Udacity : Self Driving Car Nanodegree Apr2018

Lanes Detection – Project 1
Prepared by: Sanyam Agarwal
April 15, 2018

Udacity : Self Driving Car Nanodegree Apr2018

Objective:

- ❖ Basic understanding on lane detection and its concepts.
- ❖ Make pipeline and find lane lines on the road, both on images and videos.
- ❖ Identify solid and dashed lane lines, combine and make it to one straight line
- ❖ Used Opencv api to identifying lane lines

Description:

Multiple image processing algorithm is applied at multiple stages. Each stage in the description represents the desired pipeline mentioned below:

■ Stage 1:

This stage involves converting the RGB image to Greyscale image. This step is very important to identify the lane lines, as in greysale we can have either white, grey or black color which makes it easier to find or detect lane lines. Below is the Greyscale output image

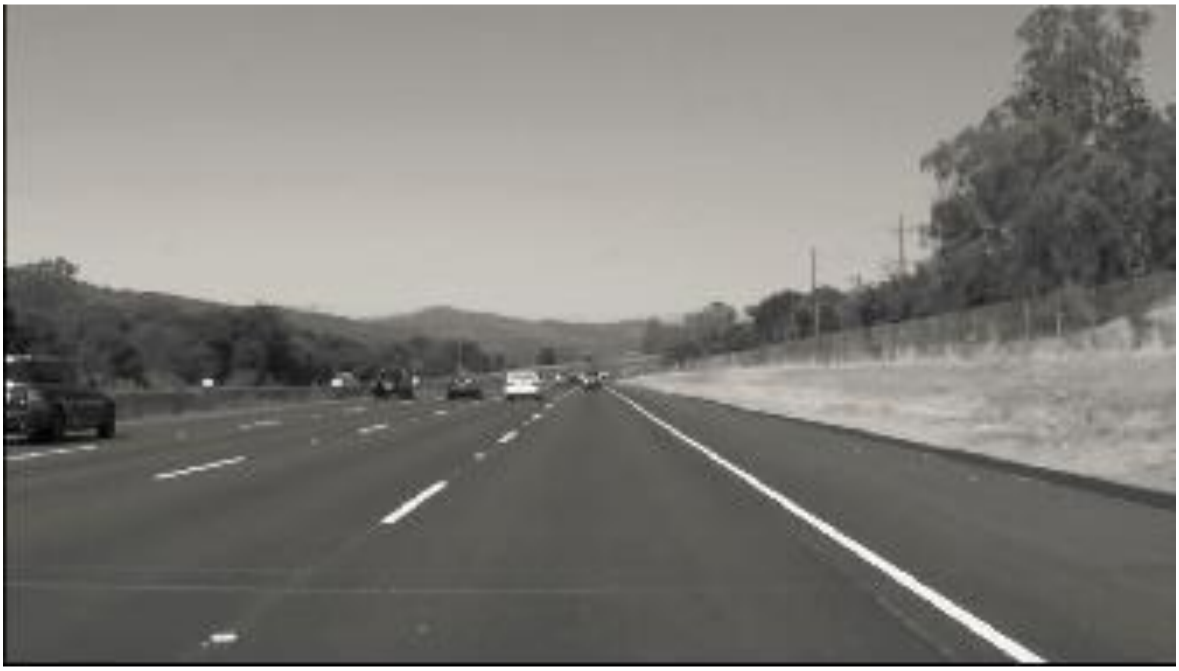


Fig 1: Greyscale Image

■ Stage 2

This stage performs Gaussian Blur. Main intention of performing this is to reduce the image noise.

