Implementation of Lane Line Detection using HoughTransformation and Gaussian Smoothing

P Ramu[1], Chintakuntlawar Vignan[2], Polisetti Radhika[3], Mahitha Chegoor[4], M Sathyanarayana[5]

[1,5] Professor, Dept. of Computer Science and Engineering, SNIST, Hyderabad-501301, India
[2,3,4] B.Tech Scholars, Dept. of Computer Science and Engineering Hyderabad-501301, India

---***---

***Abstract* :- Autonomous vehicles are currently advancing quickly to make driving easier for people. In order to accomplish this, the vehicles must be able to recognize the lane of the road in order to maintain their position and avoid car accidents. In this article, we use the OpenCV library to implement and improve a computer vision system for lane detection on roads on both images and videos. This objective will be accomplished using the well-known line detection technique known as the Hough probabilistic transform. In this study, the grey scale picture, camera calibration, masking filter, and letter-on Canny edge detection are used as preprocessing approaches before the Hough transform. Prior to moving on to video, which is a more realistic case study, we first build our method on images. We then demonstrate our detection with a red line on the screen.**

Keywords: Gray Scaling, Hough Transformation, Gaussian Smoothing, Canny Edge Detection

I. INTRODUCTION

As traffic in cities increases, road safety becomes more and more important. Most accidents on the avenues are caused by people exiting lanes against the rules. Most of these result from the driver's negligence.[1]

Inconsistent and slow behavior. Both automobiles and pedestrians must maintain lane discipline when using the road. A type of technology that enables cars to understand their surroundings is computer vision. It is a subset of artificial intelligence that aids in the comprehension of image and video input by software. Finding the lane markings is what the technology aims to do. Its objectiveis to make the environment safer and the traffic situation better. The proposed system's functionality can range fromshowing the position of the road lines to the both on any outdoor display to more complex applications like lane switching soon to lessen traffic-related concussions. It's critical to accurately recognize lane roads in lane recognition and departure warning systems.[2] The predicting lane borders system, which is installed in automobiles, warns drivers when a vehicle crosses a lane boundary and directs them to prevent collisions. It is not always required for lane borders to be clearly apparent for these intelligent systems to ensure safe travel; for example,the system may have trouble effectively detecting lanes due to poor road conditions, a lack of paint used to define the lane limits, or other circumstances. Other variables may include environmental impacts like fog brought on by constant lightning conditions, day and night conditions, shadows produced by objects like trees or other vehicles, or streetlights. These elements make it difficult to tell a person apart from a road lane in the background of a picture that has been captured. [3].

II. BACKGROUND STUDY (LITERATURE)

Statistics on road accidents show that 95% of the causesthat led to an accident were human-related, and that driver error was a factor in up to 73% of incidents. This further suggests that the driver's negligence is the primary cause oftraffic accidents.[1]

With such serious issues with traffic safety, the advancement of vehicle safety technology has emerged as a crucial area for research in an effort to lower the number of persons who are hurt or killed in auto accidents. With today's cutting-edge auto safety technology, the driver's driving safety is improved primarily by computer, automatic control, and information fusion technologies, which also provide the automobile more intelligence while driving.[2]

In the intelligent transportation system, smart automobiles are in the lead. It fully completes multi-level aided driving operations in addition to environment sensing, planning, and decision-making. It is an entirely new, cutting-edgesystem. Smart cars can fully comprehend the information about the environment around them using their own sensors, which is necessary for other operations.[3] However, no two roads are the same in the entire planet. This is due to the fact that there are too many things thatcan affect the state of the roads. It is simple for drivers to make poor driving decisions based on specific road condition information when natural and artificial elements change, which causes traffic accidents. Traffic accidents can be successfully avoided if the driver is able to make an accurate assessment of the road conditions and provides timely guidance to correct any erroneous driving behavior. Right now, the smart car system must effectively assess the state of the roads and provide the driver with precise driving instructions.[4]

