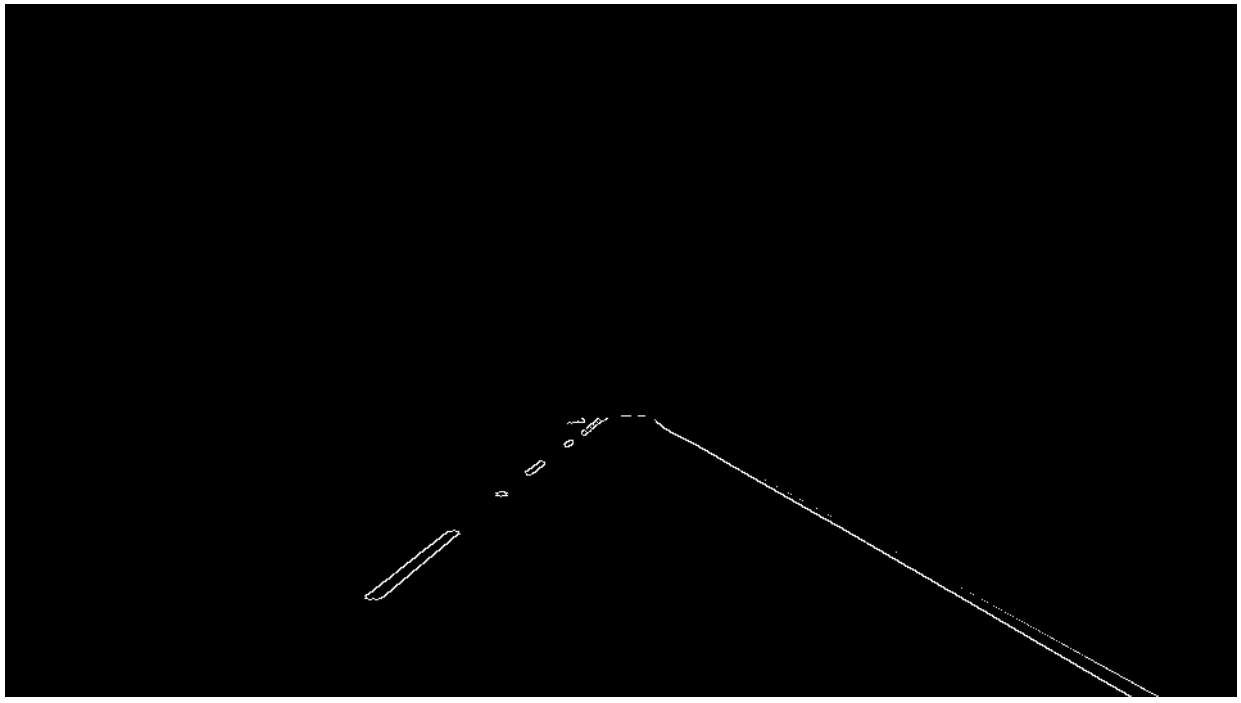
Finding Lane Lines on the Road

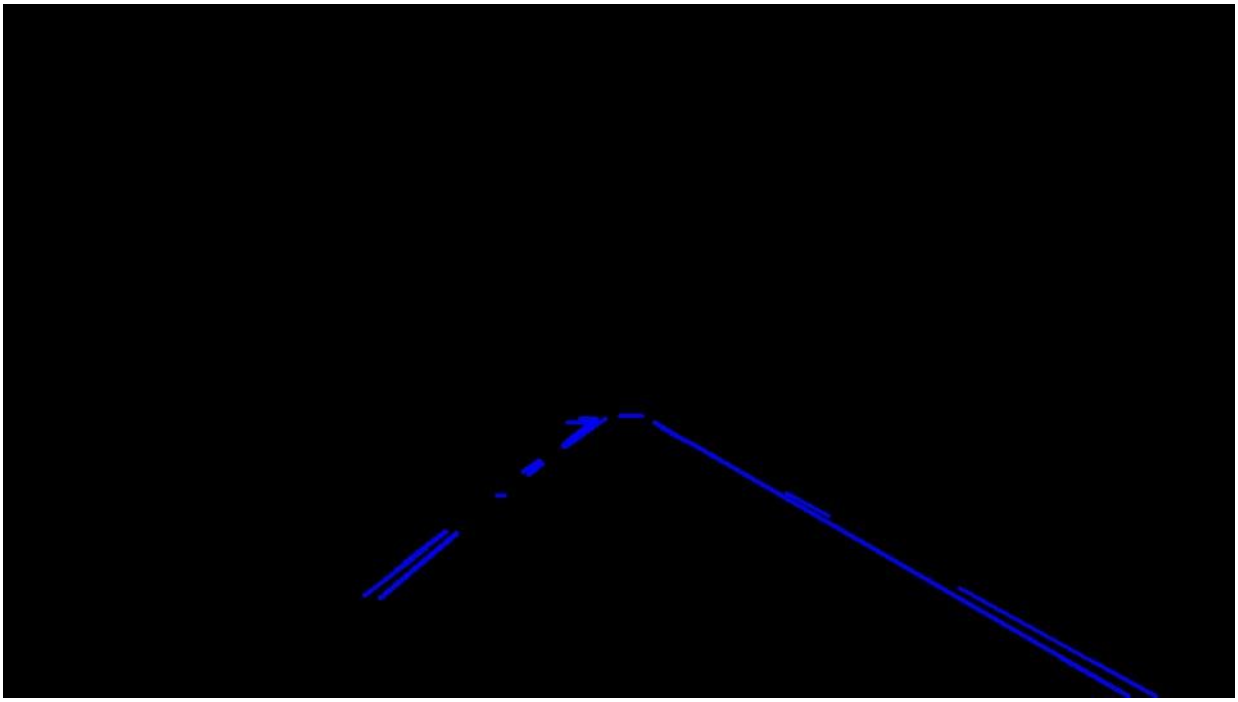
The goals / steps of this project are the following:
