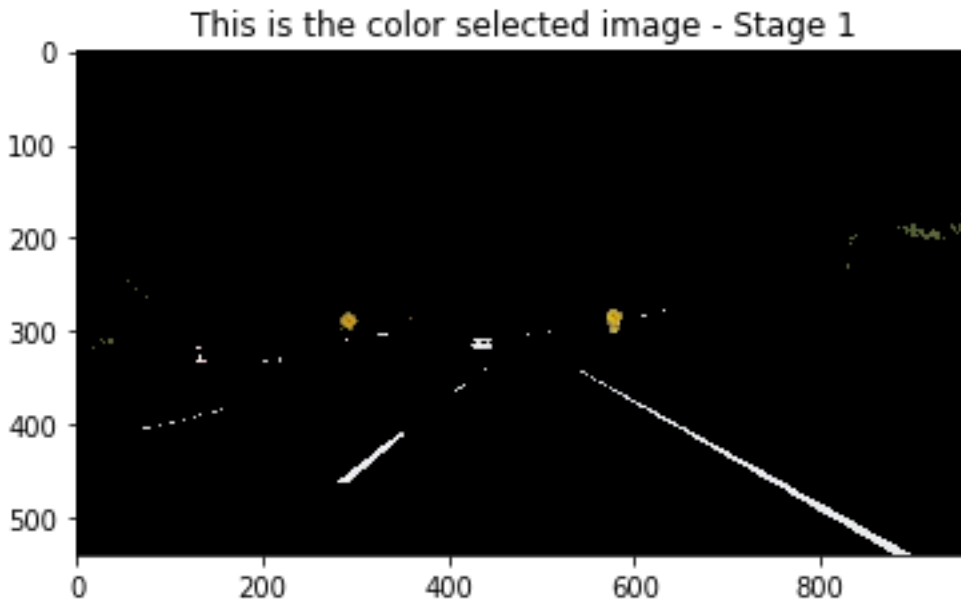


Reflections

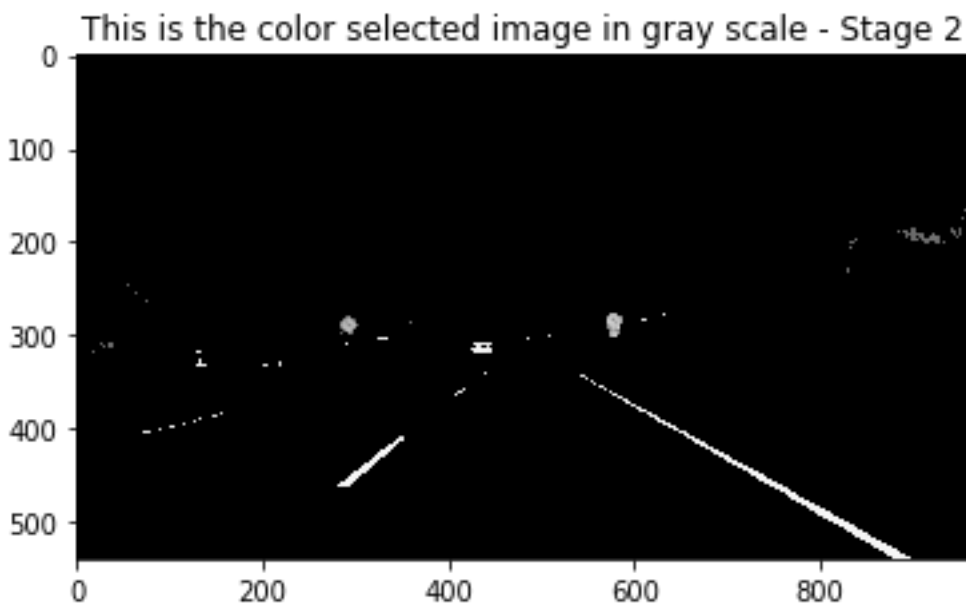
1. Description of the pipeline

An 8 stage pipeline has been implemented to achieve the final objective of lane detection.