Global trials for unmanned vehicles have already started.The present research on autonomous vehicles is focused on how to detect and recognize road signs, obstructions, and pedestrians in an unfamiliar area accurately and instantly. In conclusion, the thorough investigation of lane detection and recognition technologies offers significant practical utility.[5]

III. METHODOLOGY

To record the landscape of the road, a CCD camera is fixed to the front-view mirror. The baseline is expected to be set up as horizontal to simplify the issue and ensure that the image's horizon is parallel to the x-axis. If not, the calibration data can be used to alter the camera's image. Each lane boundary marking consists of two edge lines that are often approximate rectangles.[1]

In this study, it was envisioned that the algorithm would receive a 620x480 RGB-colored image as its input. Therefore, to reduce processing time, the algorithm attempts to turn the image into a grayscale image.[2] Second, the image will make it difficult to discern edges correctly when there is noise present. F.H.D algorithm as a result the work of Mohamed Roushdy (2007) was used to improve edge detection. the edge detector next was used to create an edge image by applying a clever filter and automatically obtaining the edges. It has significantly streamlined the image edges, lowering the quantity of learning data needed. [3]

Next edged a right and left lane border segment is produced by the line detector after receiving the image.

The horizon was identified as the projected point at which these two-line segments would intersect. The information in the edge picture picked up by the Hough transform wasused to execute the lane boundary scan. A list of right- side and left-side points were produced by the scan. The lane borders were finally represented by a pair of hyperbolas that were fitted to these data points. The hyperbolas are presented on the original color image for viewing purposes.[4]

IV. ALGORITHMS

4.1 Gaussian Smoothing

A 2-D convolution operator called the Gaussian smoothing operator is used to 'blur' images and eliminate noise and detail. It is comparable to the mean filter in this regard, but itmakes use of a different kernel that simulates a Gaussian (or "bell-shaped") hump.

This blurring technique produces a smooth blur that looks like you're seeing through a translucent screen, which isnoticeably different from the bokeh effect that is created by an out-of-focus lens or the shadow of an object under normallighting.

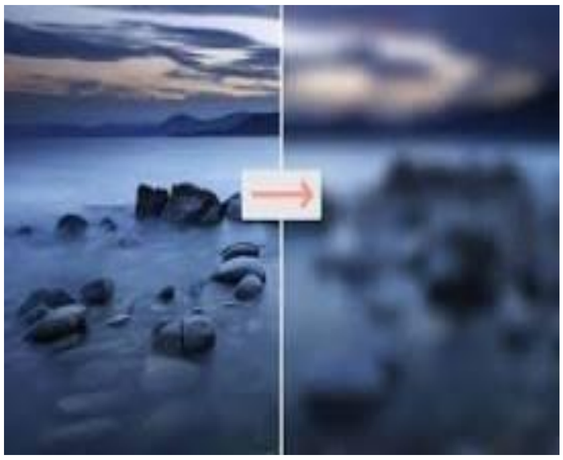


Fig 1. Gaussian Smoothing

4.2 Canny Edge Detection

With the use of the Canny edge detection technology, the amount of data that needs to be processed can be drastically reduced while still extracting meaningful structural information from various vision objects. It is frequently usedin many computer vision systems.

The edges of a picture are found using Canny Edge Detection. It employs a multistage method and accepts a grayscale image as input. The Canny () method of the imgproc class can be used to conduct this process on a picture; the syntax for this method is as follows.

The Canny Operator's objectives were made clear. Good detection entails being able to identify and indicate all actual edges. Low separation between the detected edge and the actual edge indicates good localization. Only one response is allowed for each edge, clearly.

