17IE4048 – PROJECT-1

PROJECT BASED REPORT

ON

DRIVER DROWSINESS DETECTION

submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

ALLA CHANDANA (170030028)

TETALLI BHAVANA REDDY (170031276)

YEMMIREDDY VAISHNAVI BHAVYA (170031450)

DARISI SAI VENKATA SIDDARTHA (170031537)

Under the Esteemed Guidance of

Dr.Anjali Mathur ***PhD***

Associate Professor

(DEEMED TO BE UNIVERSITY)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L (Deemed to be) University

Green Fields, Vaddeswaram, Guntur District – 522 502

EXTERNAL EXAMINER

K LEF (Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

This is certify to that the project-based report entitled **“DRIVER DROWSINESS DETECTION”** is a bonafide work done and submitted by **Alla Chandana (170030028), Tetalli Bhavana Reddy (170031276), Yemmireddy Vaishnavi Bhavya (170031450), Sai venkata Siddharth Darisi (170031537)** in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, KL (Deemed to be University), Guntur District during the academic year **2020-2021**.

ALLA CHANDANA (170030028)

TETALLI BHAVANA REDDY (170031276)

YEMMIREDDY VAISHNAVI BHAVYA (170031450)

DARISI SAI VENKATA SIDDHARTH(170031537)

(2020-2021)

K LEF (Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is certify that the project based report entitled **"DRIVER DROWSINESS DETECTION"** is a bonafide work done and submitted by **Alla Chandana (170030028), Tetalli Bhavana Reddy (170031276), Yemmireddy Vaishnavi Bhavya (170031450), Sai venkata Siddharth Darisi (170031537)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, KLEF (Deemed to be University), Guntur District during the academic year **2020-2021**.

Dr. ANJALI MATHUR
FACULTY INCHARGE
DEPARTMENT OF CSE
K L (Deemed to be University)

Dr. V. HARI KIRAN
HEAD OF THE
DEPARTMENT OF CSE
K L (Deemed to be University)

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my guide **Dr. ANJALI MATHUR**, Assistant Professor, Department of Computer Science and Engineering which cannot be expressed in words for his tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. V. Hari Kiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project.

Finally, We sincerely thank my friends and classmates for their kind help and co-operation during our work.

ALLA CHANDANA (170030028)

TETALLI BHAVANA REDDY (170031276)

YEMMIREDDY VAISHNAVI BHAVYA (170031450)

DARISI SAI VENKATA SIDDHARTH (170031537)

1. ABSTRACT

Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this project, we aim to develop a prototype drowsiness detection system.

This system works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy. The system so designed is a non-intrusive real-time monitoring system.

The priority is on improving the safety of the driver without being obtrusive. In this project, the eye blink of the driver is detected. If the driver's eyes remain closed for more than a certain period of time, the driver is said to be drowsy, and an alarm is sounded. The programming for this is done in OpenCV using the Haarcascade library for the detection of facial features. Driver fatigue is one of the major causes of accidents in the world.

Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this project, we aim to develop a prototype drowsiness detection system. This system works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy. The system designed is a non-intrusive real-time monitoring system. The priority is on improving the safety of the driver without being obtrusive. In this project, the eye blink of the driver is detected. If the driver's eyes remain closed for more than a certain period, the driver is said to be drowsy, and an alarm is sounded. The programming for this is done in OpenCV using the Haarcascade library for the detection of facial features.

CONTENTS

INTRODUCTION

Driving is a complicated task. The driver's full attention is required every time a person gets behind the wheel. This task becomes draining after long commuting. It leads to a state of Drowsiness.

According to Research, Drowsiness is related to thousands of traffic accidents each year, the accidents produce approximately 50% of death or serious injuries, as they tend to be impacted at high speed because the driver who has fallen asleep cannot brake or deviate to avoid or reduce impact. To mitigate these accidents, manufacturers have developed drowsiness detection systems that recognize signs of possible Drowsiness, alerting the driver to their condition.

The objective is to build a surveillance system to the vehicular driver based on artificial vision techniques and implemented in a car in order to detect and alert when the driver has drowsiness signs. To achieve this objective, Research analyzes other works related with detecting Drowsiness in drivers, the drowsiness symptoms in vehicle drivers; we identify yawning of the driver; we identify the technical parameters and algorithms that allow for processing signals of the state of Drowsiness. In this work, we present a developed drowsiness detection algorithm, Which detects the Drowsiness of a person and notify the person.

