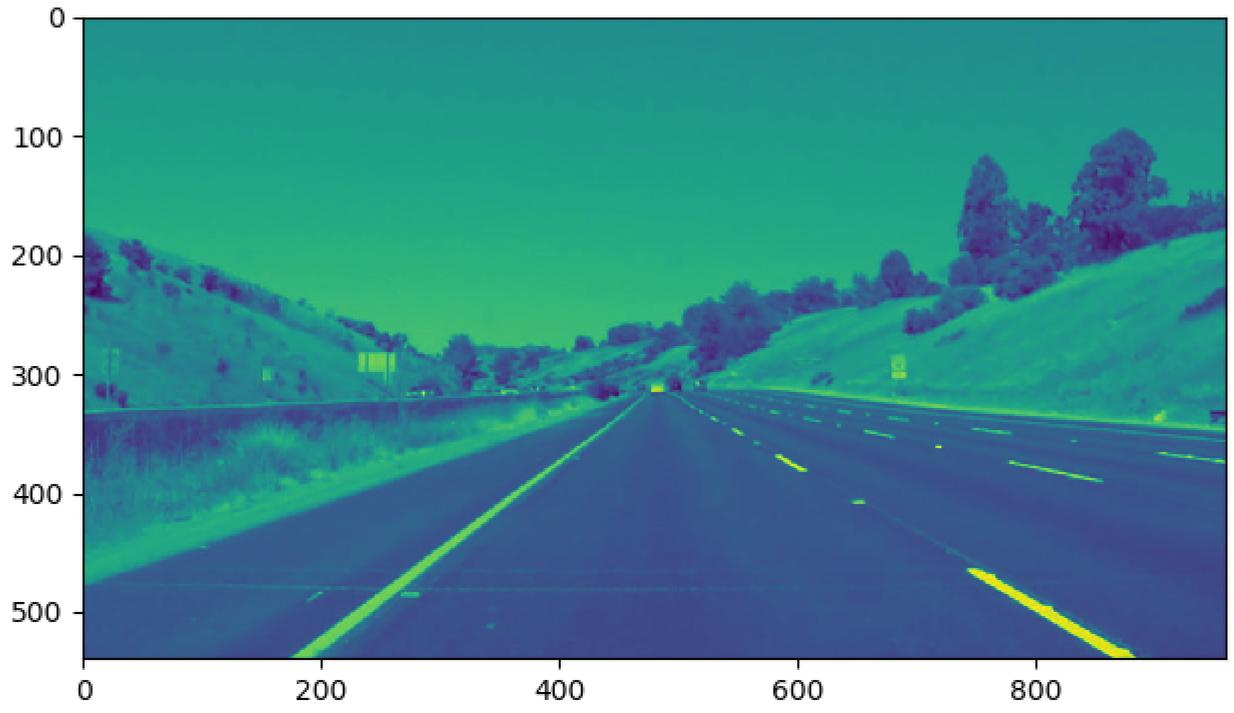
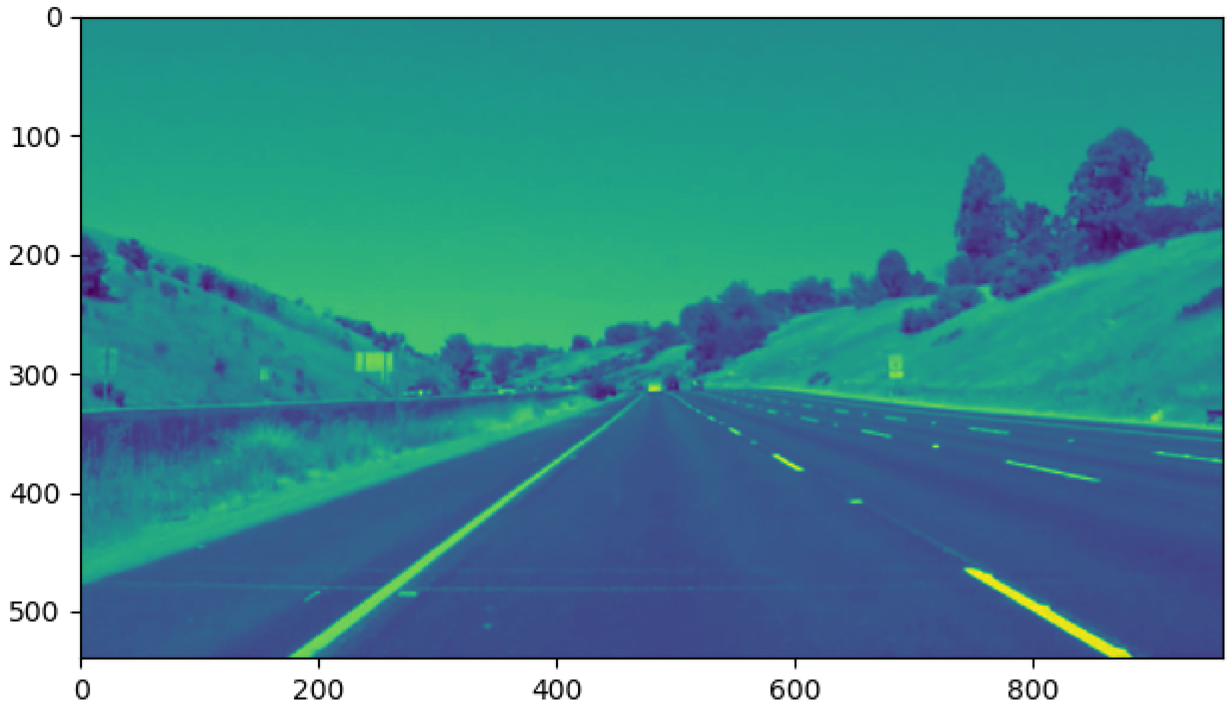
Lane detection pipeline