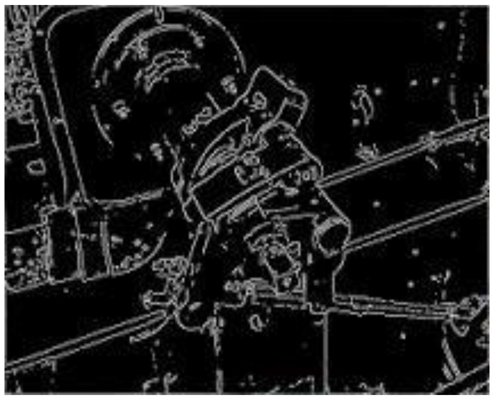


Fig2.Canny Edge Detection

4.3 Hough Transformation

The practice of automatic driving makes extensive use of lane detection technology. This study use the Hough transform to identify the valuable vertical straight lanes. First, picture preprocessing extracts the image's edge features. The area of interest in the photograph is then drawn out.

Two edge points that are on the same line will cause their associated cosine curves to cross at a particular (,) pair. As a result, the Hough Transform algorithm locates pairings of (,) that have more intersections than a predetermined threshold to identify lines.

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure.

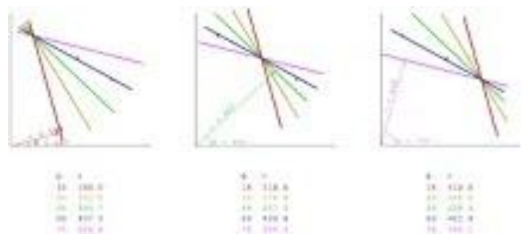


Fig3. Hough transformation

4.4 Gray Scaling

The process of gray scaling involves changing an image from another color space, such as RGB, CMYK, HSV, etc., to a variety of grayscales. There are several shades of black and white.

Grayscale refers to a situation in which each pixel in a digital image solely contains information about the light's intensity. Usually, just the range from deepest black to brightest white is visible in such photos. In other words, the only colors present in the image are black, white, and grey, the latter of which includes several shades.

New grayscale image = ((0.3 * R) + (0.59 * G) + (0.11 * B) is the new equation that results. This equation shows that Red contributed 30%, Green provided 59%, the highest of the three colors, and Blue contributed 11%.

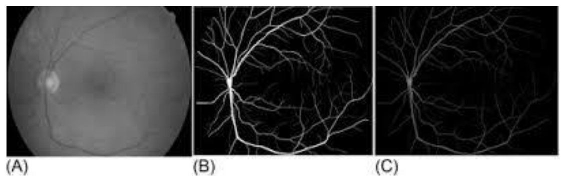


Fig 4. gray Scaling

4.5 Region of Interest

An image's region of interest (ROI) is a section you want to filter or manipulate in some way. An ROI can be shown asa binary mask picture. Pixels in the ROI are set to 1 in the mask picture, while pixels outside the ROI are set to 0.

The image window is expanded to include a default region with a rectangular shape and the associated colour. After that, click each vertex of the ROI polygon.

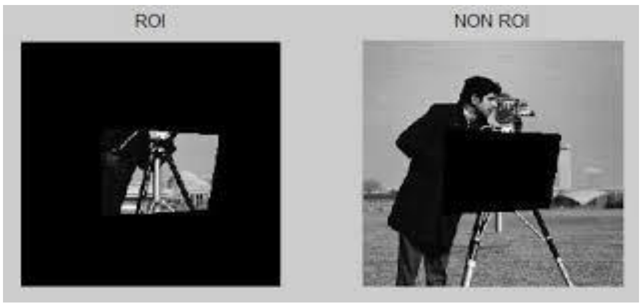


Fig5. Region of Interest

V. IMPLEMENTATION

The project involves finding lane lines in a picture using Python and OpenCV. OpenCV is an acronym for "Open- Source Computer Vision," and it refers to a collection of software tools for image analysis.

