

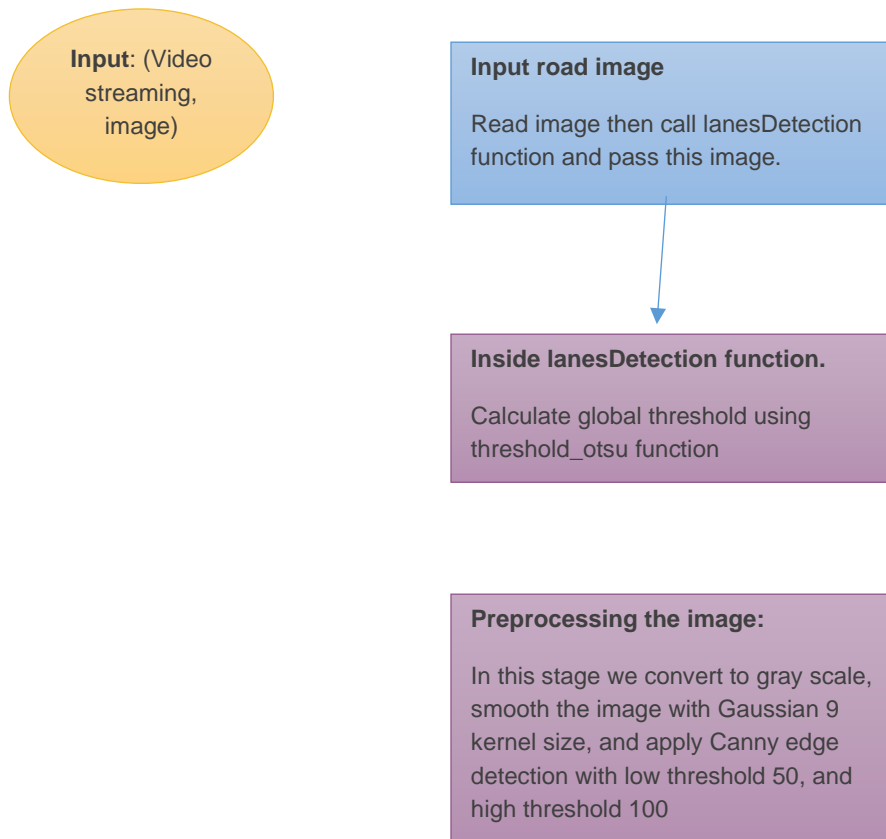
LANE DETECTION & OBSTACLES AVOIDANCE

Project Proposal

TEAM MEMBERS (TEAM 11)

Names	Section	BN	Email
Passant Abdelazim	1	21	Bassant.abdelgalil00@eng-st.cu.edu.eg
Raghad Khaled	1	31	Raghad592000@gmail.com
Menna Allah Ahmed	2	30	mennaahmedali77@gmail.com
Nada Elsayed	2	33	Nadaelsayed163@gmail.com

BLOCK DIAGRAM



Region of interest (ROI) height:

Calculate the ratio by dividing number of white pixels in thresholded image over black pixels then decide the height of ROI on ratio value.

Apply Hough transform and minimize edges:

Apply Hough transform then pass the output line and thresholded image to Reducelines() the edges depend on thresholded image from the beginning that function ensure that the lines on the white pixels.

Apply Hough transform and minimize edges:

Apply Hough transform again to out the final lanes.

Built in functions	Implemented functions
Canny	Reducelines()
Hough Transform	
Threshold_otsu	

LIMITATIONS:

- White cars
- Near cars
- Dark yellow
- Gaussian kernel = 9; but there is test cases fail on this kernel like packet 3

MAIN REFERENCE:

1. <https://www.kaggle.com/soumya044/lane-line-detection>

2. <https://www.researchgate.net/publication/220777443> An Adaptive Method for Lane Marking Detection Based on HSI Color Model

TRAILS

Perspective Transform of the ROI

In this approach we build a bird-eye view of the region of interest to get a clear view of the lane lines, by this we managed to capture lanes in the horizon.

But this approach failed in images where part of the car was presented in the image, because different vertices were required to set the right region of interest.



