

Pipeline:

Pipeline for detecting lanes consists of following steps:

1. Image is first converted to grayscale
2. Gaussian filter is applied to smoothen the image.
3. Detect edges using canny edge detection algorithm
4. Mask all the image that lie outside the trapezoidal region of interest. Output of canny edge detection algorithm is given to region_of_interest function to do this.
5. Then lines are detected using hough transform.
6. Then we overlay the lines on the original image.
7. In order to extrapolate the lines, in draw_lines function we separate out the lines based on positive or negative slope into positive line and negative lines. For each of those group, we iterate over all the lines and take average of the x_1, y_1 and the slope. x_1, y_1 acts as the initial point of the line. We choose x_2 as the top end of the trapezoid (region of interest). Using $y = mx + c$ we calculate the y coordinate. Then using opencv's line function the line is extrapolated.

Shortcomings:

1. Region of interest are fixed for each image. So if lanes move to some far away portion of the image, because of the mask, it won't get detected.
2. The x_2 (top x coordinate of the extrapolated line) is assumed based on the corner of the trapezoid.

Possible improvements:

Slope of the lines in current image in a video can be derived based on the average of slope of the lanes seen in previous image using some buffer frames and the current image. This way lanes detection would be more robust.