A. Canny Edge Detection

The goal of edge detection is to identify item borders in images. In an image, a detection is used to look for regions where the intensity sharply varies. It is possible toidentify an image as a matrix or a collection of pixels. Theamount of light existing at a particular location in an image is represented by a pixel. A value of zero implies that something is completely black, while a value of 255 indicates that something is completely white. Each pixel's intensity is represented by a numeric value ranging from 0to 255. A gradient is a pattern of pixels with different brightness's. While a little gradient denotes a shallowshift, a high gradient denotes a steep change.

Additionally, there is a bright pixel in the gradient image anywhere there is a strong gradient, or wherever there is an abrupt change in intensity (rapid change in brightness).By tracing out each of these pixels, we may obtain the edges. The edges in our road image will be found using this idea.

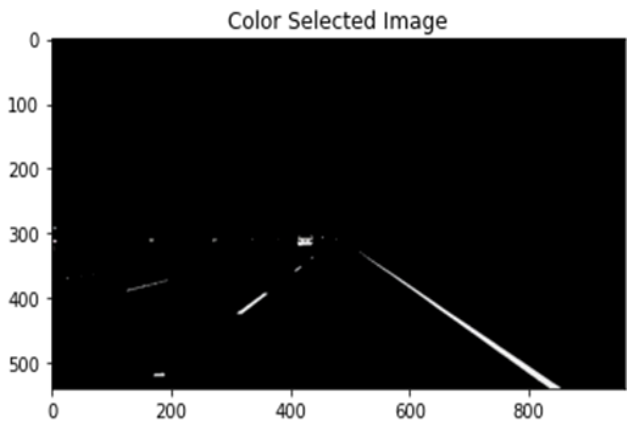
Original Image

Our image will be read into an array after being loaded:image = cv2.imread('road.jpg')

B. Gray Scaling

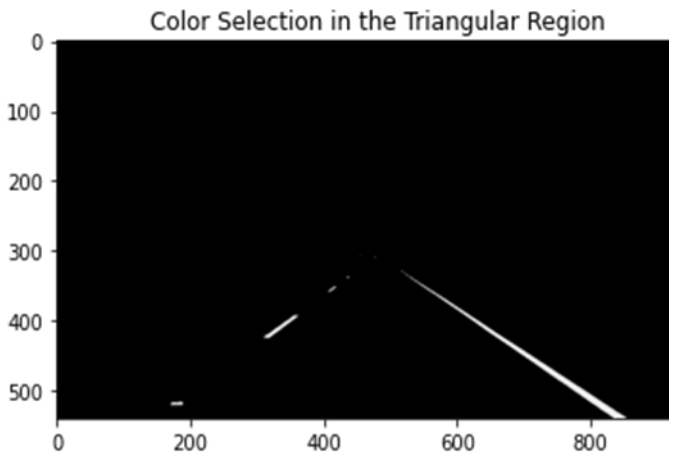
The image is then made grayscale: gray image = cv2.cvtColor(image,cv2.COLOR_RGB2GRAY)

The picture has now been made grayscale:



C. Gaussian Smoothing

Each pixel in a grayscale image is represented by a single number that represents its brightness. Changing a pixel's value to match the average value of the around pixels' intensities is a typical way to smooth an image. In order to lower noise, a kernel will average the pixels. Our entireimage is smoothed down using this kernel of normallydistributed numbers (np. array ([[1,2,3], [4,5,6], [7,8,9]]),with each pixel's value set to the weighted average of its neighbors. In this instance, blur=cv2 will be used with a 5x5 Gaussian kernel. Gray image, Gaussian Blur (5,5),0); The image with less noise is shown below.