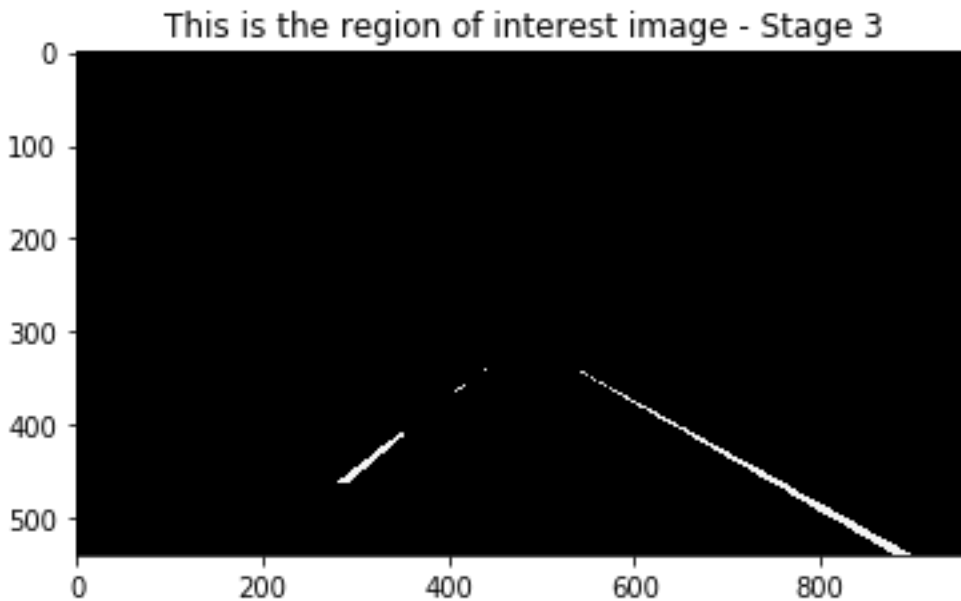
1) Stage-1 of the pipeline selects only the yellow and the white pixels from the incoming image. The rationale here is that most of the times, lanes are marked with either white or yellow.



2) Stage-2 of converts the image from RGB scale to Gray scale.



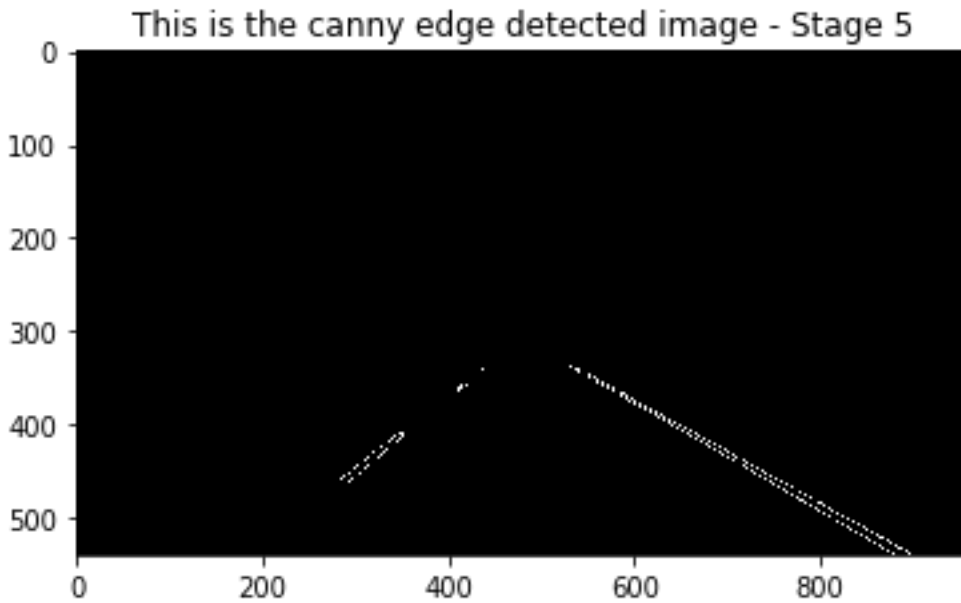
3) Stage-3 defines a mask in terms of a quadrilateral which is our region of interest i.e. this is the region where we expect the lanes to be found with a high probability. This mask is then applied to the gray scale image to obtain a masked gray scale image.



4) Stage-4 performs Gaussian filtering. This reduces the noise in the image. The kernel length should be selected carefully since if excessive filtering is applied, it will blur the image making it hard to detect edges.