Stage 3

This stage performs Canny edge detection. The goal is to identify the boundaries of an object in an image.We pass the low and high thresholds for pixels to be filtered and get the black image with boundaries traced out. Below is the image output

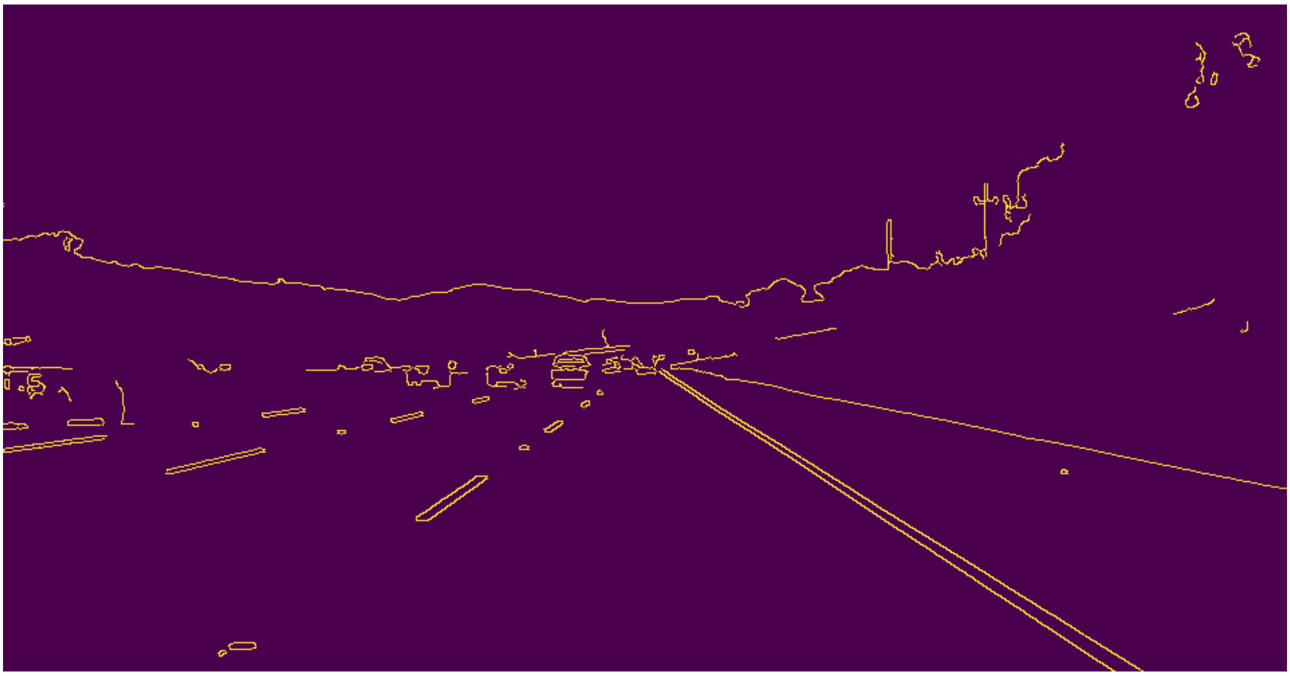


Fig 2: Canny Edge Output

Stage 4

This stage performs region of interest (i.e. region masking) by defining polygon boundaries,and then draw only the required Canny edges on an empty same size file, then just mask the resulted file with the original image to draw the detected lines.

