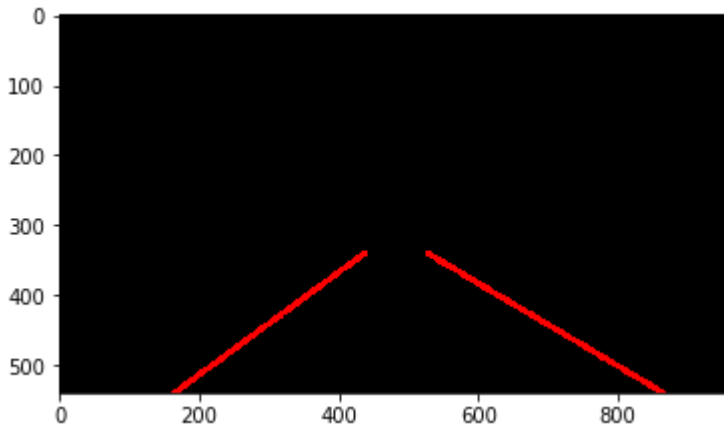
Then, I used canny edge detector to detect the edges on the image.



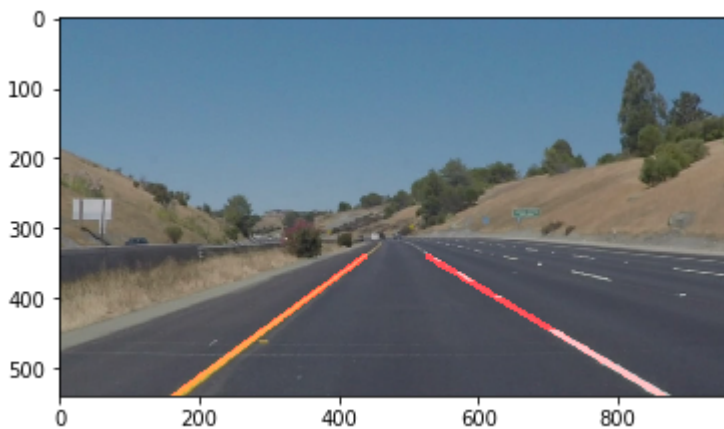
Then, I cropped the region of interest (ROI) for the lines.



Then, I used hough\_lines to draw the lines.



Then, I combined the drawn line with the original image.



In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by separating the left and right lines segments by their value of slope. I assume that left lanes are those with negative slope ( $< -0.2$  for instance) and the right lanes are those with positive slope ( $> 0.2$  for instance).

Then I used `np.polyfit()` or `cv2` version of RANSAC to fit for each of the left and right line. Then, to draw the lines, I used the bottom of the image and values close to the ROI as bounding boxes for lines. I drew the lines based on these fitted points so that it looks like there are one line for each of left and right lane.

## 2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen if the vehicle arrives at a curve, then the linear extrapolation fit would then not work on a curved line and may point off the road. This can be seen in the Optional Challenge video. There are also some fluctuations in the lane detection extrapolation especially with dashed lines

Another shortcoming could be the height of the vehicle, which might differ for another vehicle. Then, the ROI that we have defined could not work as well as in the videos that were given. Also, lines in some countries could be of different colours and the colours of the road (imagine snowing) could be lighter which causes problems on detecting lanes on those roads. Sometimes, there are arrows that point to highway exits that could cause the pipeline to go wrong.

## 3. Suggest possible improvements to your pipeline

A possible improvement would be to use an advanced ML technique to find the correct ROI, or we could refine our lane detection so that it ignores trivial points even without defining an ROI. This way, we could have a pipeline that runs well even with different height of vehicle or different angle of camera.

Another potential improvement could be to improve the colour detection method so that the yellow and white colours are not hard-coded threshold values. Additionally, we should add a more refined model that distinguishes the arrows from the lanes and can detect lanes even when the surroundings are similar in color to the lane (such as in snowy conditions, as mentioned in the shortcomings above).