A Lane Detection, Tracking and Recognition System for Smart Vehicles (LTRSV)

1. Bharathi Devi[1](#_bookmark0), G. Siva Jyothi [2](#_bookmark0), K. Ethish Kumar [3](#_bookmark1), Ch. Manoj Sai [4](#_bookmark0),[⇑](#_bookmark2)

*Department of Computer Science and Engineering, Sasi Institute of Technology and Engineering, Tadepalligudem.*

a r t i c l e i n f o

*Article history:*

Received 18 Feb 2020

Accepted 00 xxxx 2020

Available online

*Keywords:*

Lane detection,

Drowsiness detection

Vehicle detection,

Opencv,

Canny edge detection,

Hough transform,

Support Vector Machine (SVM)

1. A b s t r a c t

The main purpose of this article Lane Detection, Tracking and Recognition System for Smart vehicles(LTRSV) is to reduce the road accidents and increase safety, And it have two camera’s one camera will monitor the driver’s drowsiness, and another camera will detect the road lanes and vehicles, The detection of drowsiness by using algorithm SVM(Support Vector Machine) , To detect road lanes it uses different techniques like Canny edge detection technique and Hough Transform algorithms, For Vehicle detection we use Haar like Cascade Classifier .

2 Introduction

2.1Methodologies

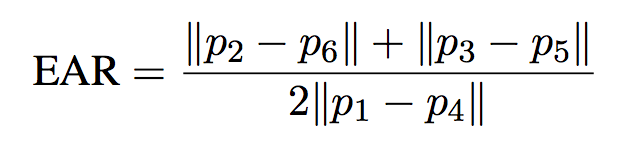
The project involves monitoring driver drowsiness and detection of road lane lines in real-time video using Python and OpenCV. OpenCV means “Open Source Computer Vision”, which is a library that has many useful tools for analyzing video.

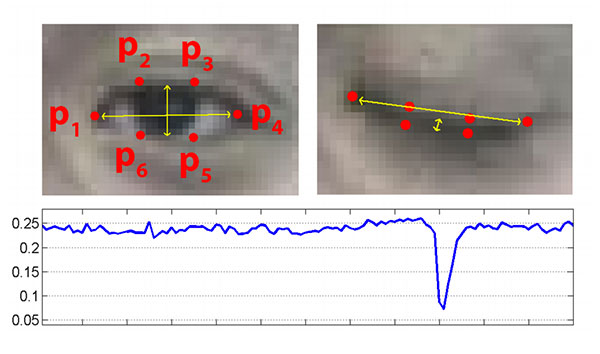
2.2 Drowsiness detection

It is used to detect the drowsiness of the driver in which if the driver feels sleepy at any time, facial expression will be detected and alert the driver with voice assistant, otherwise there is a chance to occur accidents. The facial expressions of the driver is always analyzed until he reaches the destination. Here to recognize the drowsiness It uses the Support Vector Machine algorithm. The entire drowsiness detection solution is divided into following major modules:

1. Eye shape detector
2. Eye shape predictor

Eye shape detector and Eye shape predictor works on the basis of trained data set given. A trained data set and the range value is given as an input and the position of the eye is calculated. If it crosses the given range then a voice alert is given to the driver as either wake up or park the car. The formula used to detect the shape of the eye is





* 1. Lane Detection

2.3.1 Edge Detection

The goal of edge detection is to identify the boundaries of objects within images. A detection is used to try and find regions in an image where there is a sharp change in intensity. We can recognize an image as a matrix or an array of pixels. A pixel contains the light intensity at some location in the image. Each pixel's intensity is denoted by a numeric value that ranges from 0 to 255, an intensity value of zero indicates no intensity if something is completely black whereas 255 represents maximum intensity something being completely white. A radiant is the change in brightness over a series of pixels. A strong gradient indicates a steep change whereas a small gradient represents a shallow change.





Original Images

2.3.2 Convert to gray scale

Converting original image gray scale has its benefits .we have to find yellow and white lines as in original image, by converting to gray scale it increases the contrast of the line with respect to road.





Gray scale image

2.3.3 Gaussian blur

Each of the pixels for a grayscale image is described by a single number that describes the brightness of the pixel. In order to smoothen an image, the typical answer would be to modify the value of a pixel with the average value of the pixel intensities around it. Averaging out the pixels to reduce the noise will be done by a kernel. This kernel of normally distributed numbers(np.array([[1,2,3],[4,5,6] ,[7,8,9]]) is run across our entire image and sets each pixel value equal to the weighted average

of its neighboring pixels, thus smoothening our image.

