DRIVER AWAKENING SYSTEM USING RASPBERRY PI

Sakshi Rane*, Pooja Maurya*, Priyam Sharma*, Krisha Soni*, Prof. Bhavesh K. Patel*

**Computer Science Department, Laxmi Institute of Technology, Sarigam*

Abstract- In today's scenario most of the road accidents are caused due to drowsiness. Drowsiness is a state which is most commonly faced by drivers during long journeys and due to which, attentiveness is reduced and reaction time is slower. These drivers are most commonly commercial drivers, shift workers, drivers with lack of sleep, and some of them with certain medications which lead to drowsiness. To avoid such situations, we are willing to propose an idea of the Driver Awakening System (using raspberry pi 3B+) for keeping the driver awake throughout the journey. The domain aims to keep track of a driver and their attentiveness to prevent them from falling asleep. It functions to record real -time video of the driver's eye movement and that data will be used for further process to take necessary actions. Along with eye-tracking, facial movements like yawning will also be monitored. If such movements are being detected frequently then the alarm will be stimulated and will awake the driver and further if the alarm reaches its threshold then fed 3emergency contacts and owner of the vehicle will receive a notification by message. Also we put forward an application named Drive assistant application (android application using firebase for data storage) through which the owner of the vehicle can register itself and can grab details related to driver and vehicle along with location tracking of the vehicle which will provide the status of the vehicle. Hereby, we can conclude that by utilizing our proposed system, there shall be a decrease in the number of accidents caused due to drowsiness.

Keywords- DLib, Eye detection, Haar cascade algorithm, Location tracking, Open CV, Raspberry Pi 3B+, Web cam, yawning detection.

INTRODUCTION

This project can be considered as an initiative towards decreasing the number of accidents caused due to fatigue or drowsy state i.e., a state in which the driver might feel sleepy. This approach is also helpful in keeping track of the vehicle whether it is the real-time tracking of the driver or to keep a record of the source and the destination location.

This whole system can be adopted by any transport company to keep a track of the drivers can be also adopted by any family where one can be the admin and the rest of the family members can be the members registered under the admin to keep a track and record of the family, which therefore means that it can be used commercially or personally.

This system comes to the users with many unique features like registration of the owner and the driver, the record of the source and destination location, also most importantly the real-time location tracking to check the driver's location anytime, anywhere.

Our main purpose in proposing the system is to reduce the number of road accidents caused due to the fatigue state of driver. In the system, whenever a driver would be in a drowsy state, an alarm will be invoked to keep driver awake.

This system can be used by any business community or can be used for personal purposes as safety is the main concern of the project. This system will also alarmed the driver while driving even if it is not connected to the drive assistant application, which means this system would work till the raspberry pi module is installed into the vehicle.

RESEARCH

Several works have been done in the area of driver monitoring and detection systems using several different methods. Among the possible techniques, the best approach are the ones based on human physiological phenomena. These methods can be executed by measuring brain waves (EEG), heart rate (ECG), and open or closed state of the eyes[1]. The former two methods, however being more accurate are not realistic since sensing electrodes to be attached directly onto the driver's body and hence results to be annoying and distracting the driver while driving. The later method based on eye closure is well suited for real world driving conditions since it can detect the open or closed state of the eyes using a camera. Eye-tracking based fatigue detection systems have been done by examining the duration of eye closure and developing an algorithm to detect the driver's drowsiness in advance and to warn the driver by invoking alarms.

We summarize previous methods to drowsiness detection. To improve the accuracy and speed of drowsiness detection, various techniques have been proposed. Conventional approaches to drowsiness detection, followed by the latest approaches using deep learning.

PROPOSED MODEL

We have proposed our system to overcome the flaws of the existing system. Our system has features like real-time drowsiness detection using eye detection, yawning detection, and vehicle tracking. Our system is named Driver Awakening System. Our system comprises of hardware parts and the software parts. Our system's hardware part is installed in the vehicle and the software part consists of an application named Driver Assistant. The hardware part consists of a Raspberry pi and web camera. When the vehicle gets started our installed system starts working. The driver and the owner have to register first in the Driver assistant application with the required details.

The driver has to login to the application before starting the journey and then the notification would be sent to the owner about who is driving the vehicle along with the vehicle details and there starting location would also be stored in the database. When the vehicle is started the installed system's camera starts capturing the real-time video. This video will be the input to the raspberry pi which then converts video into the video frames and then these frames are compared to check whether the driver is in the drowsy state or not. The alarm will be invoked if the driver is in fatigue state.

