Advanced Driver Assistant System

Sanjay S.[1], *Vasanthakumar S.*[2],
Mohamed Rifayee Hussain Z.[3],
Saravana Kumar K.[4],
Dineshkumar S.[5], and *Rathika P. D.*[6]

Abstract

Road accidents are the most terrifying thing that can happen to a driver. Worst of all, we refuse to learn from our mistakes along the way. The majority of road users are aware of the general rules and safety measures to take while on the road but injuries and crashes are caused by the negligence of road users. The most common cause of accidents and collision is human error. The aim of this project is to automate and improve the safety of vehicles. Lane Departure Warning System (LDWS) and Emergency Driver Assist System (EDAS) are described in this paper. The LDWS uses camera to track lane markers to see if the driver is drifting accidentally. In this project, whenever the vehicle is moving out of the lane, the device gives the driver a warning in the form of audio or visual signal. Whenever the driver's attention deviates from driving activity for a particular interval of time, EDAS alerts the driver.

Keywords : ADAS, EDAS, Lane Detection, LDWS, OpenCV, Sliding Window, TensorFlow

NOMENCLATURE

ADAS Advanced Driver Assist System
LDWS Lane Departure Warning System
EDAS Emergency Driver Assist System
CHT Circular Hough Transform

I. INTRODUCTION

An autonomous vehicle or a driverless vehicle operates itself and performs necessary functions with no human intervention through the flexibility to sense its surroundings. Sensors get the distance between neighboring cars and cameras to detect paths and traffic

Manuscript Received : May 24, 2021 ; Revised : June 24, 2021 ; Accepted : July 7, 2021. Date of Publication : August 5, 2021.

[1] *Sanjay S.,* Student
Email : sanjaisankerrae@gmail.com ; ORCID iD : https://orcid.org/0000-0001-6784-5057
[2] *V. Kumar S.,* Student
Email : vasanthakumars931@gmail.com ; ORCID iD : https://orcid.org/0000-0003-1735-9936
[3] *M. R. Hussain Z.,* Student
Email : mohamedrifayeehussain@gmail.com ; ORCID iD : https://orcid.org/0000-0002-6649-1442
[4] *S. Kumar K.,* Student
Email : saravanakumar28900@gmail.com ; ORCID iD : https://orcid.org/0000-0003-1735-9936
[5] *D. Kumar S.,* Student
Email : dineshkumarsaravanan01@gmail.com ; ORCID iD : https://orcid.org/0000-0002-9742-8569
[6] *Rathika P. D.,* Assistant Professor
Email : pdrathika@gmail.com ; ORCID iD : https://orcid.org/0000-0001-5543-9035
[1,2,3,4,5,6] Department of Robotics and Automation, PSG College of Technology, Avinashi Road, Peelamedu, Coimbatore – 641 004, Tamil Nadu, India.

DOI : https://doi.org/10.17010/ijcs/2021/v6/i3-4/165410

signs. The most crucial part of an *autonomous car* is *lane detection*, it should correctly trace the path to guide the vehicle accordingly. Advanced Driver Assistance Systems (ADAS) are an automated system which work for the safety of people and improve the vehicle safety by giving warning to the driver. The functions of ADAS include monitoring, warning, adaptive, and automated. This paper explains about the warning system which alerts or warns the driver in case of an emergency situation and gives a message to the driver to rectify it. In this ADAS warning system, instead of taking control, the system provides information and warning for the drivers to interpret on their own.

II. LITERATURE SURVEY

Lane Detection is the most critical part of driver assist system. Each research paper gives different ideas and methodologies for detecting the lane. Also, for the Drowsiness detector, this research paper gives a detailed explanation and a clear view to understand the EDAS system :

Camera-based real world lane detection with lane departure warning system for tracing the lane lines, Canny Edge Detection and Hough transform algorithms are used to extract some of the important features [1]. Some faded gray regions will appear in the output of the Hough transform and it needs to distinguish between the lane and the faded regions.

Sometimes, the driver needs to change the lane and drive off the road at this situation and there is a high chance of occurrence of accidents [2]. Seat belts and airbags can reduce damage but they won't prevent accidents. With this Collision warning system, accidents can be avoided to the maximum extent. The proposed method uses Yolo models to detect other cars and alert the driver.