LITERATURE SURVEY

Human Drowsiness Detection System

In this paper we deal with a real time situation where in our video gets recorded and needs processing to be done. But the processing can be done only on the image. Hence the captured image has to be divided into frames for further analyzation. In this stage of the process, we deal with identifying the face of the driver. By identifying the face of the driver we mean that detecting facial features or characters through the use of computer. The frame maybe random. Only facial related structures are identified other types of objects are ignored.

A Deep Neural Network for Real-Time Driver Drowsiness Detection

They develop a deep neural network (DNN) for detecting driver drowsiness in videos. The proposed DNN model that receives driver's faces extracted from video frames as inputs consists of three components - a convolutional neural network (CNN), a convolutional control gate-based recurrent neural network (ConvCGRNN), and a voting layer.
The CNN is to learn facial representations from global faces which are then fed to the ConvCGRNN to learn their temporal dependencies. The voting layer works like an ensemble of many sub-classifiers to predict drowsiness state.
Experimental results on the NTHU-DDD dataset show that our model not only achieve a competitive accuracy of 84.81% without any postprocessing but it can work in real-time with a high speed of about 100 fps.

Efficient Driver Fatigue Detection and Alerting System

Face detection and tracking are important in many computer vision applications including activity recognition. Face detection done through an image or video. There are various methods for face detection, artificial neural network method, template matching method, Voila and Jones algorithm , AdaBoost face detection algorithm. Therein, skin color detection method is useful when it appears to multi-face detection and tracking. Systems based on color can recognize human faces from different visual angles, but this method adapts to color image and cannot be used in night mode.

Viola and Jones came up with the cascade of the Haar classifiers, which increases the computational .We modified Viola and Jones algorithm based on characteristics of FPGA. First, we used 16 classifiers as parallel computing tool. Each stage has 16 integer multiple of the number of classifiers. Also, the output passing rate of Bayesian training algorithm is modified to insure the overall detection rate and false alarm rate remain un-changed after modified the number of classifiers. The eventually trained Haar classifier contains 40 stages, 2192 classifiers and 4680 features.

Only the region of the picture through all of the stages is considered as human face region, and each stage contains 16 integer multiple of the number of classifiers, paper. We designed to simultaneously process the computing of 16 classifiers, and could obtain a speed 16 times faster than the traditional computing process. Because each stage declines regions that don't contain face area, declined regions cannot enter the next stage. The pass rate of the first stage in this design is 20%, while traditional pass rate is 50%. Because most of the computation is centralized on the first stage, as long as image region is blocked by the first stage, then there is no need to initialize parameters for the second stage.

A Driver Face Monitoring System for Fatigue and Distraction Detection

Performing the face detection algorithm for all frames is computationally complex. Therefore, after face detection in the first frame, face tracking algorithms are used to track driver face in the next frames unless the face is lost. Therefore, we use an auxiliary variable denoted by *sw* for determination of face tracking status in Figure 1. If *sw* is 0, the face is lost, and face detection algorithm must be performed to localize the driver face. In contrast, if *sw* is 1, it shows that face is tracked successfully by face tracking method. For system initialization, *sw* is 0. It means that the system must perform face detection algorithm for first frame.

We used Haar-like features and adaptive boosting method proposed by Viola and Jones [14] for face detection. Face detection algorithm was trained by about 3000 faces and about 300000 nonfaces. For face tracking, full search method is used to find the driver face image in the new frame. The search region is around the center of face image in the last frame which the size of search region is changed according to the size of face image (1.5 times bigger than the size of face image). Then, correlation coefficient between the face image and the subwindows of search region is used as the matching criteria.

Driver drowsiness detection using ANN image processing

Three of these methods are based on EEG (Electroencephalography) and EOG(Electrooculography) signals measurement and on the eye state (closed or opened) image classification. The EEG method monitors the brain activity through a sensor placed on a specific part of the scalp, the EOG method tracks the eye movements by measuring the signals from the muscles which are acting on the eye and the eye image analysis can monitor the opened or the closed state of the eye. The authors discussed about the development of such a system. A scheme of the system presented. Each of the methods used in the system has its advantages and disadvantages. For example, the EEG and EOG sensors, electrodes which have to be fixed with a conductive gel and in most devices must transmit the signal by wire, present a major discomfort. Important research in the field of advanced materials and MEMS technology may solve these problems, as for example the use of dry electrodes for EEG.

A Survey Paper On Drowsiness Detection & Alarm System for Drivers

In image processing based techniques, drivers face images are used for processing so that one can find its states. From the face image one can see that driver is awake or sleeping. Using same images, they can define drowsiness of driver because in face image if driver is sleeping or dozing then his/her eyes are closed in image. And other symptoms of drowsiness can also detected from the face image. We can classify these techniques in three sub-categories.

