# Lane Detector Using Canny Edge Detection and Hough Transform

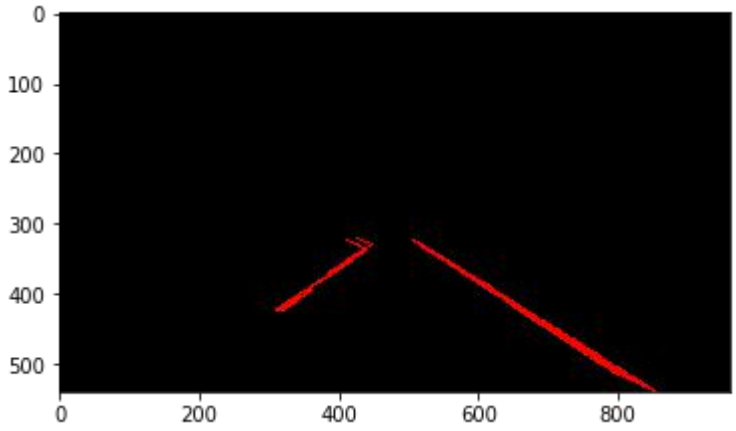
Please refer to the code for implementation details

**Pipeline:**

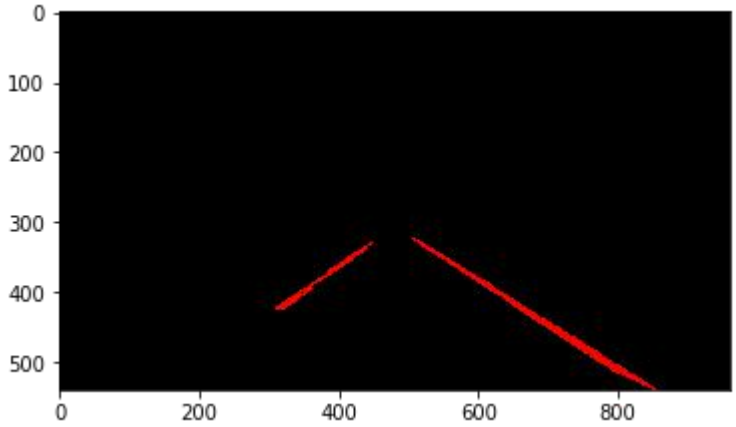
1. Convert to Gray scale
2. Apply Gaussian Blur
3. Find edges using Canny Edge detector
4. Define a Region of Interest (ROI) based on the vertices.
5. Keep only the edges that lie in the ROI
6. Apply Hough Transform and find line segments
7. From the found line segments only keep the line segments following these steps as shown below:
    a) Divide the image into two halves, i.e., left half and right half.
    b) All the line segments in the left half should have negative slope and all the line segments in the right half should have positive slope. We call this as **slope test**.
    c) So based on the above slope test, filter the line segments and keep only the lines that follow the above criterion. A simple illustration of the slope test is give in the next section.
    d) From the remaining line segments, find the tips or the end points, i.e., what is the lowest and the highest (x,y) coordinates in each halves.
    e) Based on these tips we find the line equation in each half.
    f) As the line segments do not span the whole space or they can be broken, so from the found line equation, we estimate (x1,y1) and (x2,y2) that make a fully connected line. In this (x1,y1) and (x2,y2), we know y-coordinated based on the length of the line we want to have, and we are estimating the x-coordinates from the line equation.
    g) Draw these lines on the image on the image.
8. Weigh and add images to show annotated lines in the video.

**A simple illustration of the slope test**

**Input Image:**




**Output Image:**




**Shortcomings of the proposed Lane Detector:**

1. As we only consider the tips of line segments that pass ROI and the slope test, there are some outlier line segments that pop up.
2. Specifically in the *solidYellowLeft.mp4,* we can see that at a couple of instances, the lane detector jumps and the reason is because, there are some line segments that pass both the ROI based masking and the slope test. They occurred on the bottom left where small markers are present. A simplest solution for this would be a very tight ROI.
3. Next, the challenge video highlights other failure cases where the lane detector fails with shadows and change in the color of the road.

**Proposed enhancements:**

1. To remove the first and the second shortcomings, we can use RANSAC, and find the line equation that satisfies most of the line segments. In this case, there still is an assumption that the proposed lane detector finds more number of true positives than the false positives.
2. To cater for the change in road color and shadows, one thought is to use HSV color space and a sample video output of finding the left lane after applying the HSV color space technique is present in the videos.
3. My initial experiments have shown that proper ROI selection with the proposed method detects right lane very well in the *challenge.mp4* in the RGB color space. So, another enhancement would be to make the lane detector work simultaneously on both color spaces, HSV and RGB and then append the detected lines (sometimes lane detector from one space may not detect any line segments), apply RANSAC to find the lane that satisfies most of the detected lanes.

Thank you!