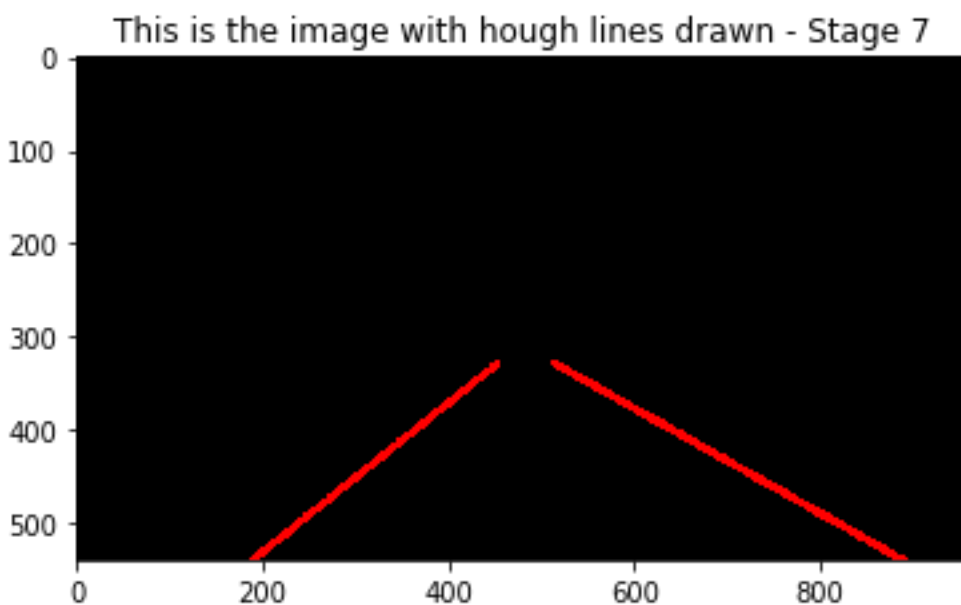


5) Stage-5 performs the Canny edge detection transform on the masked image to detect edges in the image.

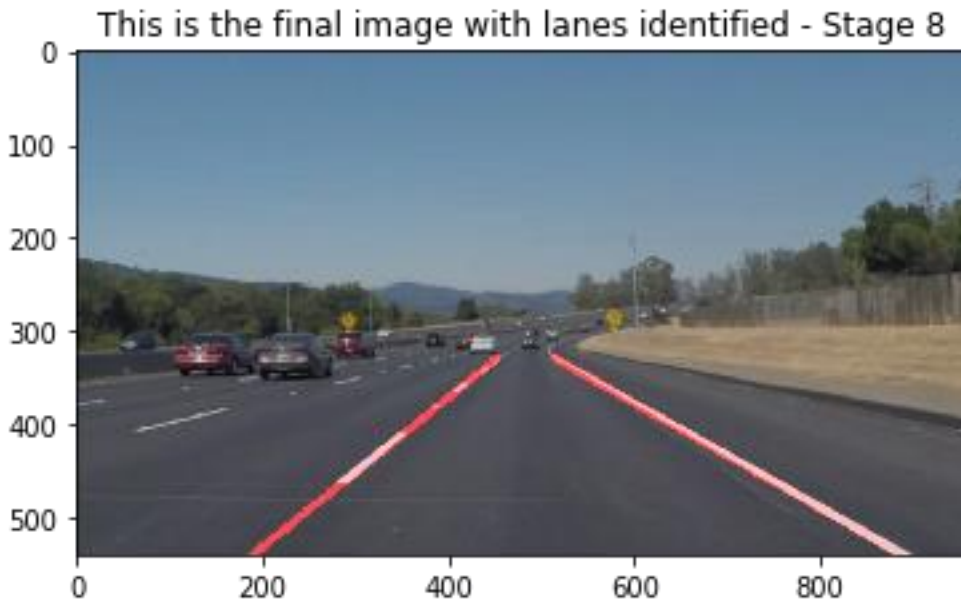


6) Stage-6 performs the Hough transform to detect all the lines in the region of interest.

7) Stage-7 replaces the smaller line segments found in step-6 and replaces them with one solid line on each of the sides. To achieve this, we classify all the points found by the Hough transform which constitute a line segment, either to the left lane or to the right lane. This is done by finding out the slope of each line segment and classifying them based on whether it is positive or negative. To eliminate the line segments which have positive slope by lie in the left lane and vice versa, we put an additional constraint of checking the relative position of the point with respect to middle of the image. Once all the points are classified, a linear regression is performed on the points to get parameters of the two lanes. Finally, the top and the bottom y-coordinate of the quadrilateral defined in step 3 is used to find out the corresponding x coordinates. These lines are drawn on a blank image then.



8) Stage-8 is the final stage of the pipeline. It combines the original image with the image obtained from stage-7 to get the final image which has the lanes identified.



2. Shortcomings

1) I found the pipeline to be extremely sensitive to the defined region of interest. It had to be hand tuned to ensure that lanes in all possible test scenarios can be covered. However, this approach cannot scale as the pipeline is applied to more and more test samples. The challenge video exposes this issue as well.

2) It is assumed that the lane makings are either in white or yellow. Although I believe this to be a fair assumption, however it is not generic enough to cover all scenarios.

3) Curves are not identified in this pipeline.

3. Improvements

There are two important directions in which this project can be improved.

1) From a software perspective: The current code is not exception safe. If any of the stages throw an exception it should be handled gracefully and proper actions must be defined.

2) From an algorithmic perspective: The pipeline can be further enhanced by using Hough transform for detecting curves as well.