The main file to run is lane_lines_main.py which instantiates the Image_Proc.py class that contains all the helper functions and other supporting methods.
1. Finding lines
The overall helper functions and lane finding functions are implemented in the Image_Proc class to facilitate reuse and extensibility as per the object-oriented principles.

1. At first, the image is converted into grayscale as shown below:

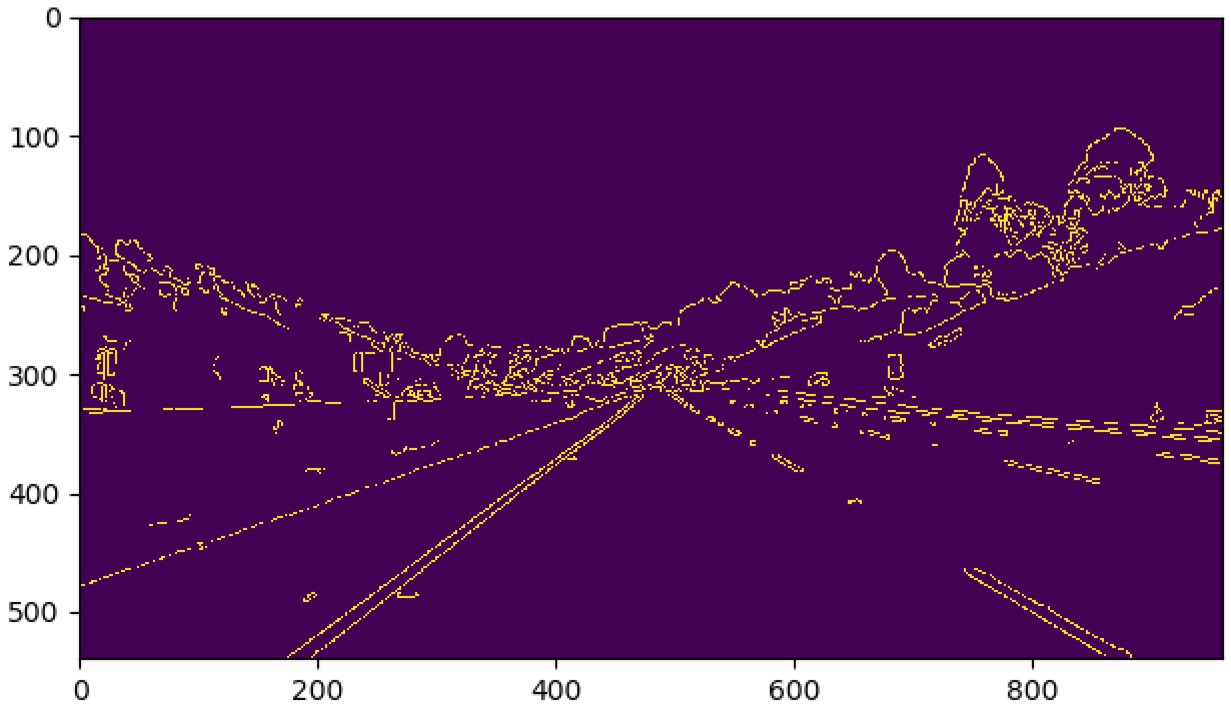


2. A Gaussian smoothing filter is applied to blur the image, remove detail and noise. The resultant image is given below:

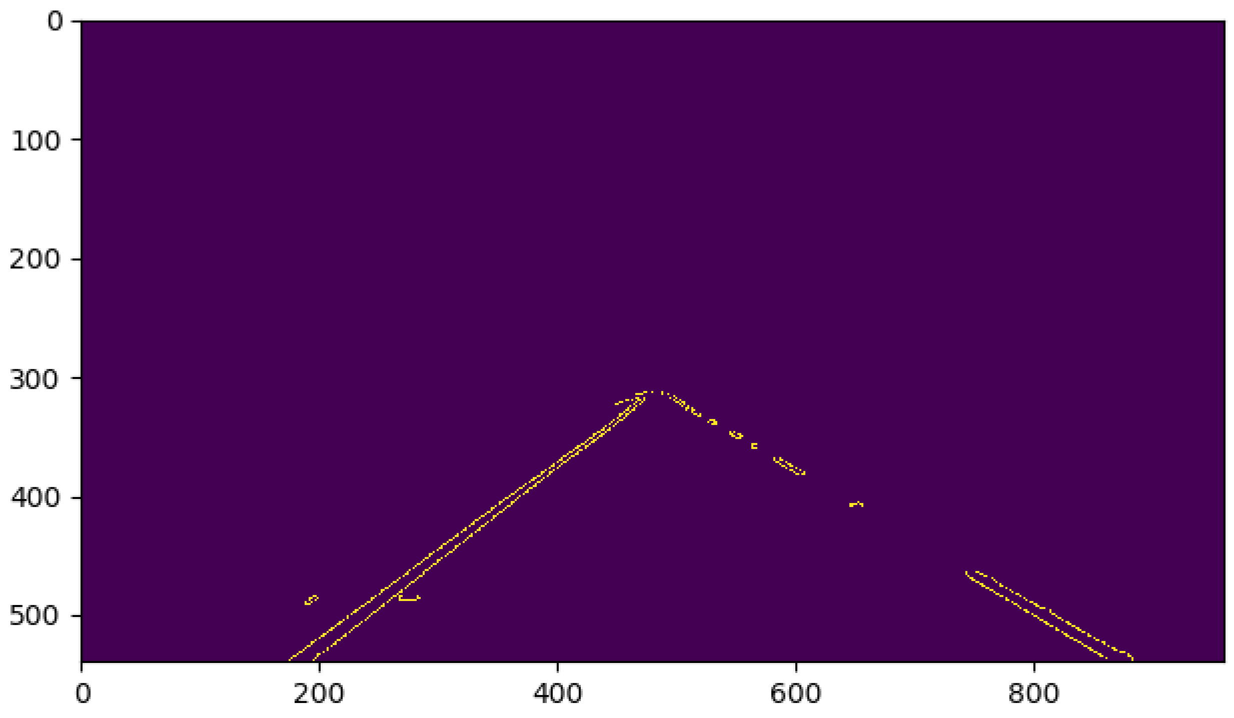


3. A Canny image kernel is used to then extract edge information from the image and further eliminate irrelevant details:

The low and high thresholds for the Canny operator were automatically calculated via the OTSU thresholding (otsu_canny) method. The method performs clustering-based image segmentation which is particularly usedful in images with deep and sharp valleys between two class peaks.



