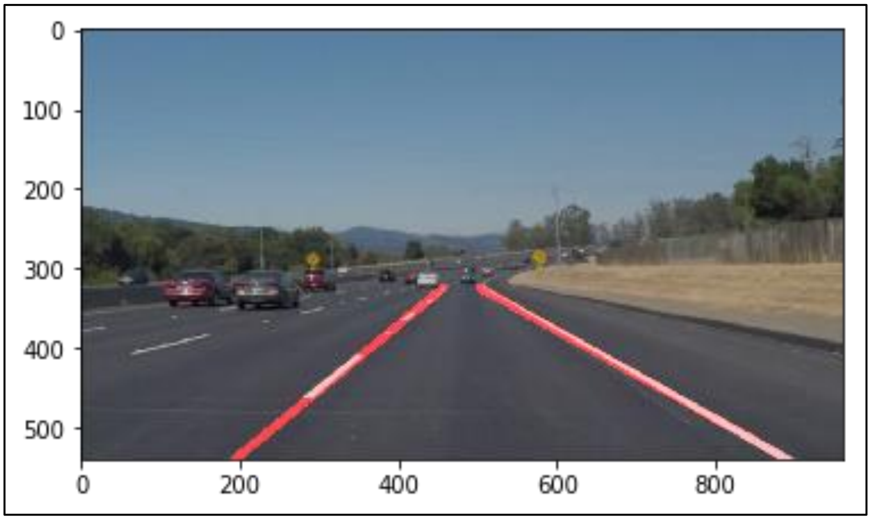
Finding Lane Lines on the Road

Goals of the project:

- Make a pipeline that finds lanes on roads
- Reflect on my work in a written report



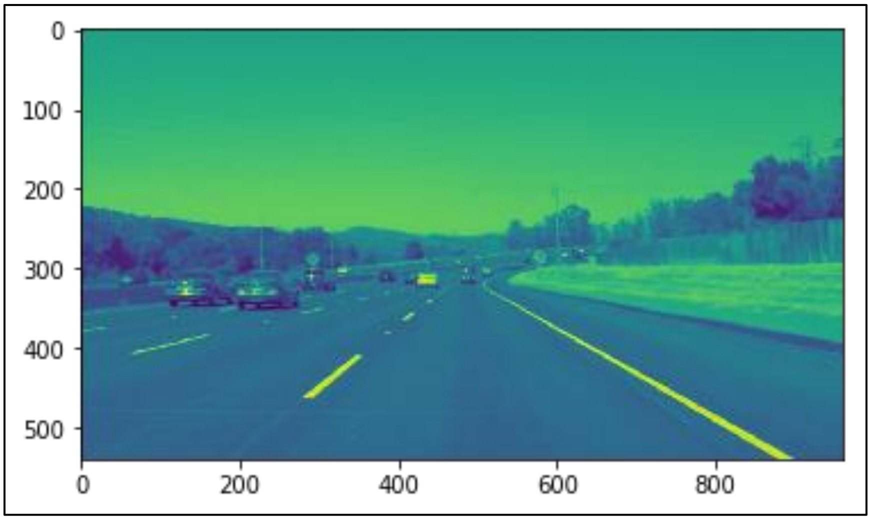
Reflection

1. Pipeline Description

The pipeline consists of the following 6 steps:

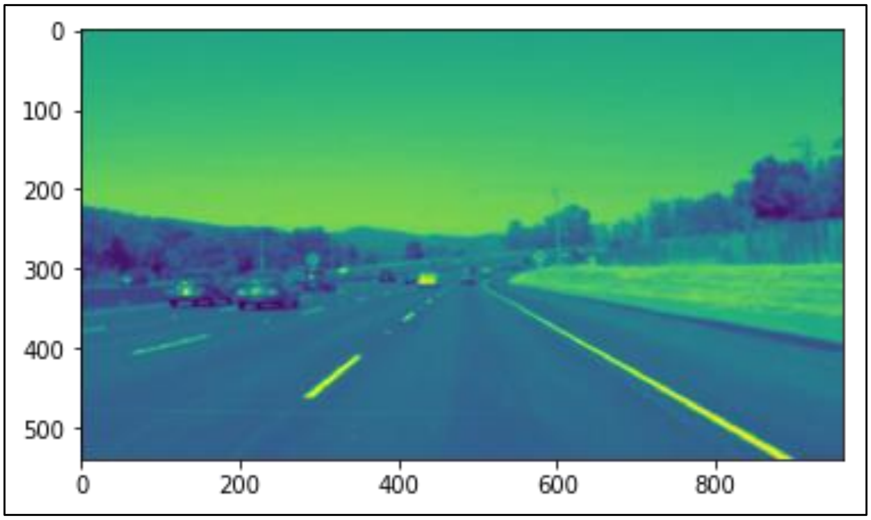
Step 1: Convert image to greyscale

The image/frame is converted to greyscale because it is easier to enable Canny edge detection on greyscale images. A copy of the image is saved for future use.



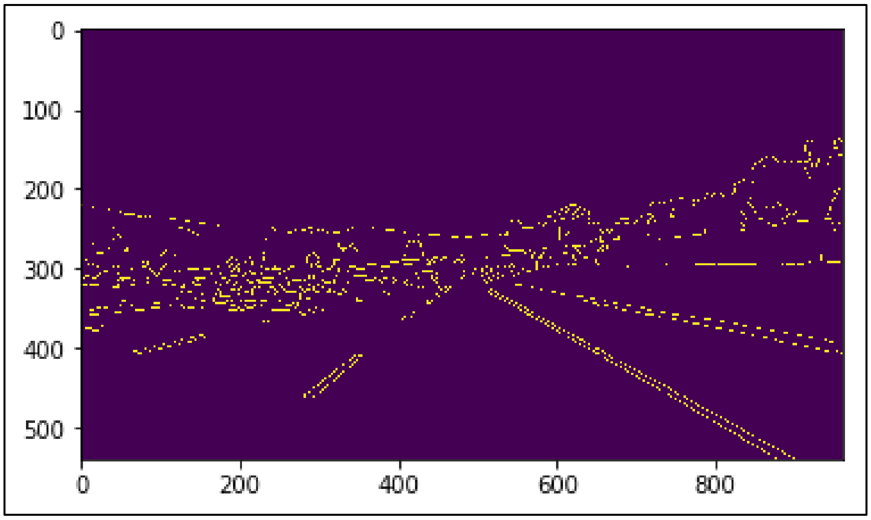
Step 2: Smoothen image

The image frame is next smoothened with a gaussian filter for noise reduction.



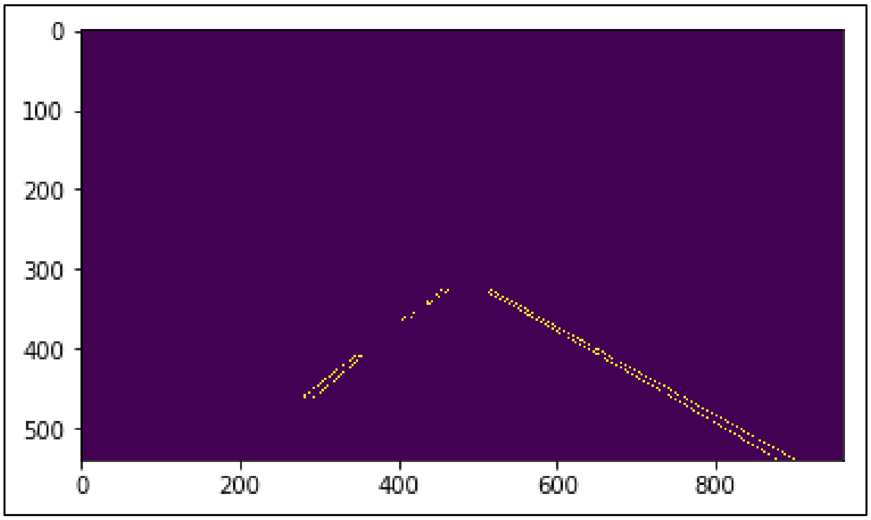
Step 3: Canny Edge Detection

The image is next convoluted with a canny filter to figure out the gradient of the image. This ill detect all the lines.



