

## Reflection

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

My pipeline consisted of 5 steps.

First, I converted the images to grayscale

Second, I applied canny edge detection after removing noise and spurious gradient using `gaussianblur`. Canny Edge Detection will detect strong gradient pixels above high threshold of 150 and then pixels with values between low threshold = 50 and high threshold = 150.

Third, canny edges image is cropped as per region of interest.

Fourth, Hough transform for lines will determine connected lines present in the canny edges. Hough transform will consider those lines that has minimum 35 votes in the hough grid,  $\rho$  and  $\theta$  resolution is kept as 1 and  $\pi/180$  respectively. Hough transform would consider a line for votes that has at least 5 pixels in a line segment and a max line gap of 2 pixels between segments to form a single line.

Fifth, Lines determined from hough transform is applied to line function of `cv2` to draw all the identified lines in step 4.

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function:

Left lane line and right lane line gradient and coordinates are calculated by taking average of slopes and average of centre coordinates of left lines and right lines respectively. Lane lines are extrapolated by determining two  $[x,y]$  coordinates for both left lane and right lane. For this, set of  $x$  is calculated by putting  $y$  values as 540 (bottom of image) and 313 (height of image) in the equation of line determined by using average slope and average point on line. Lane lines are drawn using line function of `cv2` that takes two set of coordinates as determined in this step.

Pipeline images:





## 2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be lane lines inaccurately mapped to actual lanes when average gradient of left lines and right lines belonging to a particular image becomes either close to infinity or close to zero. Curvy lanes may not be well fitted using single degree line equation

Second, at these extreme gradients, lane line equation drawn with y coordinates as bottom of image(540) and height of image(313) may differ drastically from the lane line equation belonging to immediate next frame where lane lines gradient is not that extreme. This would cause inaccurate mapping of actual lanes. This would happen on curvy roads.

## 3. Suggest possible improvements to your pipeline

A possible improvement would be to consider extrapolating to degree 2 polynomial to cover curvy lane cases.