* Make a pipeline that finds lane lines on the road
* Reflect on your work in a written report

Reflection

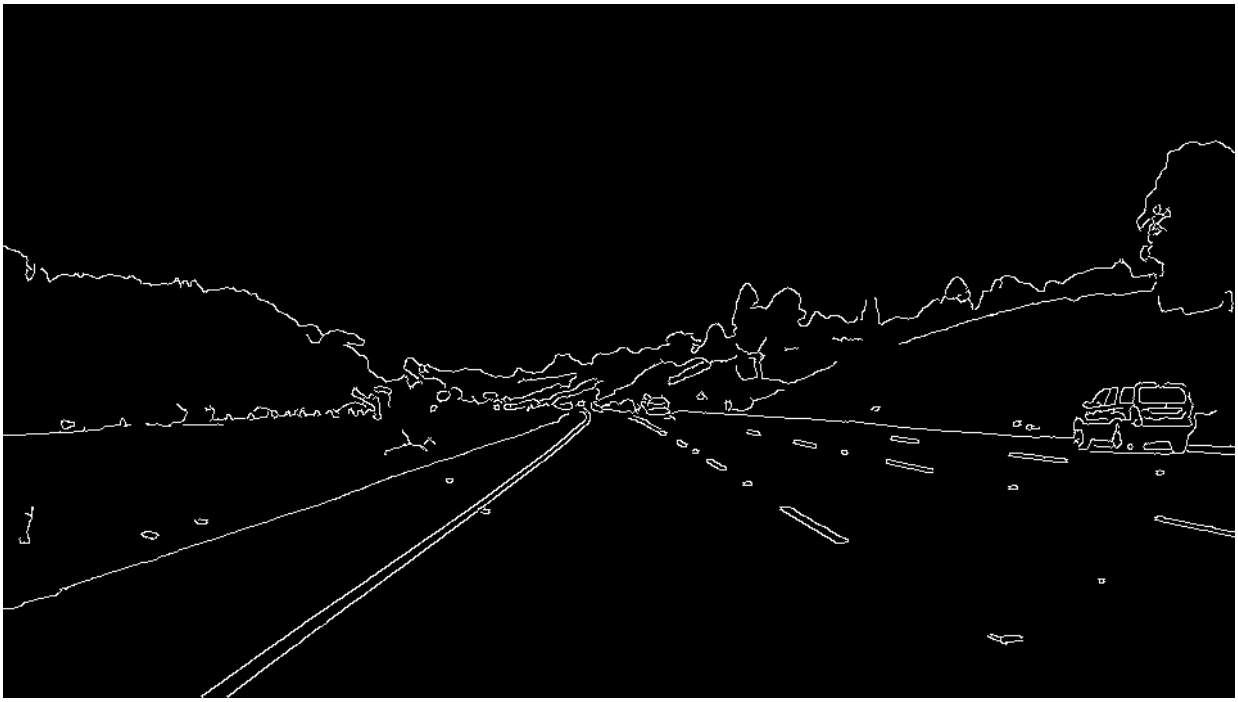
1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