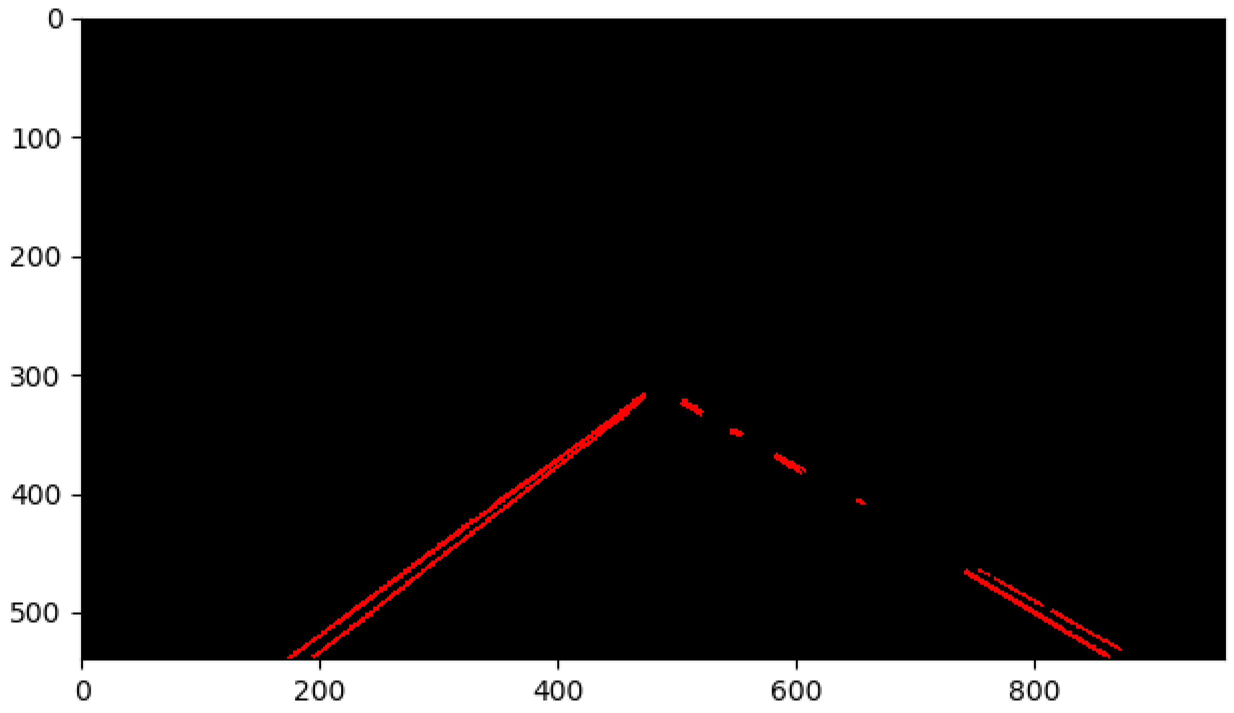
4. Prepare mask: The mask was prepared based on the following vertices:
(0.51, 0.58), (0.49, 0.58), (0, 0.58) and (original width and height)
The above generated the following polynomial coordinates for the 960 x 540 dimension image:
Printing polynomial values: 489.6, 313.2 470.4, 313.2 0, 313.2 960, 540
The vertices obtained in the previous step are then used with the OpenCV fillPoly with the provided helper function and returning the original image with respect to the mask



5. Lines: The image matrix thus obtain is then used to calculate the hough lines with the cv2.HoughLinesP helper function with the following parameters:

- Rho: 1
- Theta: 0.017453292519943295
- Threshold: 35
- Min_line_length: 5
- Max_line_gap: 2

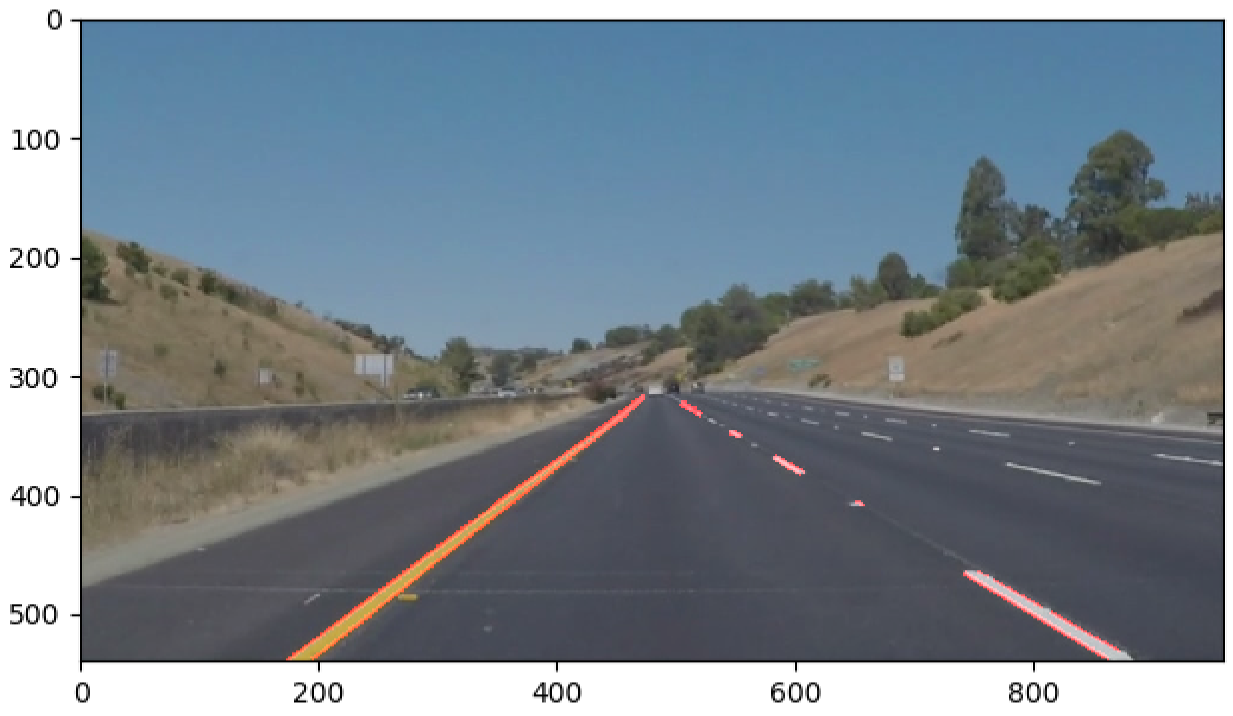
The output on the image mask is given below.