A non-intrusive driver's drowsiness system based on Computer Vision and Artificial Intelligence has been presented [3]. This system uses advanced technologies for analyzing and monitoring the driver's eye state.

Based on Circular Hough Transform (CHT), a new algorithm has been developed for yawning detection [4]. The detection system uses three steps : face detection, mouth region localization, and wide-open mouth detection.

For the Yawn detection feature, eye states are analyzed from the video of several topics [5]. It locates the face and eye of the driver using the Haar Cascade feature. Circular Hough Transform allows the detection of the iris center and the points of intersection for the two lids. The result of the signal after the extraction of two geometric descriptors is non-linear and non-stationary. The results have shown a good accuracy rate. To improve them, hybrid systems which use 3D force estimation can be used to achieve the required accuracy.

III. OBJECTIVE

The main objective is to trace the lane and adapt in all conditions (low light, smoggy road, rainy road). Lane detection is the most important task of ADAS, and the accuracy of detection should meet the expectation. Several algorithms have been proposed and their differences are mainly in image preprocessing, lane model, selected model fitting method, and tracking strategy. However, it is difficult to expect a high detection rate in complex situations involving shadows, varying illumination conditions, bad conditions of road paintings, and lane marks such as solid lines, segment lines, double yellow lines, pavement or physical barriers. Therefore, making the detection algorithm to accommodate the complex environment is an important issue.

IV. PROPOSED SOLUTION

Many solutions have been proposed for lane detection. Among all, the most common traditional method is Hough Transform. It is a feature extraction technique used in image analysis, computer vision, and digital image processing. It can be used to detect lines, circles or

Fig. 1. Hough Transform

other parametric curves, but this Hough Transform method does not fit complex situations. In some cases, this method gives the worst result. Since Lane Detection is the most crucial part of an Autonomous vehicle, the algorithm used should be precise and accurate.

In order to get better segmentation of lane, the method called the Sliding Window technique is used. It starts at the bottom of the image and scans all the way to the top of the image while searching for lane pixels in a particular window. When lane pixels are encountered, the position of potential lane line are added to a separate list. The mean position of these pixels becomes the center of the next sliding window. Sometimes lanes are non-continuous lines and even in this situation the proposed algorithm should work better. Now in the empty window or window with zero pixels the position of the window is set to the previous window's position, changing only in height. Once the pixels are identified which correspond to the left and right lane lines, the polynomial line is drawn which shows the best fit line. This line represents the best estimate of the lane lines. It segments only the lane line irrespective of all situations.