Figure 1 Original Image in gray scale

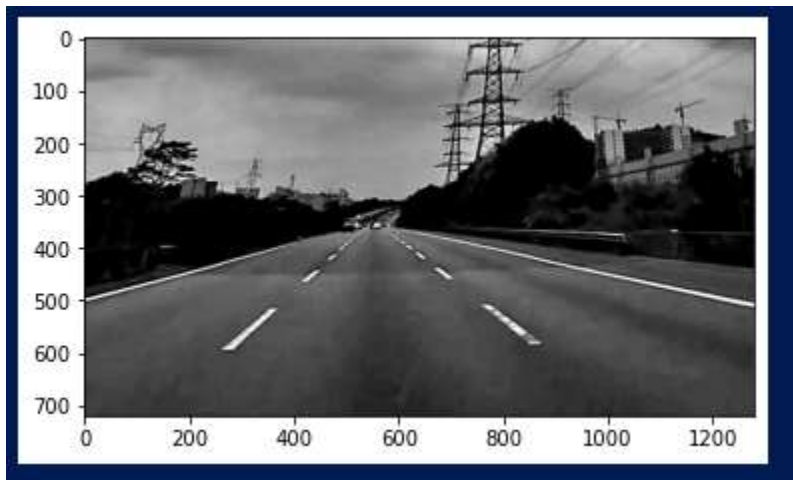


Figure 2 result of perspective transform



Figure 3 Original Image

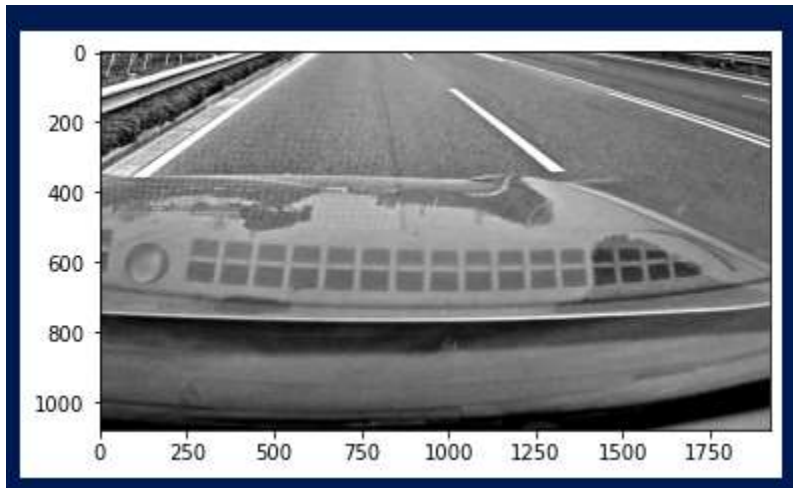


Figure 4 Result of perspective transform

Apply threshold with HSI Color Space

In this approach we convert RGB image to HSI Color space, we decided to choose this color space model because white/yellow lines can be easily separated.

Then we feed the output of this step into another function to perform edge detection and lane lines filtration explained below.

This approach was introduced in a research paper [2]