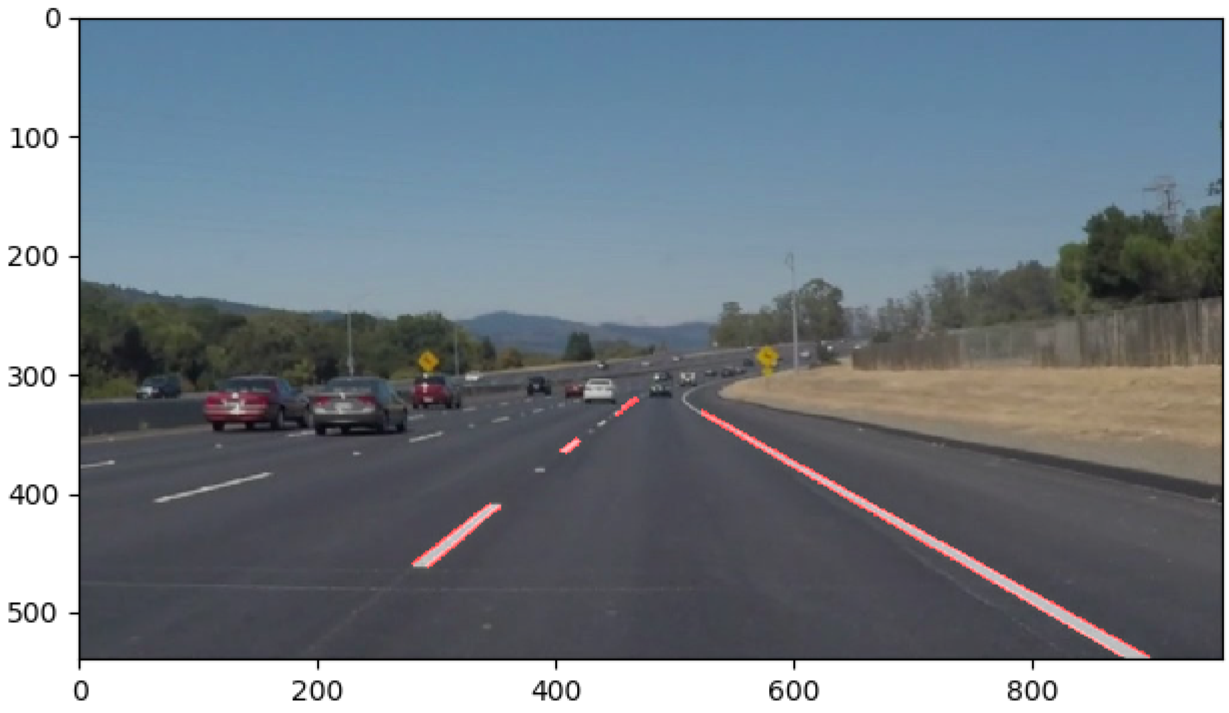
6. Weighted image: The weighted image (hough line mask) on the original image frame was then calculated as follows:

- alpha: 0.8
- beta: 1.
- lambda: 0

The outcome for a single left lane line frame case is given below:



The outcome for a single right lane line frame case is given below:



Evaluation of the two left and right handside lanes:
The outcomes of both left and right hand lanes are saved under the test_videos_output folder.

Evaluation of the challenge video:
There were issues as a lot of smaller line segments that ran horizontal appeared, especially at the car’s dashboard and sideways. One possibility to remove these is to use a HSV based inRange color selector to remove non-compliant (non-lane) pixels completely. Other is to used a different edge detection kernel (e.g. Sobel) and eliminate horizontal edges from the final frames.