Below is the image of less noise:



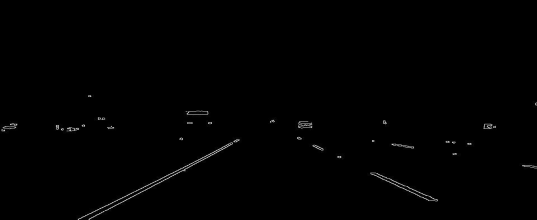


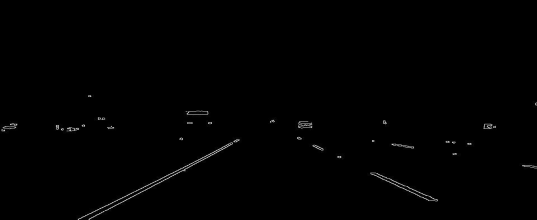
Gaussian blur

2.3.4 Canny edge detection

An edge corresponds to a region in an image where there is a sharp change in the intensity/color between adjacent pixels in the image. A strong gradient is a steep change and vice versa is a shallow change. So in a way we can say an image is a stack of matrix with rows and columns of intensities. This means that we can also represent an image in 2D coordinate space, x axis traverses the width (columns) and y axis goes along the image height (rows). Canny function performs a derivative on the x and y axis thereby measuring the change in intensities with respect to adjacent pixels. In other words we are computing the gradient (which is change in brightness) in all directions. It then traces the strongest gradients with a series of white pixels.

Below Image after canny edge:.



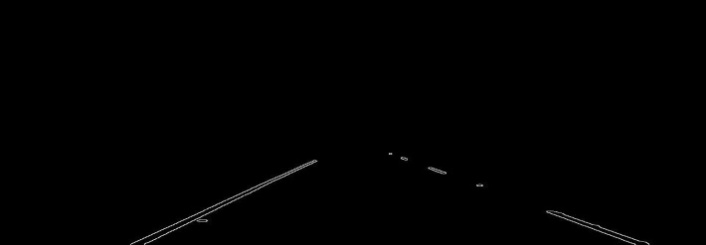


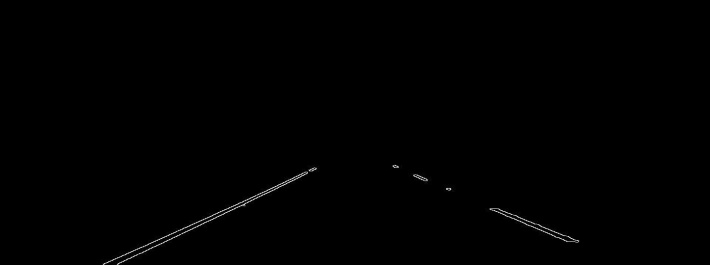
Canny Edge Image

2.3.5 Region of interest

The dimensions of the image are chosen which will contain the road lanes and mark it as our region of interest or the triangle .then a mask is created which is same as the dimension of the image which would essentially be an array of all zeros. Now we fill the triangle dimension in this mask with the intensity of 255 so that our region of interest dimensions are white. Now I will do a bitwise AND operation with the canny image and the mask which will result in our final region of interest.

Below images specify the region of interest after masking:





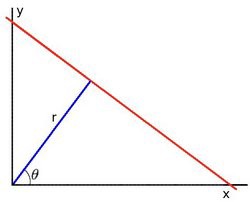
Region of Interest Image

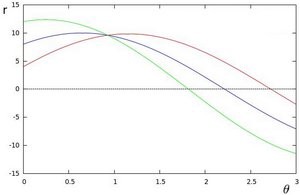
2.3.6 Hough transformation

Hough transformation is the technique to find out lines by identifying all points on the line. This is done by representing a line as point. And points are represented as lines /sinusoidal .if multiple lines /sinusoidal pass through the point, we can deduce that these points lie on the same line.

We make use of Hough transform technique that will detect straight lines in the image and thus identify the lane lines. We know that a straight line is represented by the below equation:

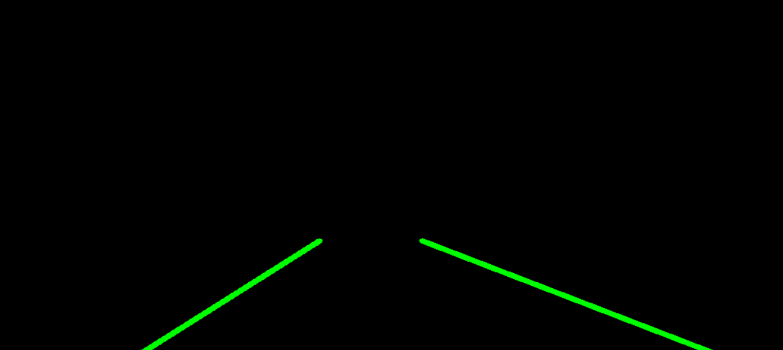
y= mx + b





Below is combined image:

After finding out Hough lines from Region Of interest images and then gives the original Image



**3. Conclusion**

In the methodology, we made use of the OpenCV library and its functions such as the Canny Function through which we achieved edge detection. Then we prepared a mask of zero intensity and mapped our region of interest by performing the bitwise operation. Then we used the Hough Transform technique that detected the straight lines in the image and identified the lane lines. We made use of the polar coordinates since the Cartesian coordinates don’t give us an appropriate slope of vertical and horizontal lines. Finally, we combined the lane image with our zero-intensity image to show lane lines