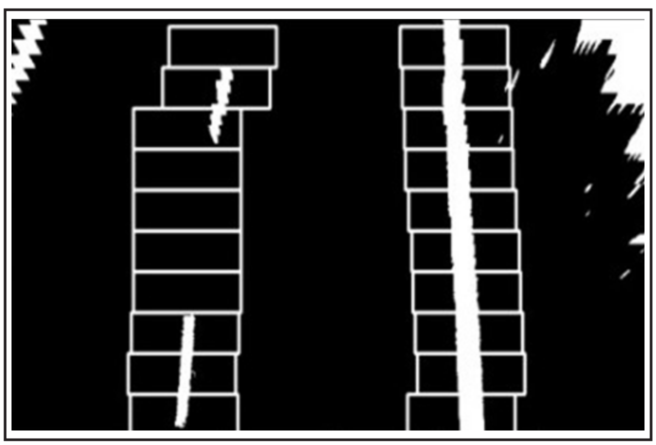

Fig. 2. Warped Frame With Sliding Windows

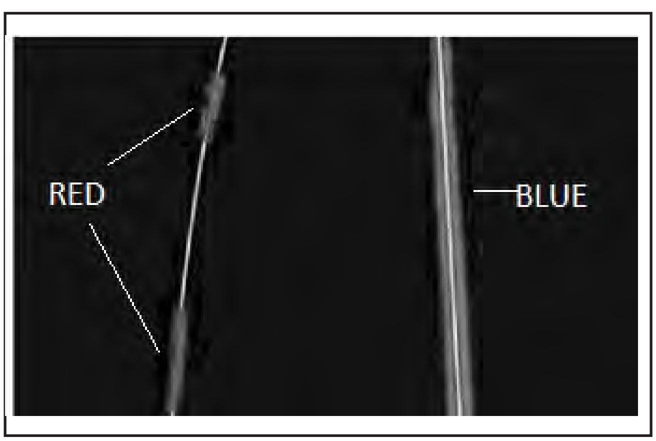

Fig. 3. Detecting Lane Lines With Sliding Windows

V. HARDWARE USED

The program is running on a single board computer Raspberry Pi 4. The Raspberry Pi used here has a Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit soc @ 1.5 GHZ processor, RAM of 4 GB, and Opengl ES 3.0 graphics. It operates at a power of 5V DC via USB-C connector and operates at a temperature range of 0 – 50° C ambient. The Webcam used for capturing the lane video is a 720P 30fps camera which has a USB connection and field of view of 60°. The 7 inch display used here has a resolution of 800 × 480 pixels. The video which is processed by the Raspberry Pi will be shown on the display which will be fitted on the car.

VI. HARDWARE IMPLEMENTATION

The camera for capturing the lane is fixed near the left sun shade of the car and another camera for capturing driver face is fixed below the steering in the speedometer dashboard. The corresponding frames will be continuously passed to the developed algorithm stored in Raspberry Pi for both LDWS and EDAS. Then the algorithm outputs as shown in the algorithm flowchart will be shown in the display.

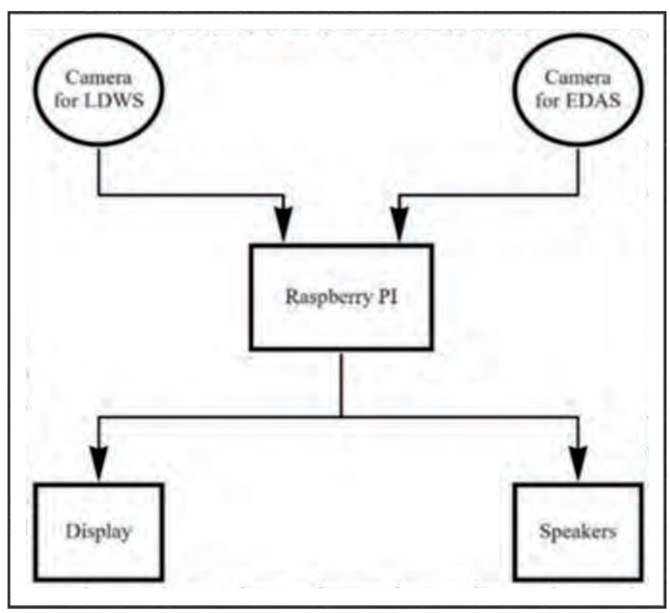

Fig. 4. Hardware Implementation

Fig. 5. EDAS Monitoring Setup

Fig. 6. LDWS Monitoring Setup

Fig. 7. Final Setup

VII. METHODOLOGY

To detect the lane, preprocessing the video is necessary. So certain techniques were performed to extract the necessary information. The flowchart given below shows the step-by-step procedure of the process.