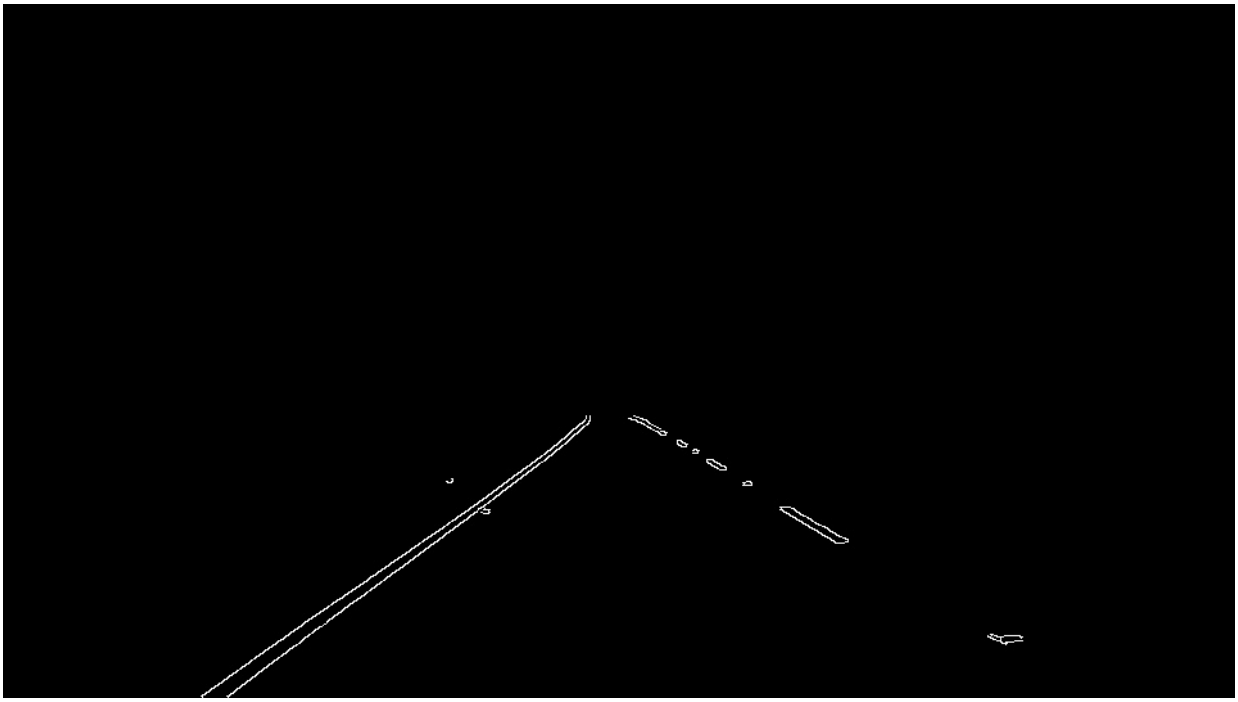
At first I called the helper functions in the right order with almost no modification, I got the following images: original image in gray, the Canny-Edge image, the ROI image, the Hough-transformed ROI Canny-Edge image and ultimately the original image with the lines drawn on it. See the used parameters in the notebook.

