

## 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:

1. First, I loaded the image



2. Then, I converted the image to grayscale



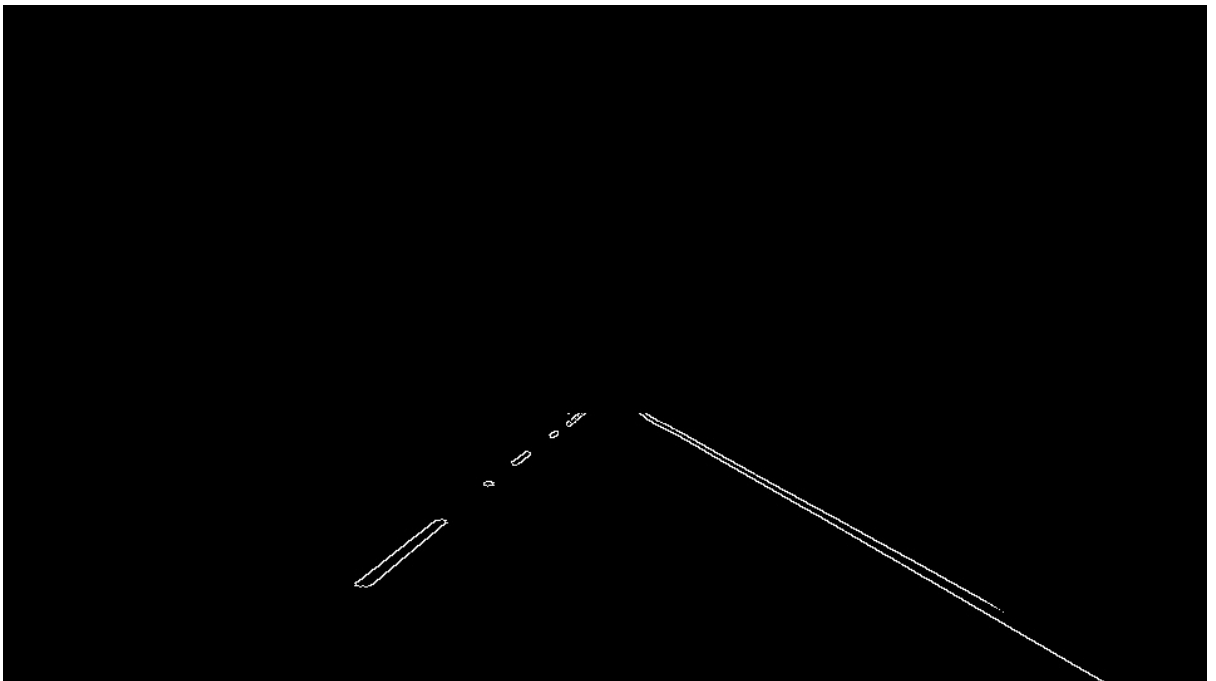
3. Then I used Gaussian Blur function to smooth the edges for better edge detection



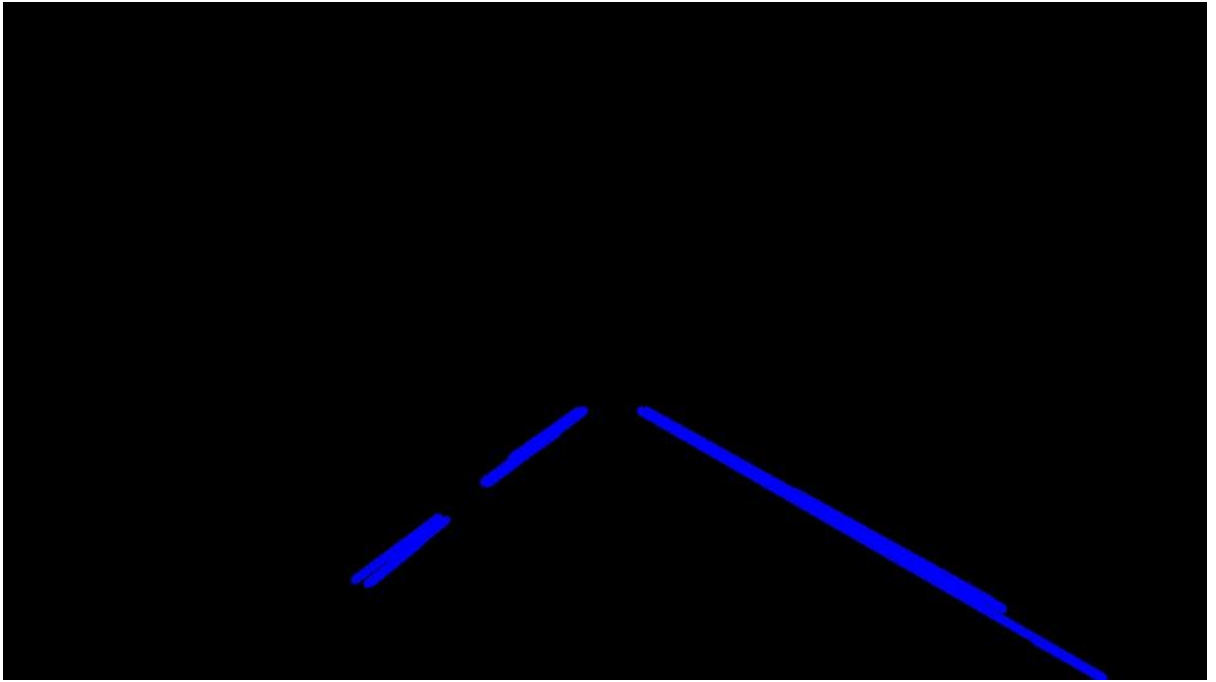
4. Now, I extracted the edges from the image using canny edge detection



