# Finding Lane Lines on the Road

The goal of this project is to build a pipeline to find lane lines in the road. Below is a brief description of the work I did to build this pipeline.

The image is converted to grayscale and a Gaussian kernel is run over the image. I chose kernel size of 5 for this project.

The blur image is then run over canny edge detection algorithm. This algorithm returns an image with edges detected. I chose 60:100 low_threshold:high_threshold ratio.

This image is then used to extract our region of interest i.e we extract the driving lane region from the image.

Once we extract our region of interest it's time for us to connect the dots and from lines. This image is fed to a Hough algorithm which draws lines for us.The following parameters were used to pass to HoughLinesP API

rho = 1 # distance resolution in pixels of the Hough grid

theta = np.pi/180 # angular resolution in radians of the Hough grid

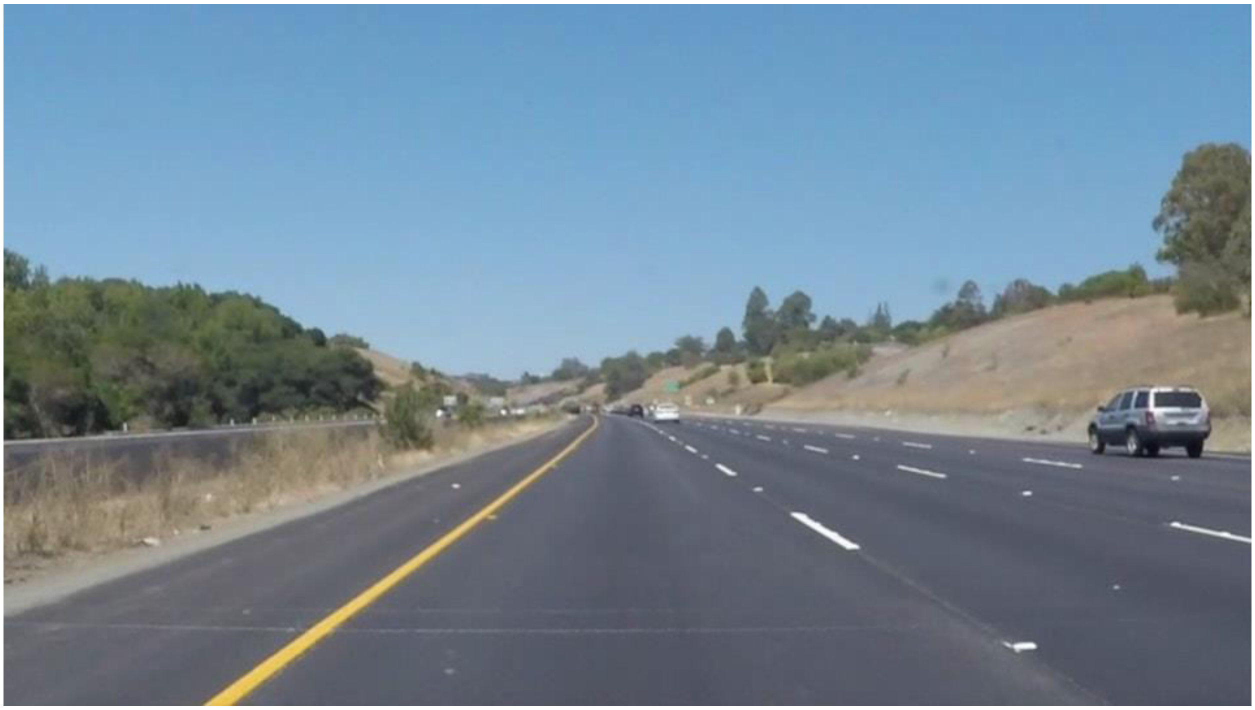
threshold = 6     # minimum number of votes (intersections in Hough grid cell)