If the number of times the alarm invokes crosses the threshold value then the SMS would be sent to the relative's and the owner. The driver will feed the relatives contact during registration. If the drowsiness is not detected then the system will continue the whole process. For drowsiness detection algorithms like Haar Cascade is been used. The owner can check the real-time position of the vehicle by login to the Driver Assistant application. The Owner can also outlook the details of the driver. This is the proposed system by our group and can further be extended with many updates.

HARDWARE AND SOFTWARE

- Raspberry pi: version 3 B+
- Operating System: Raspbian Os- 9 (stretch)
- Library: Open CV (version: 4.1.1)
- Back-end: Firebase (64- bit)

	Hardware Requirements	Software Requirements
Developer Requirements	Web Camera - mega pixel -Night vision camera Raspberry pi 3 -Quad core1.2ghz -1 GB ram -4 USB Port -4 pole stereo output and composite video port -CSI Camera port for connecting a camera -Micro SD port for loading your operating system and storing data LAN (Ethernet) USB power supply	Python idle IDE Windows 10 Android studio 3.5
User Requirements	A 4-wheel vehicle Android Phone Sim Card USB Cable Camera	Android version (4.0 and above)which supports the connection with Raspberry Pi Drive assistant application

FLOW

In the proposed system, the raspberry pi and android application works together. The following Flow charts depicts the flow of the whole system where, firstly the driver registers himself, logs in when they are driving. As soon as the system is turned on, it starts detecting the drowsiness of the driver and when detected, an alarm is being invoked, alert message is being sent to the emergency contacts and the owner and also if not detected the system continues to monitor the driver's state. At the same time if an owner wishes to check the current location, he/she can login into the app and can see their current location if they are logged into the system and driving. And also it is used to keep a track of the driver's record with the help of cloud firebase. The flow to the system is being displayed in the following figure.

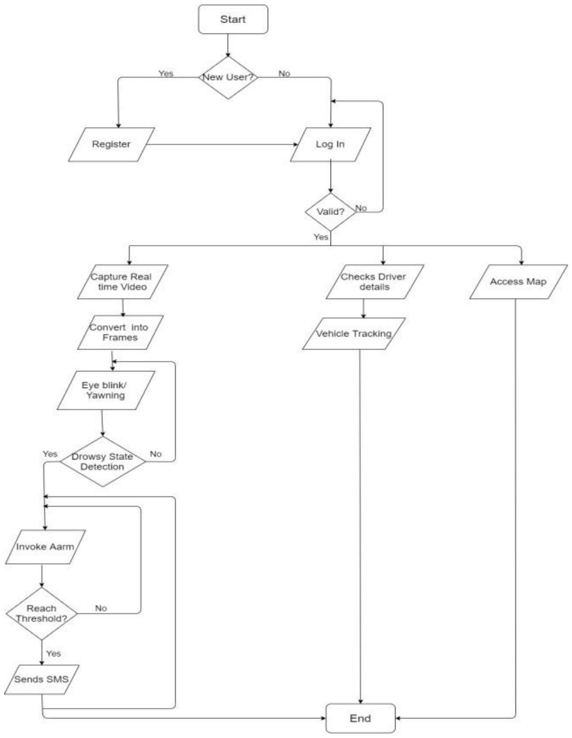


Figure 1: *The flow of the system*

RASPBERRY PI : Here, firstly lets discuss the role of raspberry pi and how does it works- the raspberry pi is being connected to a web camera, it can also be connected to a pi camera. The camera sends the frames to the raspberry pi where these frames are being processed using a python program. For this program we need several libraries for face detection and for connecting it to the database. The face detection includes eye blink detection and yawn detection. These detections were done with the help of classifiers in Open CV for eye blink and yawn detection by capturing the driver through the camera and then comparing the previous frame to the current frame. In Open CV the Haar cascade detects the face in the image by finding the bounding box with (x,y) co-ordinates of the face. These co-ordinates of the face are then used to facial landmarks by applying the dlib's facial predictor and marks the face with 68 salient points.