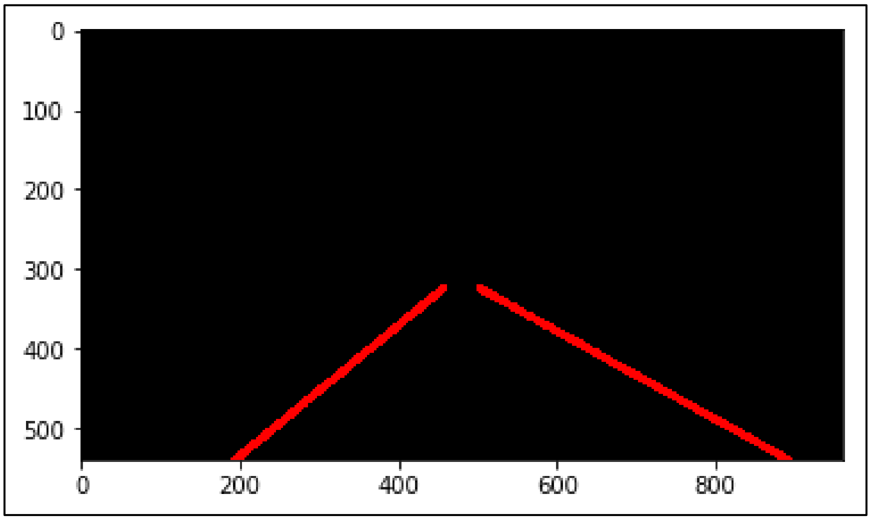
Step 4: Applying the mask

Assuming the camera is in a fixed position, we can crop out the area that is not of interest. This helps to remove unnecessary edges detected.



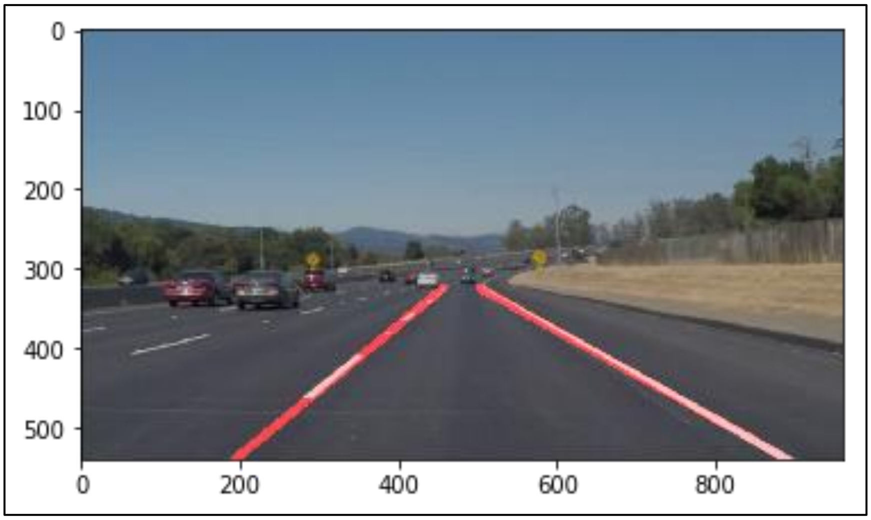
Step 5: Hough transform

Hough transform works by converting lines from x-y space into parameterized space. In the parameterized space, we can use polling to figure out which lines pass through most points in the detected image.



Step 6: Overlay with original image

Combine the above image with original images for better visualization.



2.Parameters Used

Gaussian Filter size	9
Canny Threshold Low	25
Canny Threshold High	75
Rho	3
Theta	Pi/50
Threshold	50
Min Line Length	20
Max Line Gap	20

3.Modification to draw_lines

There are two major modification made to draw_lines. Firstly, using the slopes, divide the lines found from Hough transform to the left lane line and right lane line. Then use the separated lines and find outliers and remove the outliers. Next, all the left lane lines and right lane lines are averaged to one line and draw this line on the image. This line is interpolated up to the extent of the mask we used in step 4.

4.Shortcomings

This algorithm will not work on curved lines, and any situation where the lines appear curved (such as going downhill, camera warping etc.). The edge detection is also not very robust and there is a lot of tradeoff between edge detection and noise.

5.Possible Improvements

Figure out warping caused by camera, optimized algorithm for curved path etc.