min_line_length = 7 #minimum number of pixels making up a line

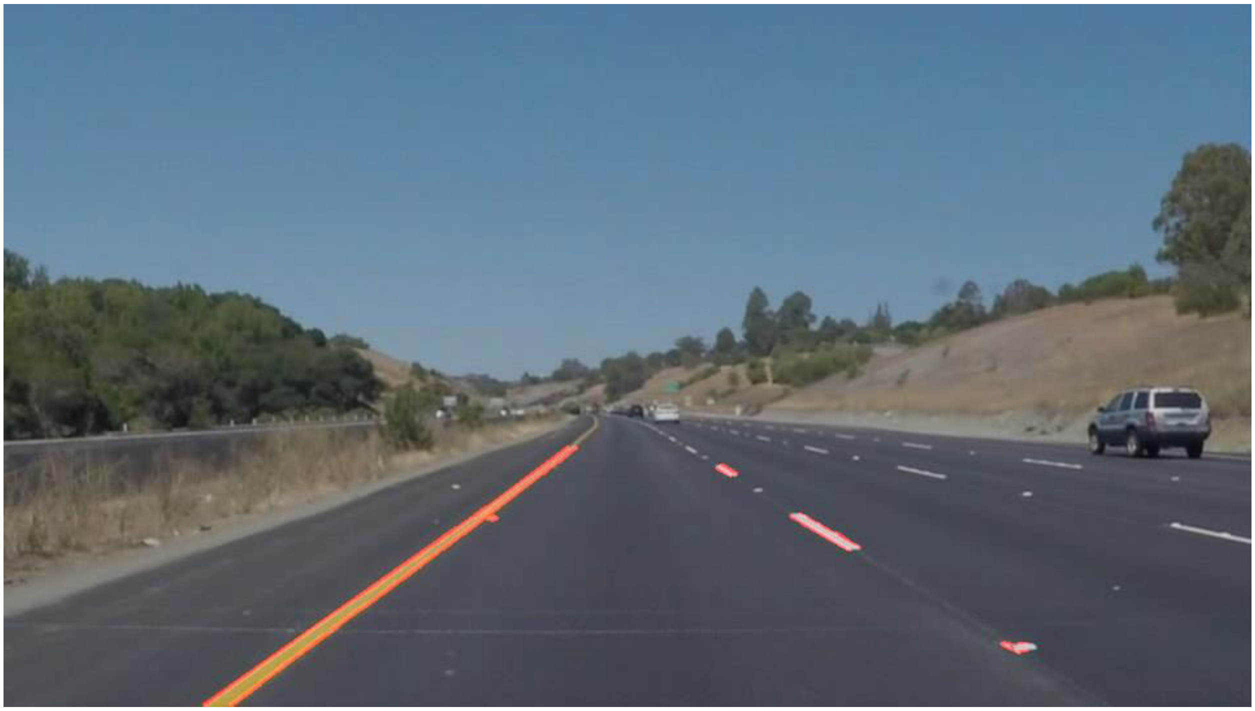
max_line_gap = 3  # maximum gap in pixels between connectable line segments

The lines returned by the hough algorithm are then drawn on a zeroed image.This image is weighted with our initial image(unprocessed image).

So far we have image transformed as shown below:



Initial Image



Processed image

Now, this processing pipeline is applied to a video clip. Since video clip is a series of images, the pipeline works well.

The interesting part lies in extrapolating the lane lines as shown in the videos. The draw_lines function is modified so as to be able  to draw continuous lines on both the sides of the lane. To do this left and right lane are classified based on their slopes. Mean line is calculated over all the lines and used as reference vector to extrapolate. The bottom point is extrapolated freely however care had to be taken when extrapolating the top point. The y axis point is restricted and based on that the x axis is calculated since we know the slope and thus this fixes our coordinates for each line.

There are shortcomings with this pipeline, particularly draw_lines function, which might not be able to correctly draw lanes on sharp curves.

Another shortcoming could be : if on a dotted side of the lane(i.e the lane without solid line) if there appear horizontal long edges or markings, there are chances that the average vector will lie perpendicular to the road.

More robustness needs to be added to address the above issues.