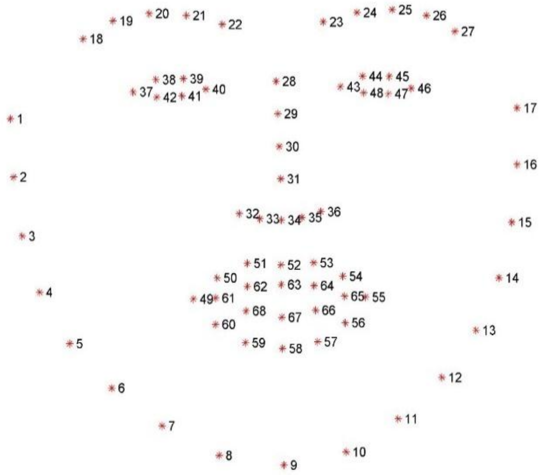


Figure 2: *68 Facial landmarks co-ordinates.* [2]

These facial landmarks are being used in this module for detecting the eyes and the mouth for eye blink detection and yawn detection. This can be done using the Python's simple array slices i.e., for example, after giving the facial landmarks of an eye, we just need to apply the Eye Aspect Ratio (EAR) algorithm.

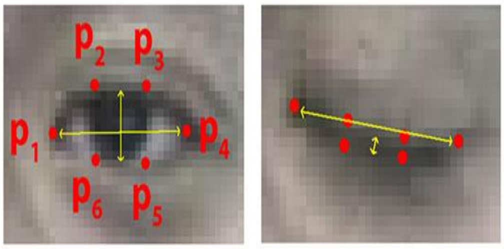
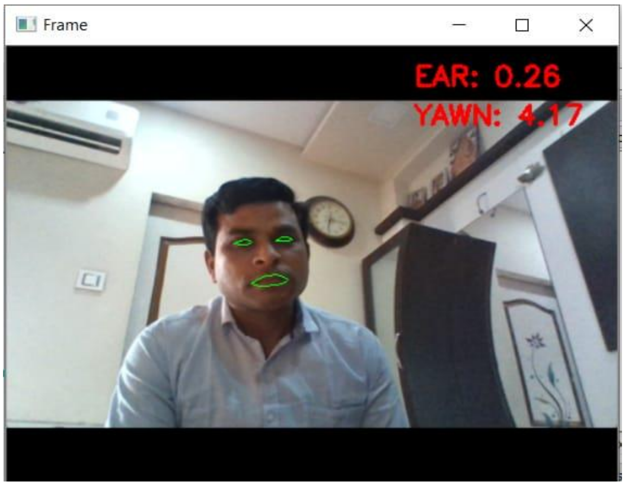


Figure 3: *Left - Landmarks when the eyes are open. Right - Landmarks when the eyes are closed.* [3]

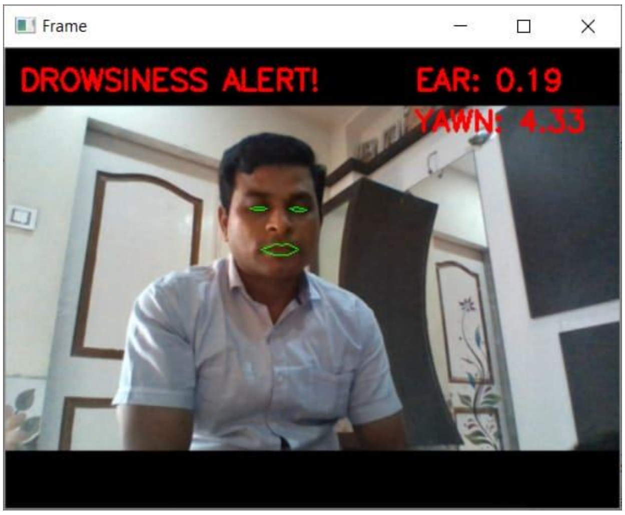
The above figure depicts the landmarks of eyes used for calculating the EAR Ratio when there is any movement of the eyelids. This concept is applied for Eye Blink Detection. In the EAR algorithm, it counts the number of open eyed frames and calculates the criteria for detection of drowsiness. If the criteria are satisfied, then an alarm is being invoked.

Here is an example of eye blink detection using the above mentioned methodology displayed with the help of a live monitor display when the program is being executed. This display monitor can be used for testing purpose, later it can be programmed in such a way that whenever the system is turned on, the program is being executed automatically.