Figure 5 Original RGB image

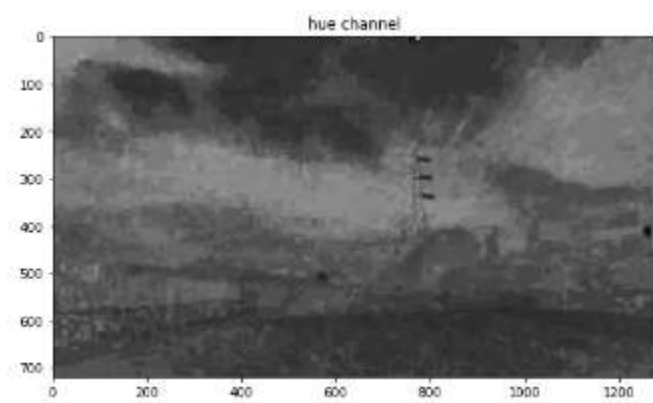


Figure 6 hue channel from HSI mode

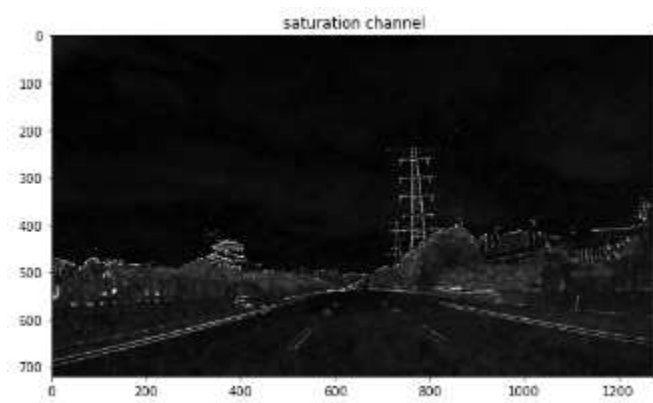


Figure 7 Saturation channel from HSI model

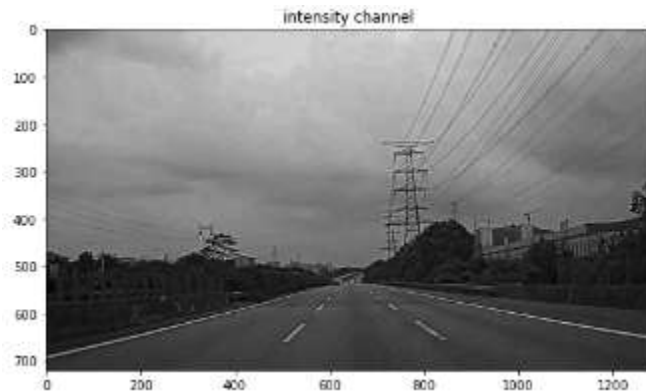


Figure 8 Intensity channel from HSI model

Filtering:

- **Filtering with magnitude:**

in this function we mainly depend on that the edges in road is minimum edges in the photo; it slides a patch on the Sobel edged on x direction (assuming lanes are vertical) image to calculate the number of edges and put it 0 if it exceeds a threshold; this threshold was tuned by us.

This function problem was:

- If the input image was with low texture
- If the lane covered with shadow (this problem was partially solved by HSI threshold)
- If road has different gray levels (this problem was partially solved by HSI threshold)

- **Filtering with angle:**

in this function we mainly depend on small sized of patches and the truth that lane has same angle; it slides a patch on the Sobel edged on x direction (assuming lanes are vertical) image to calculate edges direction and it calculates mean and variance to this set of tans and threshold on specific threshold; this threshold was tuned by us.

This function problem was:

- If the input image has curves
- If the lane covered with shadow (this problem was partially solved by HSI threshold)
- If road has different gray levels (this problem was partially solved by HSI threshold)

HSI threshold wasn't the perfect solution due to high computing power and time.

Then we try to use otsu threshold

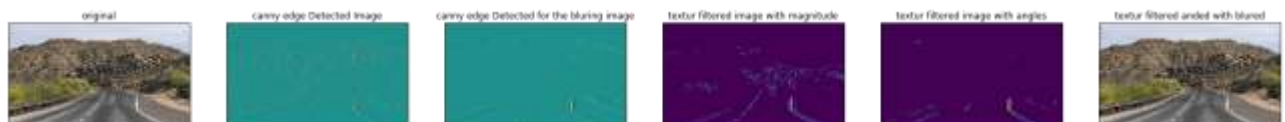


Figure 9 image with high texture

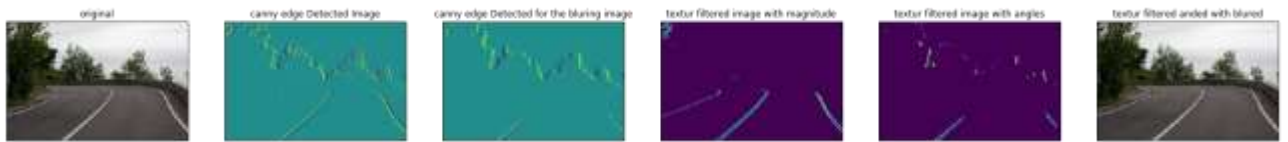


Figure 10 image with curves

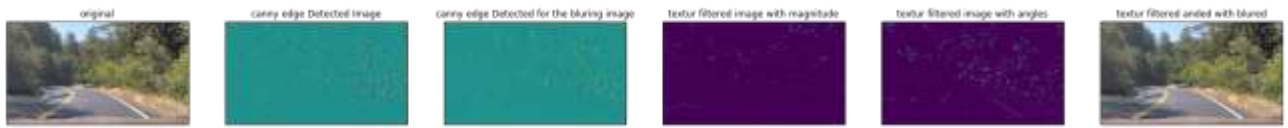


Figure 11 image with shadows