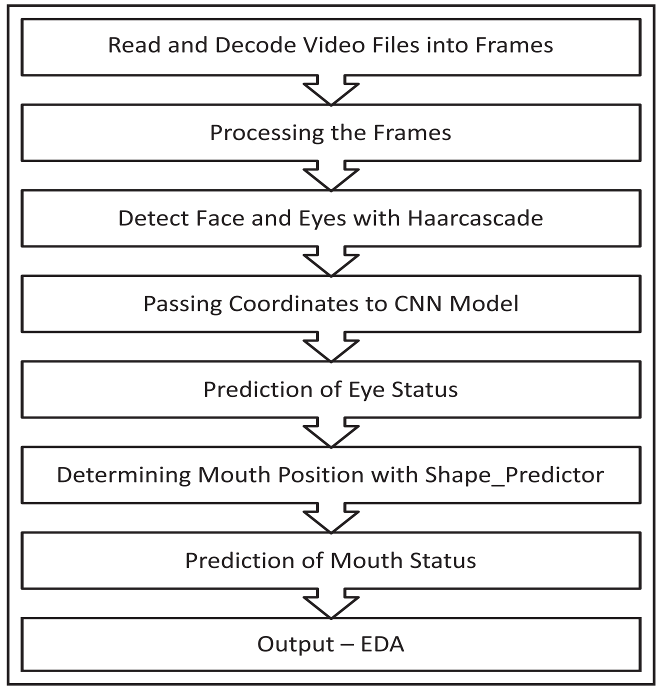

Fig. 8. Lane Departure Warning System Algorithm Flow

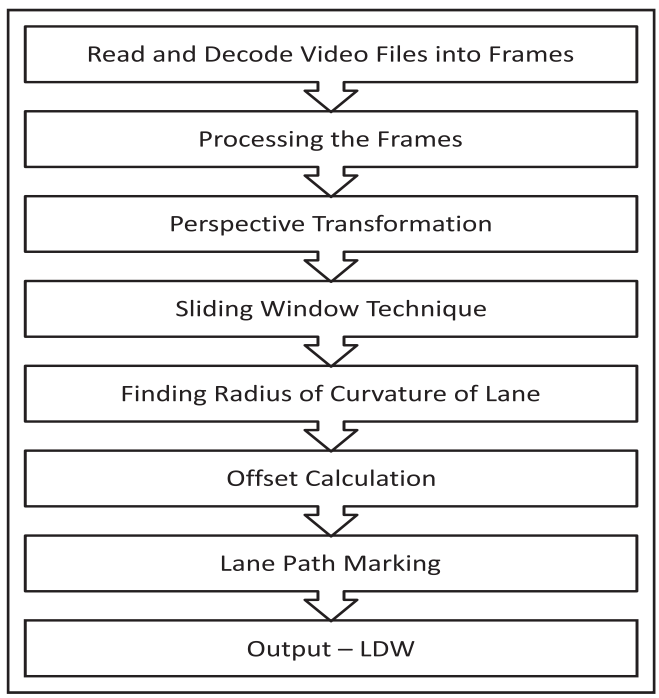

Fig. 9. Emergency Driver Assist System Algorithm Flow

For the drowsiness detector, certain procedures and techniques have to be followed in order to get the necessary information and obtain the precise output.

VIII. LANGUAGE AND LIBRARIES USED

Python 3.8.5

Python is interpreted as a general-purpose language. It is easy to learn and use. It supports various fields like computer vision, machine learning, neural networks etc. Many Libraries are available in this language and they are very easy to understand. It is a built-in, high level data structure that combines dynamic typing and dynamic binding. The main reason to use this language is that it is completely open source.

OpenCV 4.4.0.46

OpenCV is a library used for implementing computer vision concepts in the real world. It is an open-source computer vision library.

NumPy 1.18.5

NumPy is a numeric python which deals with the array where the images are stored as an array of pixels and to perform certain operations on that array to modify and extract information from those images. It has functions like linear algebra, Fourier transform, and matrices to work on images.

Matplotlib 3.3.3

Matplotlib is a data visualization library which is supported by Python. This library is the most interactive, animated visualization. The distribution of that array of pixels corresponding to particular images is visualized for better understanding.

TensorFlow 2.4.1

TensorFlow library is used in developing advanced topics like Neural Network and Machine Learning Architectures. It is developed by Google and is an open-source library. It is commonly used to load pre-trained models and is used for prediction.

Dlib 19.21.1

Dlib is an open-source C++ library which is widely used for computer vision and machine learning topics. In this paper, the dlib library is used to detect the mouth status of the driver from the input frame to detect the drowsiness of the driver.

IX. ALGORITHM IMPLEMENTATION : PROCESS OF LDWS

Initially, video has to be taken for processing using some computer vision techniques and extract the information. The processing technique includes applying thresholding and removing noise in the frame. This threshold value is fixed using the trial-and-error method. Many videos were pre-processed in order to achieve the standard threshold value. After obtaining the standard threshold value, binary threshold is applied where it outputs the frame with only black and white color. This black and white color corresponds to the pixel value of 0 and 255. Actually, there are two different types of lanes. White lane for the roads in public places and yellow lane for the roads in private places. However, both are lanes and these two types of lanes are to be included. So, the range of white color is specified and mask is applied and it gives out the white lane excluding all other unwanted things. A similar procedure is followed to get the yellow lane by specifying its range. Now both of these lanes are included in the output frame. To achieve this bitwise operation is performed. Removing noise is one of the important processing techniques because it can affect the final