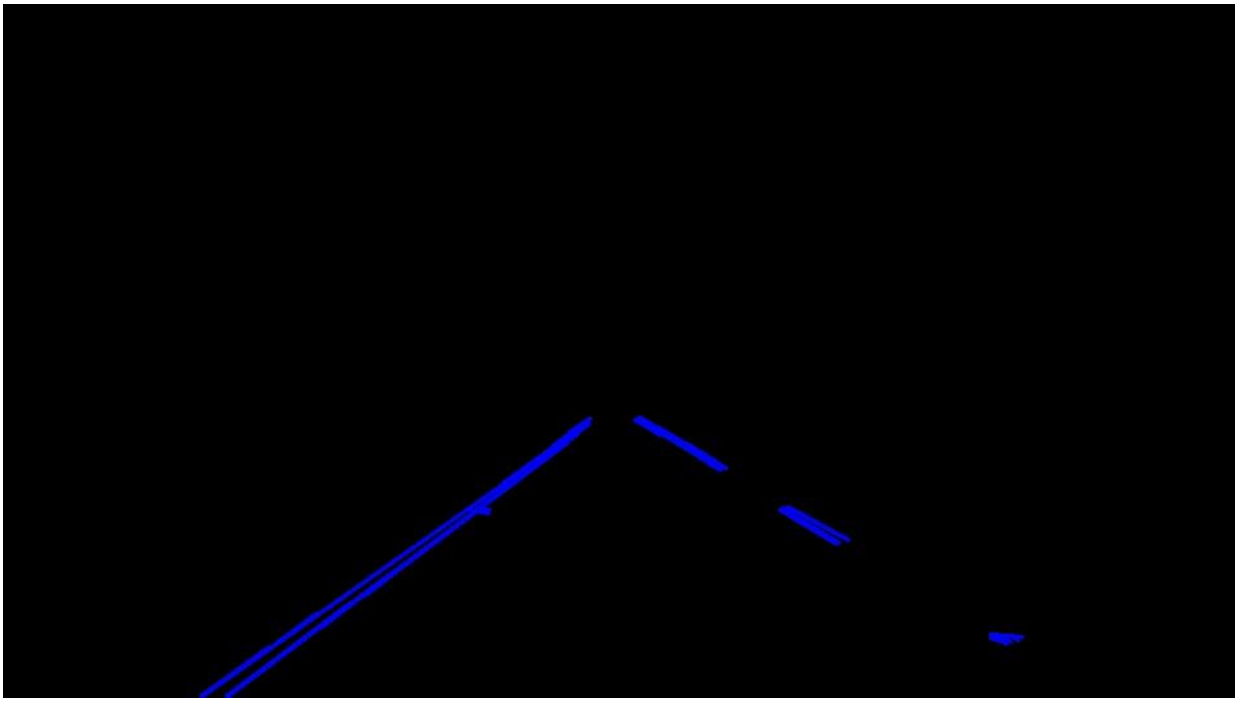




How about the same steps applying them on a different test image:







In order to draw a single line on the left and right lanes, I modified the draw_lines() function by using polyfit() function from the numpy library. It fits a polynomial on both the right and left lane candidate points respectively.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when the lane must be identified in a curve. As we see in the “challenge” video, the pipeline just fails big time when it comes to a curve.

Another shortcoming could be if there was a lane consists of short lane segments – e.g. dots, there might be an issue to detect any lane segments with looking for lines using the Hough transform.

Lighting as well as weather conditions might also influence the detection results in the negative way.

3. Suggest possible improvements to your pipeline

A possible improvement would be to take tracking into consideration

Another potential improvement could be to keep improving the current pipeline parameters.

Hard coded ROI is probably a really beginner way to implement lane detection, there should be other ways to figure out where to focus on the road.