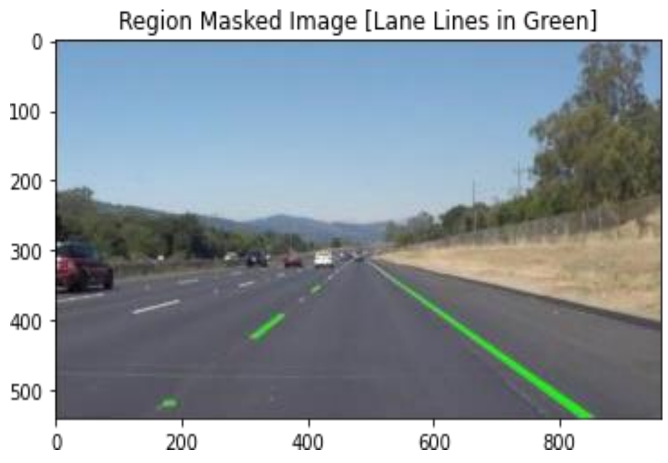
D. Region of Masked Image

The line detector operator in a convolution-based method consists of a convolution mask tailored to find the presence of lines with a specific width n and orientation. Here are the four convolution masks for detecting straight, oblique (+45 degrees), oblique (45 degrees), and vertical lines in a picture.

color_select

[color_thresholds | ~region_thresholds] = [0, 0, 0]
line_image

[~color_thresholds & region_thresholds] = [9, 255, 0]

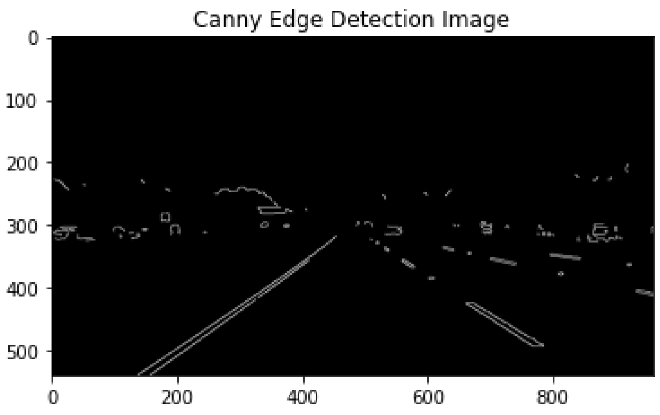


E. Canny Edge Detection

An edge is a spot in a picture where the contrast between adjacent pixels in the image drastically varies. A considerable gradient is one that is steep, as opposed to one that is shallow. An image can be compared to a matrix with rows and columns of intensities in this manner. With the x axis travelling the width (columns) and the y axistraversing the image height, a picture can also be represented in 2D coordinate space (rows). The Canny function computes a derivative on the x and y axes to determine the brightness difference between adjacent pixels. In other words, we are figuring out the gradient (or change in brightness) in every direction. It then uses astring of white pixels to trace the gradients that are the strongest.

```
canny_image = cv2.Canny(blur, 100, 120)
```

Here is the result of applying the Canny function on the image:



The low threshold and high threshold functions can be used to distinguish between neighboring pixels that adhere to the strongest gradient. If the gradient exceeds the upper threshold, it is accepted as an edge pixel; if it falls below the lower level, it is discarded. Only if the gradient is connected to a strong edge and falls within the parameters is it permitted.

The entirely black areas correspond to modest intensity variations between adjacent pixels, whereas the white line shows a spot in the image where there is a substantial change in intensity above the threshold.

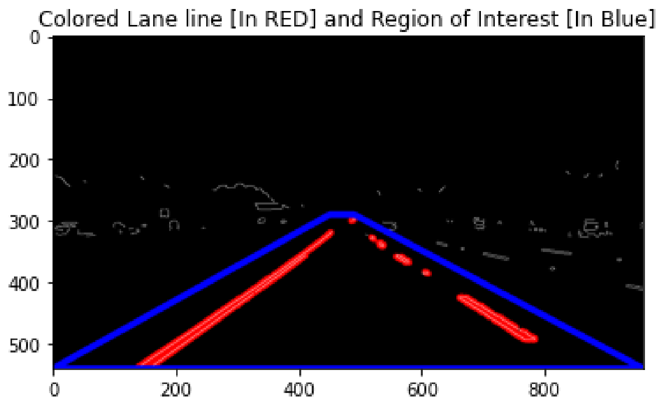
F. Region of Interest

The image's size was selected to show the lanes on the roads and to highlight the triangle as our area of attention.