In this eye blinking rate and eye closure duration is measured to detect driver’s drowsiness. Because when driver felt sleepy at that time his/her eye blinking and gaze between eyelids are different from normal situations so they easily detect drowsiness. In this system the position of irises and eye states are monitored through time to estimate eye blinking frequency and eye close duration. And in this type of system uses a remotely placed camera to acquire video and computer vision methods are then applied to sequentially localize face, eyes and eyelids positions to measure ratio of closure . Using these eyes closure and blinking ratio one can detect drowsiness of driver.

Yawn is one of the symptoms of fatigue. The yawn is assumed to be modeled with a large vertical mouth opening. Mouth is wide open is larger in yawning compared to speaking. Using face tracking and then mouth tracking one can detect yawn. In paper , they detect yawning based on opening rate of mouth and the amount changes in mouth contour .

THEORETICAL ANALYSIS

3.1:Face Detection

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from the sum of pixels under the black rectangle.

Drowsiness of a person can be measured by the extended period of time for which his/her eyes are in a closed state. In our system, primary attention is given to the faster detection and processing of data. The number of frames for which eyes are closed is monitored. If the number of frames exceeds a certain value. Then a warning message is generated on display showing that the driver is feeling drowsy.

In our algorithm, first, the image is acquired by the webcam for processing. Then we use the Haarcascade file face.xml to search and detect the faces in each individual frame. If no face is detected, then another frame is acquired. If a face is detected, then a region of interest in marked within the face. This region of interest contains the eyes. Defining a region of interest significantly reduces the computational requirements of the system. After that the eyes are detected from the region of interest by using Haarcascade_eye.xml. If an eye is detected, then there is no blink and the blink counter K is set to ‘0’. If the eyes are closed in a Particular frame, then the blink counter is incremented, and a blink is detected. When the eyes are closed for more than 4 frames, then it is deducible that the driver is feeling drowsy. Hence drowsiness is detected, and an alarm sounded. After that, the whole process is repeated as long as the driver is driving the car.

CvSeq *eyes = cvHaarDetectObjects(img,cascade,storage,1.1,5,0,cvSize(10, 5));

It is the same function as that of the face detection .Here eye cascade is used and minimum size of the object is decreased and optimized to detect eyes of various sizes in the image.
As described earlier cvRectangle() is used to draw rectangle for sequences of eyes detected in a given frame . then cvResetImageROI(img) is used to reset the ROI so that the whole image can be used to be displayed in a window using cvShowImage("video", img) function.
Since the cascade is constructed for only open eyes ,so when eyes are closed even for

momentarily they are not detected. The closing state of eyes are detected as blink and if the closing state continues for more than 7 consecutive frames then drowsiness condition is displayed using cvPutText() function . We can also use batch file command system(―abc.mp3‖) to play any audio file at the instant drowsiness is detected .
The detection process is continued until a key is pressed .When the key is pressed the program stops executing the detection function and stops capture form camera, releases memory and closes video window using the functions given below.

```
cv.ReleaseCapture( &capture );
cv.DestroyWindow( "video" );
cv.ReleaseMemStorage( &storage );
```

3.2.Image acquisition:

The function cvCaptureFromCAM allocates and initialized the CvCapture structure for reading a video stream from the camera.

CvCapture* cvCaptureFromCAM(int index);
Index of the camera to be used. If there is only one camera or it does not matter what camera to use , -1 may be passed. cvSetCaptureProperty Sets camera properties For example,

```
cvSetCaptureProperty( capture, CV_CAP_PROP_FRAME_WIDTH, 280 );
cvSetCaptureProperty( capture, CV_CAP_PROP_FRAME_HEIGHT, 220 );
```

The function cvQueryFrame() grabs a frame from a camera or video file, decompresses it and returns it. This function is just a combination of GrabFrame and RetrieveFrame, but in one call. The returned image should not be released or modified by the user. In the event of an error, the return value may be NULL.

Set Image Region of interest:-

```
cvSetImageROI(
img, /* the source image */
cvRect(
face->x, /* x = start from leftmost */
face->y + (face->height)/5, /* y = a few pixels from the top */
face->width, /* width = same width with the face */
(face->height)/3 /* height = 1/2 of face height */
)
```

It sets the ROI for the corresponding image. We have taken from the left most point in the face to a few pixels from the top to half of face height. Width of the region is same as that of the face.
The rectangular region is now used to get eyes.