EAR: 0.26
YAWN: 4.17

Figure 4: *When the person's eyes are open: EAR-high YAWN-low*



DROWSINESS ALERT!
EAR: 0.19
YAWN: 4.33

Figure 5: *When the person's eyes are closed: EAR-low*

In the above figures 4 and 5, one can conclude that the whole eye detection depends on the EAR algorithm and whenever the EAR is low, an alert message is displayed at the monitor and an alarm is being invoked.

Similar concept works for the mouth while Yawn Detection. The facial landmarks extracts the mouth location and starts to process it for the YAWN ratio as quickly as possible in order to reduce the search region and avoid it confusing with other facial landmarks like eyebrows.

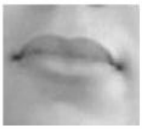
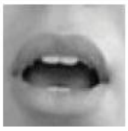
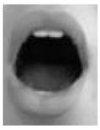
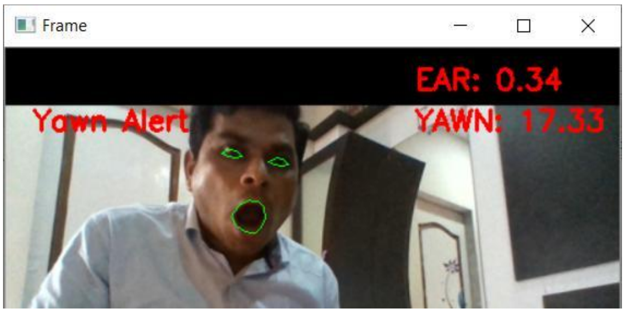
	Closed mouth	Slightly open mouth	Widely open mouth
Original image			

Figure 6: *Comparison of yawning mouth* [4]

The above image depicts different levels of yawn, like when the mouth is slightly open and when the mouth us widely open.

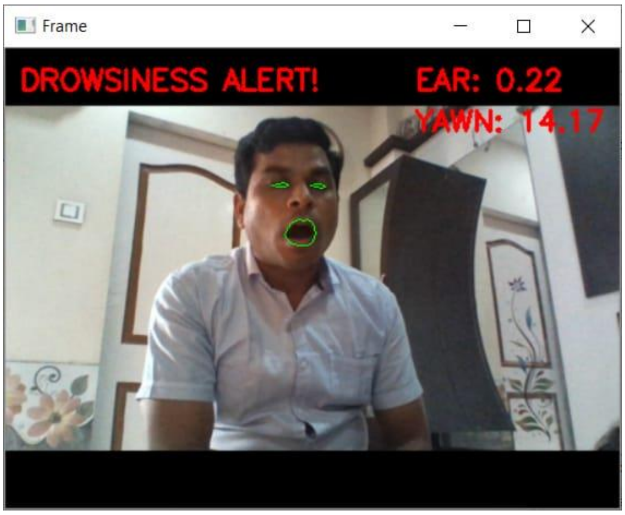


Yawn Alert
EAR: 0.34
YAWN: 17.33

Figure 7: *When the person is yawning: YAWN-high*

In the above figure 7, when the person is yawning, the YAWN ratio increases and at a certain level of the ratio, an alert message is displayed at the monitor and an alarm is being invoked.

The above mentioned both the algorithms for EAR and YAWN can also work together and can be used for the proposed domain for Drowsiness Detection.



DROWSINESS ALERT! EAR: 0.22
YAWN: 14.17

Figure 8: *When the person's eyes are closed and the mouth is open: EAR-low YAWN-high*

As depicted in the above figure 8, both the algorithms work together and their simultaneous alarms are being invoked when detected to be in drowsy state.

ANDROID APPLICATION : Moving towards the Android Application's role, here we are using and android application as it is a very convenient way of interfacing the database to any physical system. Here, it is given the name "Driver Assistant" as it acts as an assistant to the driver by taking the driver's details, allowing them to login and feed the source and the destination location, allow access to google map and keep a record of them for the owner.

DRIVER: The driver is able to register himself with the help of his contact number and by filling the personal, address, license, vehicle details, emergency contact details. After registering themselves they can login with the help of their contact number with an OTP generated by the API through the application. They can also be able to navigate to the destination by accessing to the google map through the application.

OWNER: The owner is able to login to the application same as the driver i.e., through OTP generation. The owner after logging in is able to check the driver's profile and current location of them while driving. The owner is also able to access to the record of the drivers- combined and individually.

On the other hand both, the owner and driver are able to update their profile. The following figures are some of the important activities of Driver Assistant Application.