Then, a mask that has the same dimension as the image and is essentially an array of all zeros is created. In order to render the dimensions of our region of interest white, we will now fill the triangle dimension in this mask with 255. To obtain our final region of interest, we will now perform a bitwise AND operation on the clever picture and the mask.

```
lines_edges = cv2.addWeighted(color_edges, 0.8,
line_image, 1, 0)

lines_edges = cv2.polylines(lines_edges,vertices, True,
(0,0,255), 10)

plt. imshow(image) plt.title("Input Image") plt.show()
plt.imshow(lines_edges)

plt.title("Colored Lane line [In RED] and Region of
Interest[In Blue]")

plt.show()
```

The picture, also known as the masked image, is shown below after performing a bitwise operation on the clever image and the mask:



G. Hough Transform

Now, we use the Hough transform method to identify thelane lines in the image by looking for straight lines.

The following equation describes a straight line:
Y=mx+b

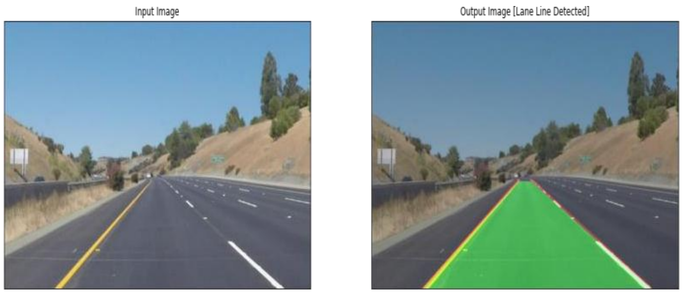
The slope of the line is just a climb over a run. The line canbe represented as a single dot in Hough Space if the y intercept and slope are specified. There are many lines, each with a different ‘m' and 'b' value, that can pass through this dot. There are numerous lines that can cross each point, each with a distinct slope and y intercept value. However, a single line runs between the two locations. By examining the point of intersection in sufficient space, which reflects the ‘m' and 'b' values of a line that crosses both places in Hough Space, we may determine this.

We must first create a grid in our Hough space before we can identify the lines.

The slope and y intercept values of the line are represented by a bin in the grid. The bin to which a point of intersectionin a Hough Space belongs will cast a vote for that point. The container with the most votes will be where we draw our line. On the other hand, a vertical line has an infinitely steep slope. Therefore, we shall utilize polar coordinates rather than Cartesian coordinates to express vertical lines. Thus, the equation for our line is as follows:

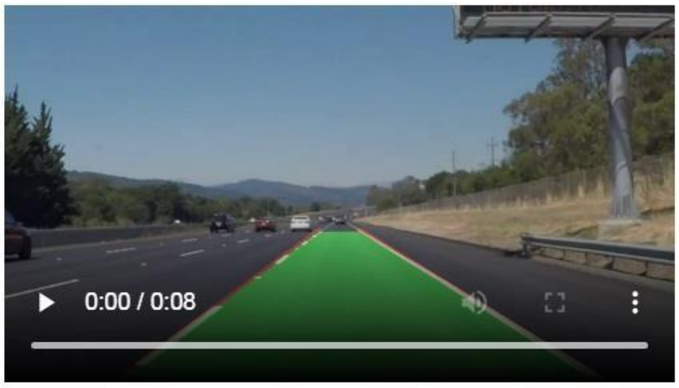
```
def draw_the_lines(img, Lines):

img = np.copy(img)

blank_image = np.zeros((img.shape[0], img.shape[1], 3), dtype=np.uint8)

for line in Lines: for x1, y1, x2, y2 in line: cv2.line(blank_image, (x1, y1), (x2, y2), (0, 255, 0), thickness=4)

img = cv2.addWeighted(img, 0.8, blank_image, 1, 0.0) return img

lines = cv2.HoughLinesP(cropped_image, rho=2, theta=np.pi/60, threshold=50, lines=np.array([]), minLineLength=40, maxLineGap=80)

image_with_lines = draw_the_lines(image, lines)
```