Fig. 10. Masked Binary Image

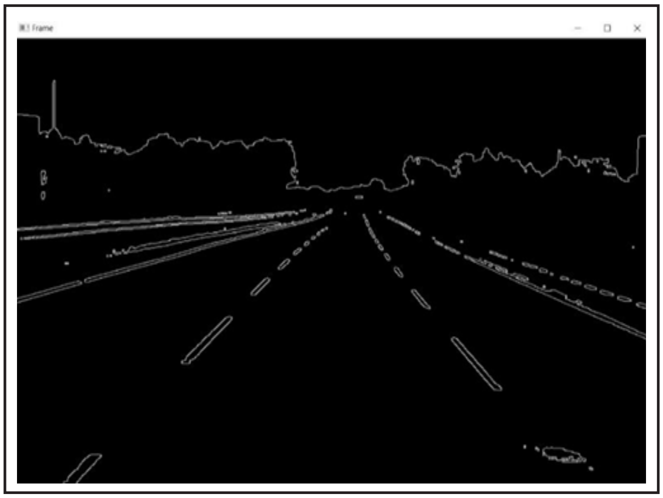

Fig. 11. Canny Edge Detection

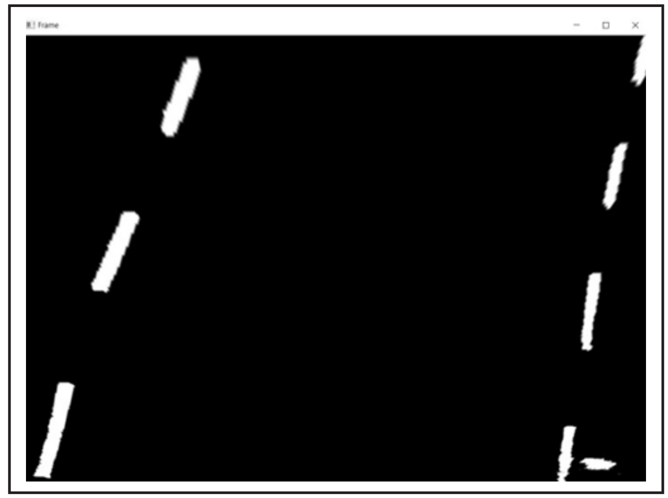

Fig. 12. Perspective Transformation

output adversely. Gaussian blur is applied to reduce the noise in the frame and gives a smooth output. Kernel size for this Gaussian blur is taken as 3x3 which is a standard kernel size. Finally, Canny edge detection is applied to the frame. It gives out the edges of the lane efficiently.

After applying some preprocessing technique, only the pathway of the lane is required to perform further operations. A pre- built function is available to transform the perspective of the frame and convert it according to the requirement. Now, the frame is changed from a human perspective to a bird-eye perspective. To perform this transformation, the frame's width and height are given as source point and the expected bird-eye view frame's width and height as a destination point. Bird-eye view could give a more efficient result because only the lane pathway is focused and processing time on some unwanted things is reduced. The main advantage of performing this transformation is that time complexity of program is reduced.

Coming to the main part of Lane detection which is applying Sliding Window Technique. Initially, the position of lane pixels is obtained using a pre-built function. These obtained positions need to be checked, whether it is a potential lane. Since the binary threshold is applied all pixel value except lane pixels are zero (0). Lane pixels correspond to some value. The size of the window is specified with some constant height and width. Now number of windows is also specified. More the windows, more accurate are the results obtained. At the same time, time consumption is also increased. Optimum value is chosen and proceeded. Then the window starts for searching the place with maximum pixel value. Here only the lane has high pixel values. While the window is searching, it encounters the lane after some time. Once it encounters, the position of that lane line is stored in an empty list. At this moment, there may be some storage problem because the list gets appended continuously with the position of lane. So, whenever a new frame starts running, the list is emptied and starts over the appending process again. Once the potential lane is detected, the rectangle is drawn on that lane and mean position of that lane is calculated. The purpose of calculating the mean position is that some lane lines are non-continuous, in such a case output polyfit line may get interrupted. When the lane is not encountered, the mean position of previous window is assumed for current window. It avoids the occurrence of error and gives a clean and smooth segmentation of lane. After encountering all the lane lines