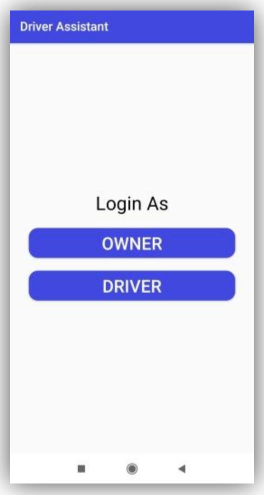
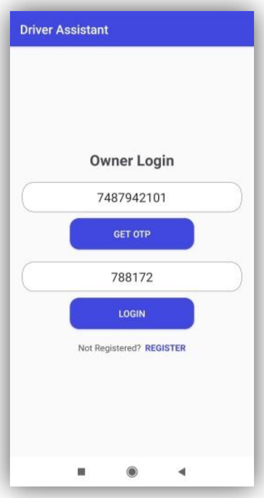


Figure 9. *Choice Activity*

Description: Here, the owner/driver will select whether to login as owner or driver.

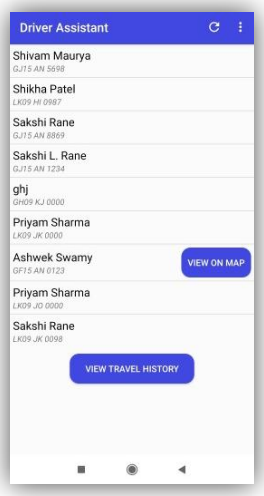


7487942101

788172

Figure 10. *Login Activity*

Description: This is the login activity of the owner after registering. Same is for the Driver login activity.



Driver Assistant

Shivam Maurya
GJ15 AN 5698

Shikha Patel
LK09 HI 0987

Sakshi Rane
GJ15 AN 8869

Sakshi L. Rane
GJ15 AN 1234

ghj
GH09 KJ 0000

Priyam Sharma
LK09 JK 0000

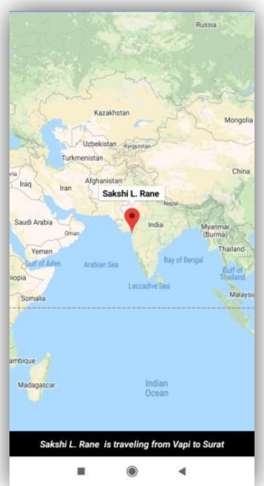
Ashwek Swamy
GF15 AN 0123

Priyam Sharma
LK09 JO 0000

Sakshi Rane
LK09 JK 0098

Figure 11. *Owner Dashboard*

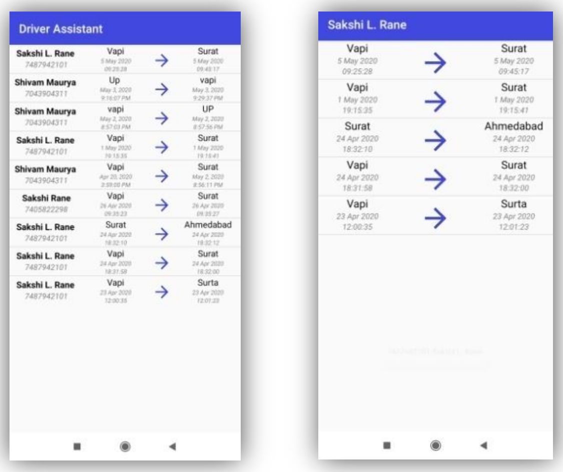
Description: This activity is displayed to the owner after logging in to the application.



Sakshi L. Rane is traveling from Vapi to Surat

Figure 12. *Maps Activity*

Description: This activity displays the current location of the driver if he/she is logged in and has started the journey.