5. I am interested in the region containing lanes, so I define a mask to reduce the scope to detect lanes. For this I used a quadrilateral shaped region to better restrict the region of interest



6. Then I used `houf_lines` function with various parameters specifying certain condition like minimum number of intersections (threshold), minimum line length, maximum gap between the line segments, angular resolution and distance resolution to get the actual lanes.

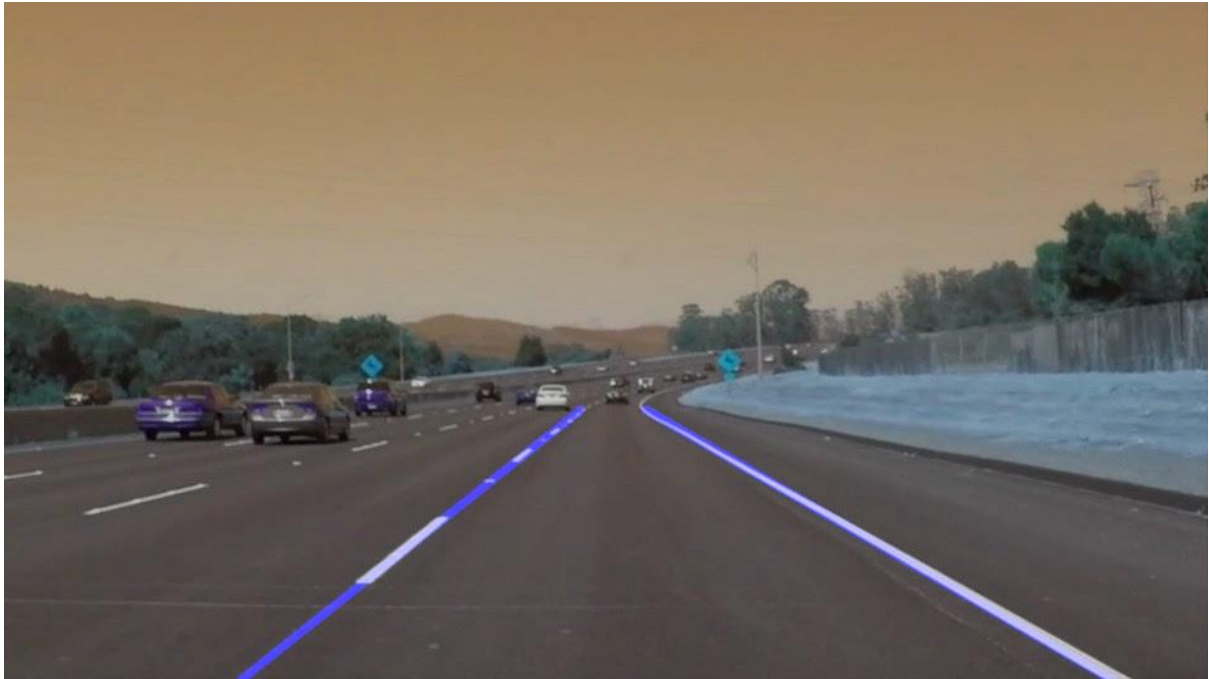


7. At last, I used addWeighted function to combine the lane lines image and the actual image to mark them.



In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by first calculating the slope of the line segments. Then based on the value of slope, lines will be separated in to two groups, one left and another right group. If the slope is negative, they will come in left group and if the slope is positive, then they will be in the right group and we will not consider slopes which are less than 0.5 and above 0 so that only extreme slopes are considered and we save the x and y endpoint coordinates to lists for each side. I have the top and bottom endpoint which I can connect to draw the lines along the lanes. Now I just needed to find the x positions for each line. For

I need to create a linear equation as a function of  $y$  to calculate the  $x$  values. For this I used numpy's `polyfit` and `poly1d` function. Once all values are calculated, I store it in an array and then looped over it to draw the line for left and right lane line and below is the result after running on the image.



## 2. Identify potential shortcomings with your current pipeline

The shortcoming of this pipeline is when there are curved lane lines which will not get detected properly because currently our region of interest is a strict straight one (quadrilateral) and it cannot adjust to the shape of the lane lines.

## 3. Suggest possible improvements to your pipeline

A possible improvement would be to somehow make the region of interest flexible with the change in the shape of the lane lines.