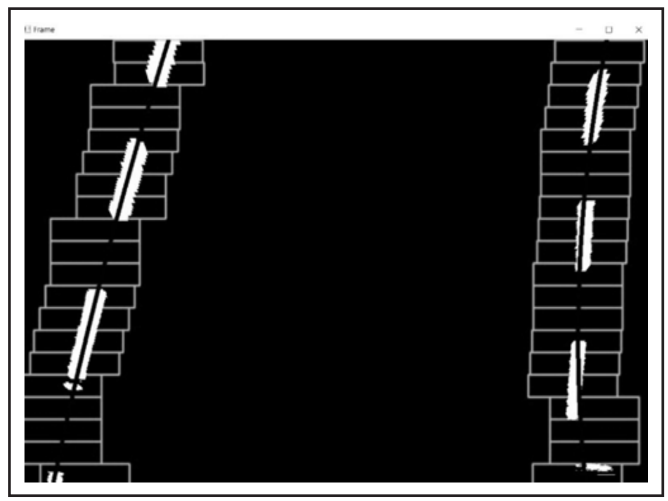

Fig. 13. Sliding Windows

from specified number of windows, polynomial function is obtained that gives the polynomial curve which fits only the lane perfectly. Using these polynomial functions and points obtained, exact lane is visualized as expected.

Radius of curvature of the lane is calculated from the polynomial functions and position of pixels obtained from the sliding window technique. The most important step is to convert all the pixels value into real-world values. Generally, the width of the lane is 3.7 meters and it is converted. There is a specific formula to calculate the radius of curvature. With the calculated value, it is more certain to say whether there is a left turn or right turn in the road. This data will be very useful in the auto-steering system and self-driving vehicle. After performing the sliding window method, many pieces of information are extracted and using this information some important parameters like radius of curvature and offset value of vehicle can be calculated for easy convenience.

$$R = \frac{\left(1 + \left(\frac{dy}{ax}\right)^2\right)^{\frac{3}{2}}}{\left|\frac{d^2y}{dx}\right|}$$

After calculating the radius of curvature, the algorithm will be able to say whether there is a left turn or a right turn. Now all the calculated values are in the algorithm and it will be displayed in the real output frame to make it visible to the user. In this function the original frame and binary image are taken as argument. Then the left and right lane points are to be arranged properly so that it will match with the original frame. Finally, both the points are stacked horizontally into a single list. Since the task is developing a lane departure warning system, it has to alert the driver if the vehicle goes or moves away from the offset with some tolerance. The algorithm is developed with offset value and fillpoly function which fills the space between left and right lane in different colors to distinguish when vehicle maintains the offset and deviates from the offset. Finally, the warp perspective transformation of the frame is done and weight is added to maintain pixel clarity in case of perspective transformation. Then the final output image is returned. The algorithm will have the left and right lane values and its positions. This parameter is used to calculate the centre of both the lanes and find the offset values from the centre of the lane. Adding text function is used to print all the calculated values in the screen.

This LDWS system works well and gives better result. Also, it warns in form of text message when the vehicles move away from the centre of lane. In Fig. 15, the vehicle moves off from the offset. As a result, pathway of line becomes red colour and warning message in displayed.

X. ALGORITHM IMPLEMENTATION PROCESS OF EDAS

For drowsiness detector, the identification of the mouth and eyes landmark positions and its movements has to be

Fig. 14. Real Time Output (LDWS)

Fig. 15. Vehicle Moving Away from Lane

taken in order to provide the alert. The first step in the detection of status of eyes and mouth is cropping out the region of interest. For this process, Haar cascade is used to select the region of interest. The Haar cascade files which are used here are the front face, right eye, and the left eye. Initially, input is passed through front face file and face area is segmented and then this segmented image is passed through left and right eye files to segment them individually. The eye part will be sent into the pretrained eye classifier model which will process it and return the status of the eyes as closed/open. Since it is a classifier, it will return a variable. If it is 1 then eye is open and if it is 0 then eye is closed. The eye aspect ratio is tested to see if it is less than the "blink/closed" eye ratio. If the eye aspect ratio is less, then the algorithm will increase the score (the total number of consecutive frames where the person has had their eyes closed) till the person opens up his eyes and the ratio is high. Here the algorithm is written in the assumption that the person is starting to doze off if the score exceeds the threshold value (i.e. 8). Once the score exceeds the threshold value the alarm will go off. For the alarm sound this algorithm consists of a predefined module named *pygame*. The *pygame* library will create a sound from the object 'alarm.wav' file. For identifying the mouth status, dlib library is used. It contains 68 landmarks in the face which are coordinates in the form of (x,y).

