# Finding Lane Lines on the Road

---

**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. Pipeline used in the code consists of 6 steps:

A. Step 1: Convert image to grey scale.
B. Step 2: Add Gaussian blur to resultant image of step 1.
C. Step 3: Apply Canny edge detection algorithm on resultant image of step 2.
D. Step 4: Apply 4 sided mask to resultant image of step 3 to filter out unwanted edges.
E. Step 5: Apply Hough transform on resultant image of step 4.
F. Step 6: Draw the lines on resultant image.

**Procedure used for extrapolation:**

A. We know that lane lines cannot be horizontal or vertical in image. So, I filtered out the edges/lines which were following in this category from Hough transform output in “draw_lines” function.
e.g. This is the screenshot of “solidYellowLeft.mp4” around 00:11 sec. We can see that there is a horizontal line on top of yellow line.

B. Then I used least square method to fit a line passing through resultant points.

$$A^T Ax = A^T y$$

## 2. Identify potential shortcomings with your current pipeline

A. Method might fail if some vehicle is very near to our vehicle. (E.g. In case of heavy traffic)
B. Method might give (2 wheeler)biker as edge.
C. If camera is shifted by some distance on car, few parameters in this method needs to be changed. Method is not shift invariant.

## 3. Suggest possible improvements to your pipeline

A. More data points(images/videos) is needed to test this pipeline. By doing this, if some corner case is missing, we can take that into account and that will in turn improve the pipeline.
B. Need to consider “challenge.mp4” (optional challenge) case.
C. This pipeline does not consider the case in which lane lines are not present for some distance. Need to test behaviour in such scenario.
D. Cases like, “if some vehicle crosses the adjacent lane lines” or “some junction comes” also needs to be tested.
E. Automatic algorithmic mask calibration will improve this pipeline.