Here are the resulting pictures:



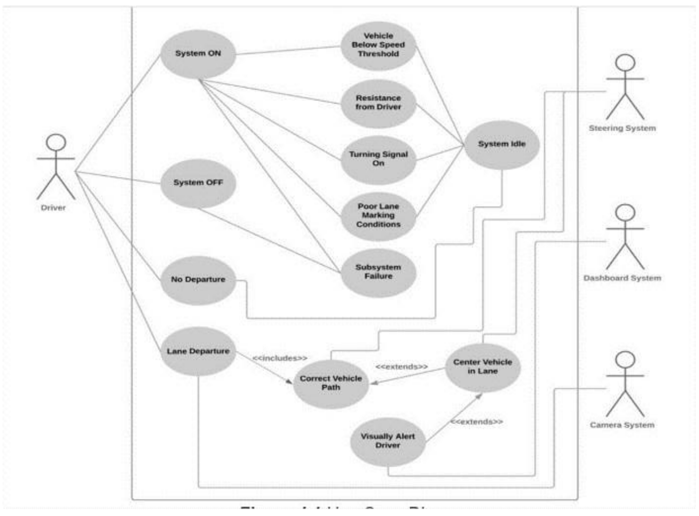
Now, utilizing the video capture feature and reading each frame in the movie with the aid of the while loop, the same process can be used to a video that comprises a number of frames or images as follows:

```
cap = cv2.VideoCapture('test_video.mp4')while cap.isOpened():

ret, frame = cap.read()

frame = process(frame) cv2.imshow('frame', frame)if cv2.waitKey(1) & 0xFF == ord('q'):

break
```



VI UML Diagrams

6.1 Use Case Diagram



6.2 State Diagram

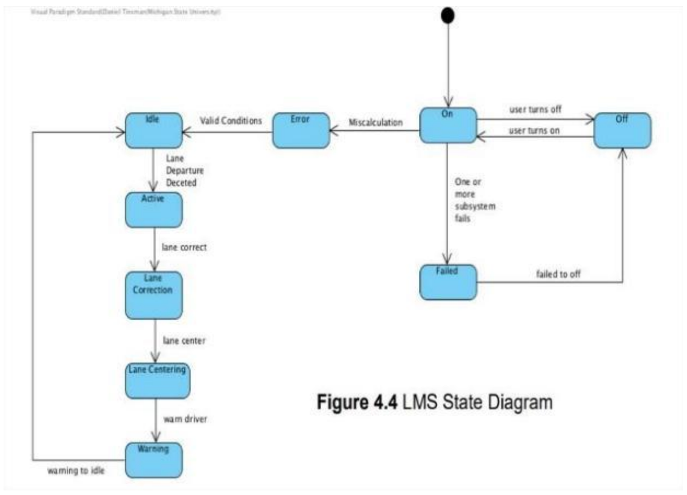
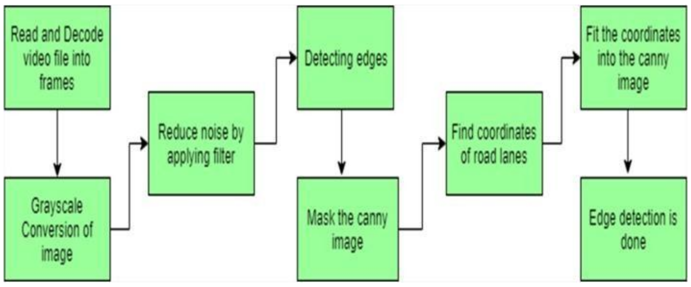


Figure 4.4 LMS State Diagram

6.3 Activity Diagram


Read and Decode video file into frames
Grayscale Conversion of image
Reduce noise by applying filter
Detecting edges
Mask the canny image
Find coordinates of road lanes
Fit the coordinates into the canny image
Edge detection is done

VII Advantages

No further data, such as lane width, time to lane crossing, or offset between the center of the lanes, is needed by the proposed system.

