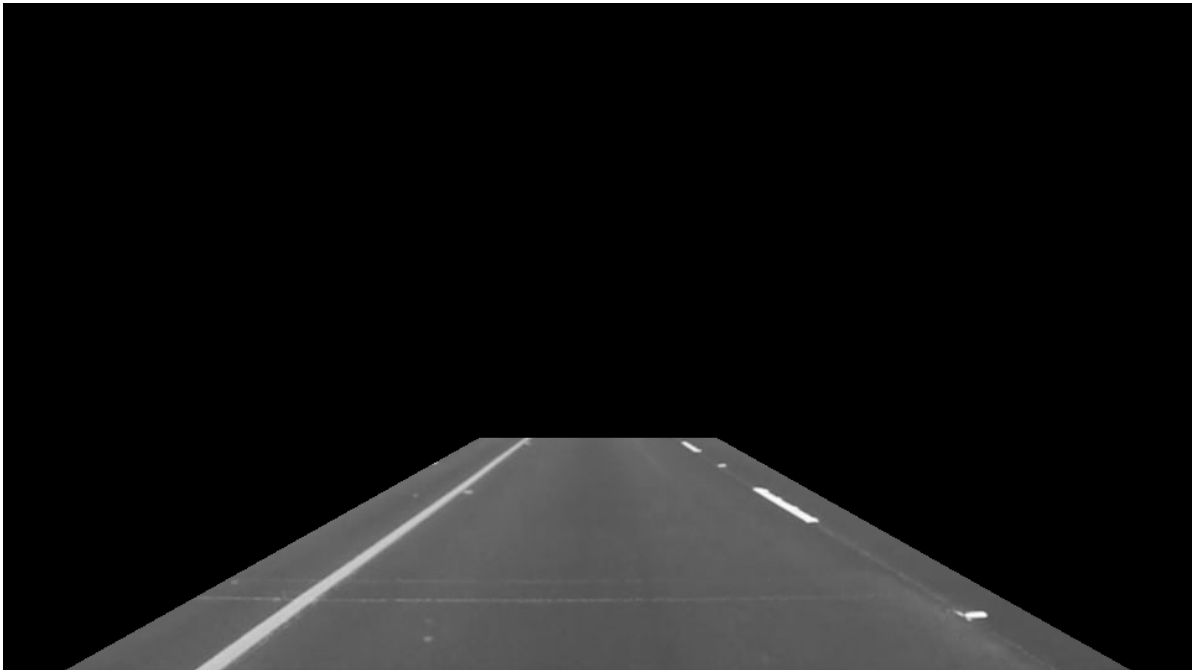


Finding Lane Lines on the Road

Pipeline

My pipeline consisted of the following steps:

- Convert to greyscale
- Apply a size 3 Gaussian blur to remove noise
- Apply a canny edge detector with low threshold of 50 and high threshold of 150
- Apply a mask filter to isolate a region of interest
 - The mask filter restricts further processing to a quadrilateral area in the bottom middle of the image with vertices:
 - $(\text{imgshape}[1] * 0.4, \text{imgshape}[0] * 0.65)$
 - $(\text{imgshape}[1] * 0.6, \text{imgshape}[0] * 0.65)$
 - $(\text{imgshape}[1] * 0.95, \text{imgshape}[0])$
 - $(\text{imgshape}[1] * 0.05, \text{imgshape}[0])$



- Apply a hough transform on the masked image with the following parameters:
 - $\rho = 2$
 - $\theta = \text{np.pi} / 180$
 - $\text{threshold} = 10$
 - $\text{min_line_len} = 25$
 - $\text{max_line_gap} = 15$

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by first grouping the lines outputted by the hough transform to those with positive slope and those with negative slope.

Given that we are looking for lane lines, we reject those with absolute slope greater than 2 or less than 0.5 (Assuming the car is driving in the middle of the lane, the lane line is expected to not have a very steep or shallow gradient). This is done to remove outliers.

Within these groupings, the endpoints of each line is extracted and a line is fitted to these points (each point is weighted by the length of the line. The assumption here is that we give more weight to longer lines as they should contain more information). The line generated from this modelling is then extended to the bottom of the image and 35% up from the bottom.

Shortcomings

One potential shortcoming is that the extended lane line is hardcoded to always start from the bottom and ends 35% up from the bottom. Thus it is not a good representation of the lines we actually detect.

A second potential shortcoming is that the region of interest makes some restrictive assumptions about where the lane lines should be. Depending on the height of the camera or placement in the car (may not be in the centre of the car) or the type of road we are on (perhaps lanes have different spacing in a different country), our assumption may be violated

Improvements

A possible improvement would be to contrast adjust the image. The pipeline struggles with the different brightness in the challenge video.

Another potential improvement could be to remember what lane lines we detected in the previous frame and use that to detect lane lines in the current frame. This would improve lane line detection in the challenge video where in some cases there are no lane lines, so an "implied" lane line would be an improvement. This would also make detection more consistent and robust.

Another improvement is to use polynomial fitting to detect curved lane lines (for example for a corner). Currently we are only able to detect lines.