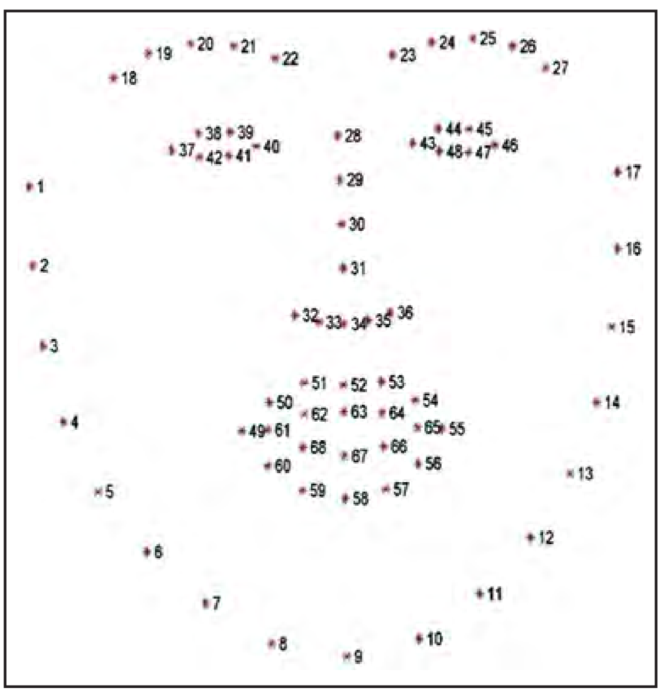

Fig. 16. Facial Landmarks

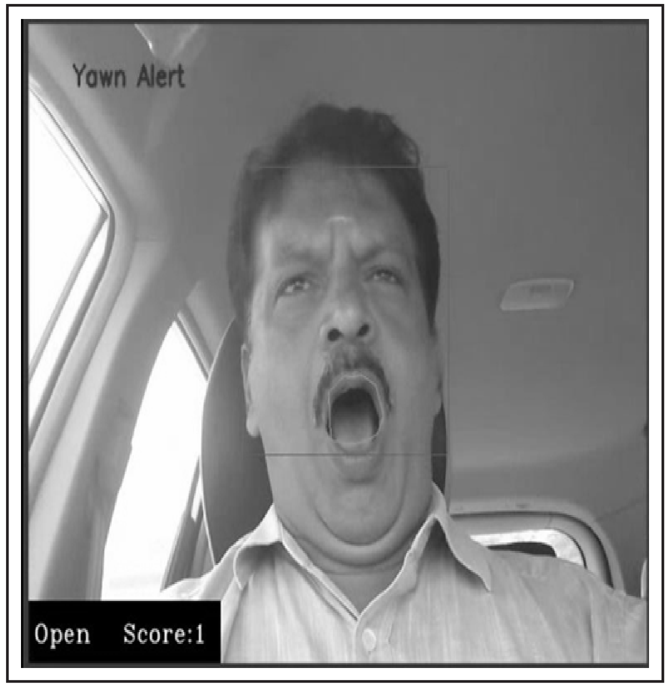

Fig. 17. Yawn Detection

Fig. 18. Drowsiness Detection

The coordinates of upper and lower lip will be obtained from the dlib landmarks. Mean of both top lip and bottom lip and the distance between the top lip and bottom lip will be calculated. The entire outer boundary of the lip is obtained. Then the contours are drawn around the detected lip pixels.

There are also two constants included in our algorithm: score lip and yawn threshold. Score lip is for keeping track of the status of the lips as open/closed and yawn threshold is for the number of consecutive frames the mouth was being opened. The score lip value must be below the threshold, if it is above the threshold value then the yawn warning message is shown and that is the output of the mouth status which will be displayed on the screen.