Driver Assistant

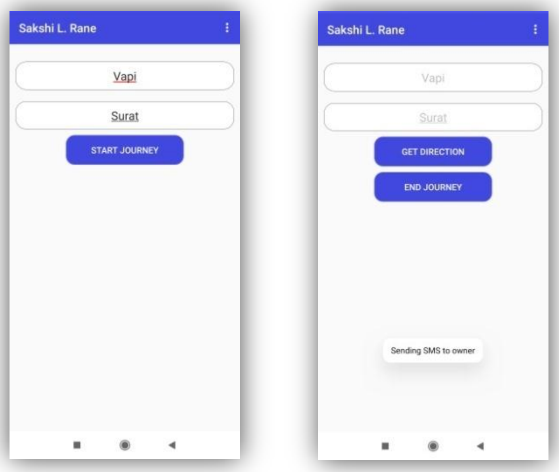
Sakshi L. Rane 7487942101	Vapi 5 May 2020 09:25:28	→	Surat 5 May 2020 09:45:17
Shivam Maurya 7043904311	Up May 3, 2020 9:16:07 PM	→	vapi May 3, 2020 9:29:37 PM
Shivam Maurya 7043904311	vapi May 2, 2020 8:57:03 PM	→	UP May 2, 2020 8:57:56 PM
Sakshi L. Rane 7487942101	Vapi 1 May 2020 19:15:35	→	Surat 1 May 2020 19:15:41
Shivam Maurya 7043904311	Vapi Apr 20, 2020 2:59:00 PM	→	Surat May 2, 2020 8:56:11 PM
Sakshi Rane 7405822298	Vapi 26 Apr 2020 09:35:23	→	Surat 26 Apr 2020 09:35:27
Sakshi L. Rane 7487942101	Surat 24 Apr 2020 18:32:10	→	Ahmedabad 24 Apr 2020 18:32:12
Sakshi L. Rane 7487942101	Vapi 24 Apr 2020 18:31:58	→	Surat 24 Apr 2020 18:32:00
Sakshi L. Rane 7487942101	Vapi 23 Apr 2020 12:00:35	→	Surta 23 Apr 2020 12:01:23

Sakshi L. Rane

Vapi 5 May 2020 09:25:28	→	Surat 5 May 2020 09:45:17
Vapi 1 May 2020 19:15:35	→	Surat 1 May 2020 19:15:41
Surat 24 Apr 2020 18:32:10	→	Ahmedabad 24 Apr 2020 18:32:12
Vapi 24 Apr 2020 18:31:58	→	Surat 24 Apr 2020 18:32:00
Vapi 23 Apr 2020 12:00:35	→	Surta 23 Apr 2020 12:01:23

Figure 13. *Record Activity*

Description: Here, two different activities with same purpose is being displayed i.e., the record of the driver as follows : 1) Common record 2) Individual record



Sakshi L. Rane

Vapi

Surat

Sakshi L. Rane

Vapi

Surat

Sending SMS to owner

Figure 14. *Driver Dashboard*

Description: This activity is displayed to the driver after logging in to the application.

In the proposed system, the above mentioned all the activities are the most important ones. The application also includes the registration pages for the driver and the owner for personal details, license details, vehicle details and the relative details. Also displaying the driver profile to the owner and the driver himself, updating the details entered and also deleting the account created by the driver.

RESULTS

The proposed drowsiness detection system detects the drowsiness of the driver through eye and yawn detection. The detection system differentiates the normal eye blink and drowsiness. The system is portable and can be easily equipped with any vehicle. And also most importantly, the system is capable of detecting drowsiness despite the spectacles worn by the driver. It also detects if the person yawns by calculating the distance between upper lips and lower lips. The system takes action if drowsiness is detected by invoking the alarm and sending messages to emergency contacts and the vehicle owner with the driver's current location and details. Thus, a small initiative has been taken to avoid the accidents caused due to drowsiness.

CONCLUSION

The system's visual definition is to refrain from accidents caused due to drowsiness. The drowsiness detection system developed here is capable of detecting drowsiness rapidly. The system can differentiate normal eye blink, drowsiness, and yawning and can prevent the driver from entering the state of sleepiness while driving by invoking the alarm. Along with it keeping the driver awake and alert throughout the journey without causing any mishap. Also endeavouring to save the lives of the civilians around and the driver. The system works well even in case of drivers wearing spectacles and under low light conditions also. Thus, helping to promote the reduction of accidents caused due to drowsiness.

AUTHORS

First Author – Sakshi L Rane, BE (C.S.E), Laxmi Institute Of Technology

Second Author – Priyam K Sharma, B.E (C.S.E), Laxmi Institute of Technology

Third Author – Pooja R Maurya, B.E (C.S.E), Laxmi Institute of Technology

Fourth Author – Krisha K Soni, B.E (C.S.E), Laxmi Institute of Technology

Fifth Author - Proff. Bhavesh K. Patel, Laxmi Institute of Technology

REFERENCES

1. https://www.hindawi.com/journals/cin/2019/4721863/
2. https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
3. https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/
4. https://www.hindawi.com/journals/ijvt/2014/678786/