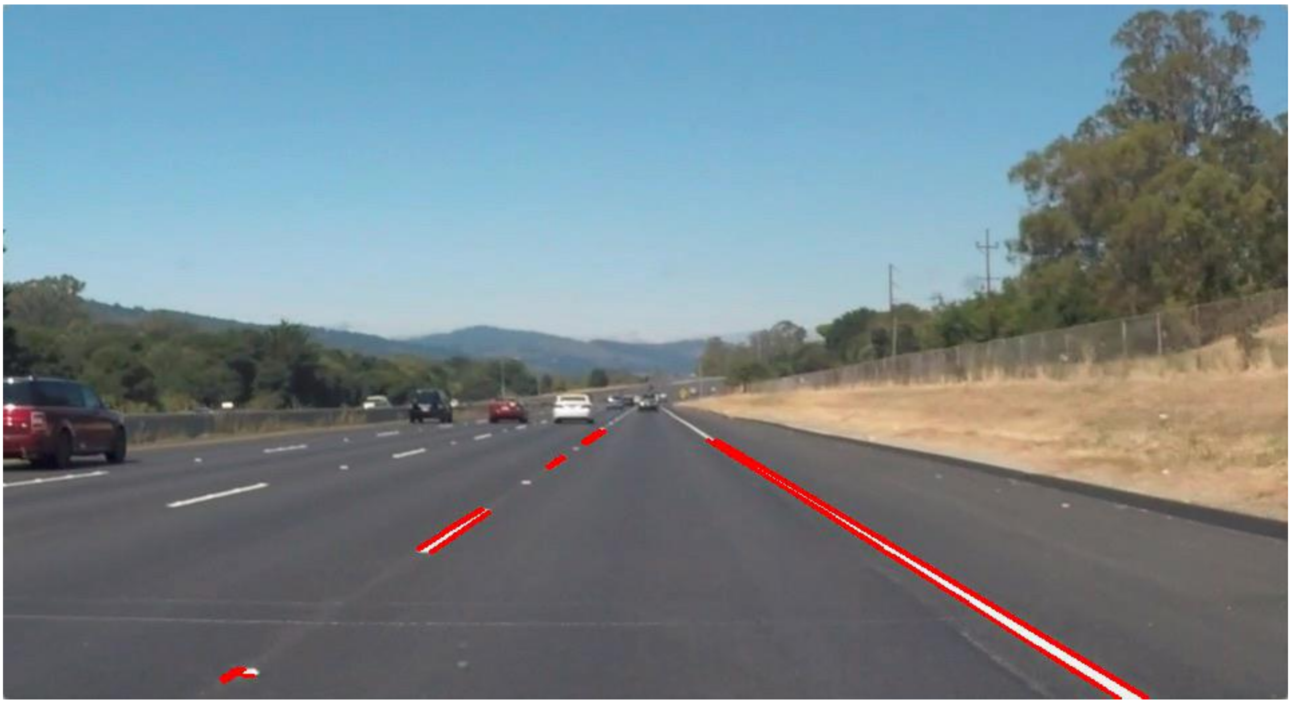
Stage 5

At this stage we apply Hough transform on the Stage 4 resultant image, that could make a line with the given boundary which are passed as parameter to Hough openCV api.These parameters control the accuracy by applying max_length, threshold, and min length

Udacity : Self Driving Car Nanodegree Apr2018

Challenges Involved:

After stage 5 we get the output mentioned below, where we detected lines as segments



For drawing line we need top and bottom coordinates of left and right lanes.

We know that slope = Change in height / Change in run

(We know Negative slope is Left and Positive slope is Right)

We apply the above formula to all the points which were the output of Hough transform and try to find slope or all the points, we also discard the points which are of very steep slope.
And also locate center by:
Center = (x1 + x2) / (y1 + y2)

So now using line equation:

(x1 - x2) = m (y1 - y2)

We average all the points to get single top and bottom left and single top and bottom right, draw line accordingly, and apply the same steps when it runs on video as well, since video is collection of images.

Udacity : Self Driving Car Nanodegree Apr2018

Shortcomings :

- One major shortcoming\ drawback is , it currently doesn’t cover all practical scenarios like bad weather(snow, foggy, rain) , night, low light.

- We have not used object classifier to validate lanes are correctly detected. Many false objects can also be treated as lanes if it matches with the lane property (I.e. Near to Traffic lights multiple arrow direction are marked with white color on road, Sometimes On highways some weared character’s or some notation is marked with white color in between lane lines).

- Current algorithm does not work well on curves

Improvements:

- Make this pipeline more accurate and working with curves, bad weather(snow, foggy, rain) , night, low light.
- Add Object classifier to remove false object detection.