XI. EFFICIENCY OF THE PROPOSED SOLUTION

Comparing both the methods, Sliding Window Method seems to be more accurate and gives desired output. Sometimes, Hough transform misclassifies the lane lines. It can give misleading results when objects happen to be aligned by chance. Detected lines are infinite lines described by their (m,c) values, rather than finite lines with defined endpoints. Finally, the Sliding window technique is chosen for the implementation of LDWS. On repeating this over and over until the top of the lane is reached to get the better position of potential lane. So, this technique is expected to give a better outcome compared to other methods.

XII. CONCLUSION

This paper provides an understanding of how a self-driving car operates and also gives an in-depth knowledge of the field for the reader. LDWS and EDAS are put into effect. The most significant takeaway was that the ideas are turned into a real- time solution. In future this project can be brought upto next level of ADAS where if any of the situations like deviation of the vehicle from the lane or drowsiness of the driver is detected, the vehicle will take control and perform necessary actions such as going back to lane or stopping the vehicle and prevent accidents effectively.

AUTHORS' CONTRIBUTION

Sanjay S. conceived the idea to undertake the Advanced Driver Assist System. Literature survey was done by all the members. The algorithm for the Lane Departure Warning system was developed by Sanjay S, Vasanthakumar S. Lane Departure warning system is a complete Computer Vision Algorithm. Emergency Driver Assist System Algorithm was developed by Saravana Kumar K., Mohamed Rifayee Hussain Z. This algorithm has some Deep learning concepts and Computer Vision concepts. The final integration of the algorithms and their implementation in the embedded system was done by Dinesh Kumar S.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non – financial interest in the subject matter, or materials discussed in this manuscript.

FUNDING ACKNOWLEDGMENT

The authors received no financial support for the research, authorship, and/or for the publication of this article.

REFERENCES

[1] P. R. Nagrale and V. P. Kshirsagar, “Lane detection with lane departure warning system,” *Int. J. of Scientific Develop. and Res.,* vol. 4, no. 8, pp. 14 – 17, 2019. [Online]. Available: https://www.ijsdr.org/papers/IJSDR1908003.pdf

[2] F. Habeeb, K. N. Kunan and N. Tuturaja, “A study on lane departure warning system and object detection on roads for forward collision avoidance,” *Int. J. of Innovative Res. in Elect., Electron., Instrumentation and Control Eng.,* vol. 6, no. 5, pp. 51 – 54, 2018. [Online]. Available: https://ijireeice.com/wp-content/uploads/2018/05/IJIREEICE-11.pdf

[3] M. J. Flores, J. M. Armingol, and A. Escalera, “Real-time drowsiness detection system for an intelligent vehicle,” *2008 IEEE Intelligent Vehicles Symp.,* vol. 2008, pp. 637 – 642, June 2008. [Online]. Available: https://doi.org/10.1109/IVS.2008.4621125

[4] N. Alioua, A. Amine, and M. Rziza “Driver’s fatigue detection based on yawning extraction,” *Int. J. of Veh. Technol.,* vol. 2014, pp. 1–7. [Online]. Available: https://doi.org/10.1155/2014/678786

[5] B. Akrout and W. Mahdi, “Visual based approach for drowsiness detection,” vol. 2013, pp. 1324 – 1329, *IEEE Intelligent Vehicles Symp,*. June 2013. [Online]. Available: https://doi.org/10.1109/IVS.2013.6629650

About the Authors

Sanjay S. is 4th year student at Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore, India. His research interests include Computer vision, Hydraulics and Pneumatics, and Deep Learning.

Vasanthakumar S. is pursuing Robotics and Automation Engineering from PSG College of Technology, Coimbatore. His work focuses on developing novel solutions using Machine Learning and Computer Vision to solve real world problems.

Dinesh Kumar S. is a 4th year student at Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore, India. His research interests include Embedded systems and Computer Vision.

Saravana Kumar K. is pursuing degree from the PSG College of Technology as a Robotics and Automation Engineer and is predominantly working in the field of Computer Vision, and Autonomous Robots.

Mohamed Rifayee Hussain Z. is pursuing Robotics and Automation from PSG College of Technology, Coimbatore. His research interests include Computer Vision, Automation and Data analysis.

P. D. Rathika is working as Assistant Professor in the Department of Robotics and Automation Engineering, PSG College of Technology, India since 2019. Her research interests include Vision Systems, Wireless Sensor Networks, IoT, and Embedded Systems. She has a doctorate degree in the domain of Wireless Networks.