Additionally, not needed are coordinate transformation andcamera calibration.

The suggested lane detecting technology can be used in a variety of weather circumstances on straight, painted, and unpainted roadways.

VIII Conclusion

We employed methods like the Canny Function and the OpenCV package to achieve edge detection. Then, using a zero-intensity mask, we used the bitwise method to map our region of interest. Then, the image's straight lines and lane lines were located using the Hough Transform technique. Since Cartesian coordinates did not provide a sufficient slope for vertical and horizontal lines, we used polar coordinates instead. To display lane lines, we finally combined the original image with our zero-intensity image.

IX Future Works

We utilize a modular implementation strategy since it makes updating algorithms easy and allows us to continue developing the model in the future. We place the model's pickle file in the appropriate places so that it may be easily moved to products. As a result, it may be simple to skip compiling the entire huge code. Another way to strengthen the concept is to imagine a future in which it is possible to see the road at night or in the dark. The process of color selection and recognition is quite efficient in daylight. Although adding shadows will make things a little noisier, driving at night or in low light will be more difficult (e.g., heavy fog).Furthermore, this study can only identify lanes on bituminous roads; loamy soil roads, which are common in Indian villages, are not included. This project can be enhanced to detect and prevent accidents on roads with loamy soil that are present in community

X Acknowledgement

We would like to extend our gratitude to P Ramu for his astute suggestion, wise advice, and moral support throughout the writing of this work

XI References

1. A. Berninger, J. Schroeder, D. Somary, D. Tinsmanand M. Wojno, Lane Management System – Team 1,Michigan State University, October 2018. http://www.cse.msu.edu/~wojnomat/

2. Berninger, J. Schroeder, D. Somary, D. Tinsman and M. Wojno, "Requirements Outline", Michigan State University, October 2018.

3. "How does lane departure warning work?", Ziff Davis, LLC. PCMag Digital Group, 2018. https://www.extremetech.com/extreme/165320-what-is- lane-departure-warningand-how-does-it-work

4. "Lane Departure Warning", National Safety Council MediaRoom, 2018. https://mycardoeswhat.org/safety-features/lane-departure-warning/

5. "Lane DepartureWarning System", Robert Bosch GmbH., 2018. https://www.boschmobilitysolutions.com/en/products- and-services/passenger-cars-and-light- commercialvehicles/driver-assistancesystems/lane- departure-warning/

6. “Lane Keeping Assist”, National Safety Council Media Room, 2018. https://mycardoeswhat.org/safetyfeatures /lane-keeping- assist/

7. “Lane Keeping System”, Ford Motor Company,2018. https://owner.ford.com/support/howtos/safety/driver- assist-technology/driving/how-touse-lane-keeping- system.html

8. “Lane Keeping System”, Robert Bosch GmbH.,2018. https://www.boschmobilitysolutions.com/ en/products- and-services/passenger-cars-and-light- commercial vehicles /driver assistance-systems/lane- keeping-support/

9. “Toyota Corolla Lane-Centering Tech -- A Step Toward to Self-Driving -- Cures Annoying "Ping-Pong", Forbes Media LLC, 2018. https://www.forbes.com/sites/doronlevin/ 2018/04/13/toy ota-corolla-lanecentering-techa-step-toward-to-self- driving-cures-annoying-ping-pong/#39e17fbc7b26

10. Monticello, Mike. “Guide to Lane Departure Warning &Lane Keeping Assist,” Consumer Report,