

|  |
| --- |
| IIOT LAB – LANE FINDING REPORT |
|  |
| Done by: Muhammad Emad 200107010 6th Semester  muhammad.emad@dxb.manipal.edu |

INTRODUCTION :-

In this report, I will be going over how to make a pipeline that finds lane lines on the road by applying OpenCV functions to detect lanes in an image and later a video by following a pipeline. The following steps will be done in order to obtain the result : Greyscale imaging, Noise filtering, Smoothing, Region masking, Canny edge, Hough transformation, Video processing frame-by-frame and Saving the video as a mp4.

PROCEDURE :-

A basic Lane Detection pipeline can be made using straightforward Computer Vision algorithms. This report will provide an easy-to-use Python and OpenCV pipeline that may be applied to straightforward lane detection.

Language : Python (JupyterLab)

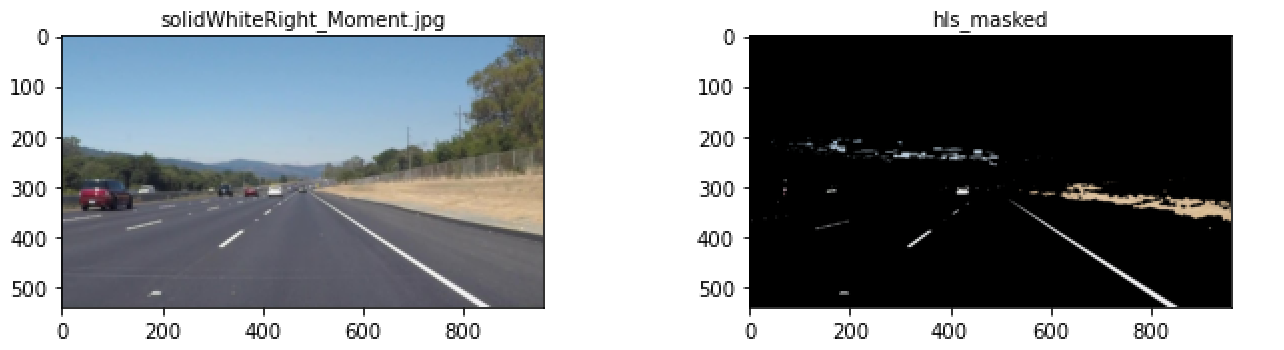
Modules used : OpenCV, moviepy, Matplotlib, NumPy

GitHub Link : [GitHub](https://github.com/Emadddddd/Lane-Detection)

1. Capture Screenshots from original video.



1. Convert the above image into HLS masked image so that rest of the environment isn’t visible.



1. Convert the HLS masked image into greyscale image. This helps us identify the yellow and white lanes.

Chart

Description automatically generated

1. Blur the image to reduce the noise and smoothen the image. We applied a kernel size of 15.

Chart

Description automatically generated

1. Apply Canny edge detection to the blurred image to sharpen the edges. The threshold was 50 (low) and 100 (high) respectively.

Chart

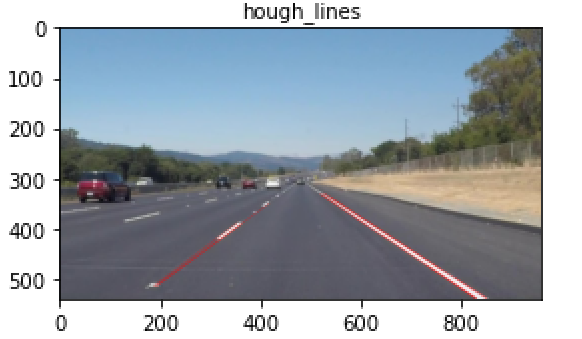
Description automatically generated

1. After canny edge, we can still see that there are other edges which are not lanes. To solve this, we use region of interest masking where we form a polygon based on the vertices. Rest of the image will be changed to black.

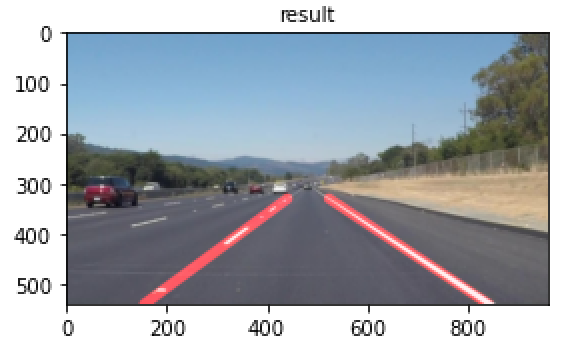
Chart

Description automatically generated

1. The Hough Transform is a method for locating lanes by locating each point along the line. From the region masked, we find the Hough lines and draw on the original images.



1. The lines are drawn on original image and result is obtained.





1. Video Processing is done using moviepy module. Lane lines are displayed on the road on each frame of the video. The resulting videos have been uploaded on the GitHub link.

CONCLUSION :-

In the end, the project was a success as all lanes were correctly marked on the video.

Only straight lane lines can be identified, which is a disadvantage of this technique. The method might be altered such that it finds curved lines by utilizing poly fitting. As the lines closest to the automobile in the given video are straight, there is no issue.

REFERENCES :-

Tutorials :-

* <https://www.youtube.com/watch?v=yvfI4p6Wyvk>
* <https://www.youtube.com/watch?v=G0cHyaP9HaQ>
* <https://www.youtube.com/watch?v=2t0I0URiS40>

Image Processing Code :-

* <https://www.purdue.edu/hla/sites/cea/wp-content/uploads/sites/15/2019/09/Guidelines_image-processing-using-python-opencv.pdf>
* <https://www.linkedin.com/pulse/finding-lane-lines-road-felipe-martinez>
* <https://www.geeksforgeeks.org/opencv-python-program-vehicle-detection-video-frame/>