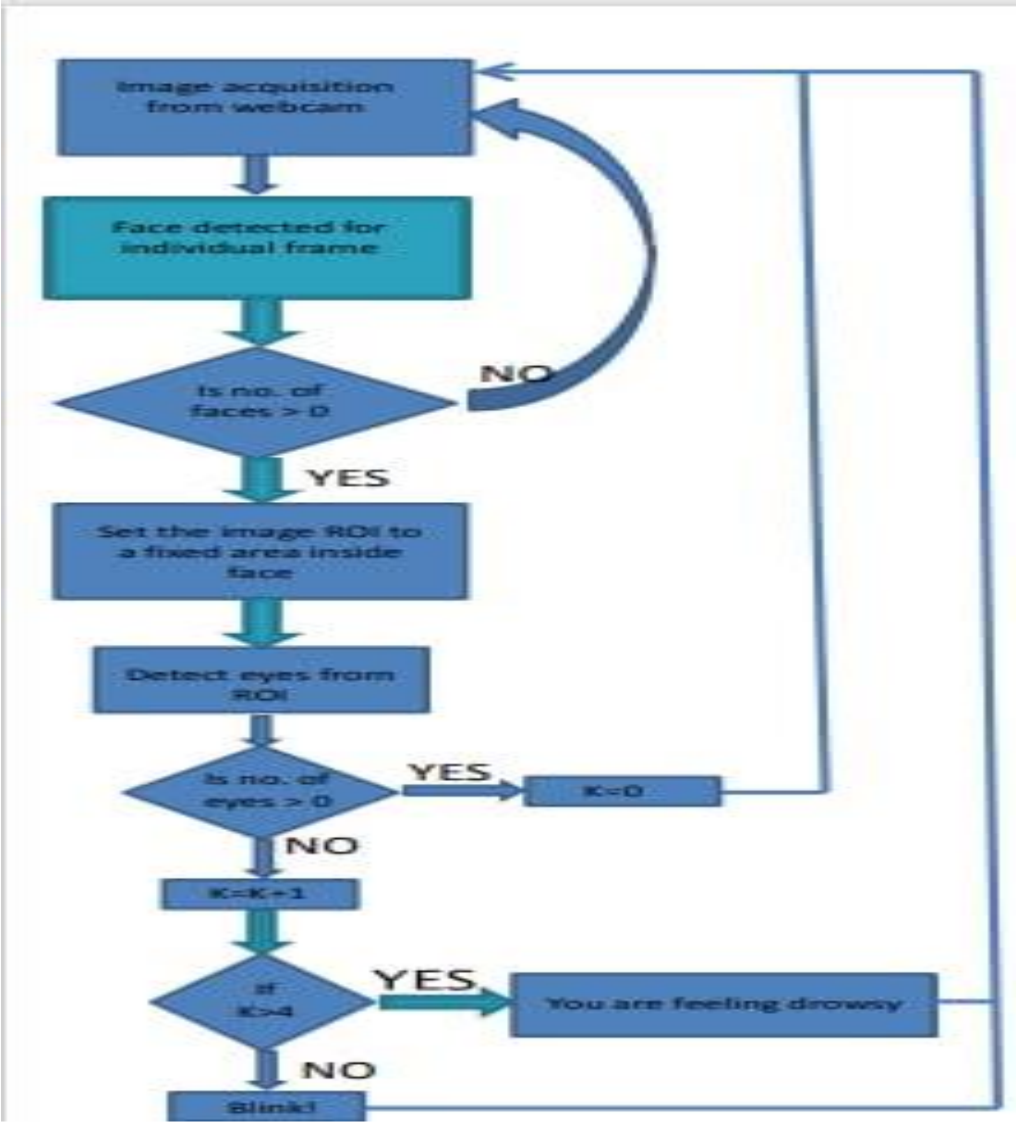
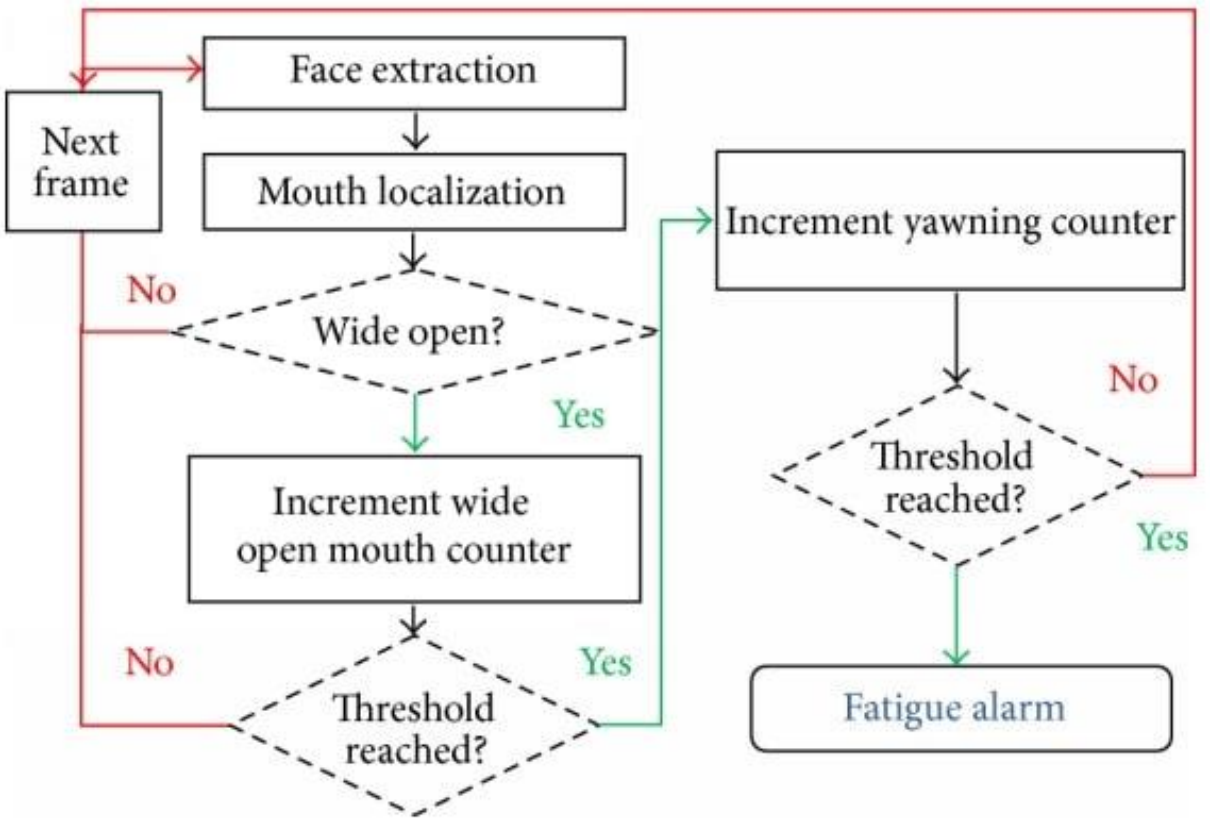


Image acquisition from webcam
Face detected for individual frame
Is no. of faces > 0
NO
YES
Set the image ROI to a fixed area inside face
Detect eyes from ROI
Is no. of eyes > 0
YES
K=0
NO
K=K+1
if K>4
YES
You are feeling drowsy
NO
Blink!

3.2.Yawn Detection:

Driver's fatigue is characterized by a high yawning frequency. If the mouth is widely open, the counter of consecutive wide open mouths is incremented. When the wide opening lasts more than 5 seconds, we consider that we have a yawning and the corresponding counter is incremented. Once this last counter is important, the system indicated that the driver is suffering from fatigue by issuing a warning signal.



EXPERIMENTAL INVESTIGATIONS

4.1 EXISTING MODEL:

The driver face monitoring systems can be divided into two general categories. In one category, driver fatigue and distraction is detected only by processing of eye region. There are many researches based on this approach. The main reason of this large amount of researches is that the main symptoms of fatigue and distraction appear in the driver eyes. Moreover, the processing of the eye region instead of the processing of the face region has less computational complexity. In the other category, the symptoms of fatigue and distraction are detected not only from eyes, but also from other regions of the face and head. In this approach, in addition to processing of eye region, other symptoms including yawning and head nodding are also extracted.

Driver face monitoring system includes some main parts: (1) face detection, (2) eye detection, (3) face tracking, (4) symptom extraction, and (5) driver state estimation. These main parts are reviewed in different systems in the current section.

In the most of driver face monitoring systems, the face detection is the first part of the image processing operations. Face detection methods can be divided into two general categories [9]: (1) feature-based and (2) learning-based methods.

In the feature-based methods, the assumption is that the face in the image can be detected based on applying heuristic rules on features. These methods are usually used for detecting one face in the image. Color-based face recognition is one of the fast and common methods. In these methods, the face is detected based on the color of skin and the shape of face. Color-based face detection may be applied on different color-space including RGB , YCbCr [12], or HIS [13]. In noisy images or in the images with low illuminations, these algorithms have low accuracy.

Learning-based face detection uses statistical learning methods and training samples to learn the discriminative features. These methods benefit from statistical models and machine learning algorithms. Generally, learning-based methods have less error rates for face detection, but these methods usually have more computational complexity. Viola and Jones presented an algorithm for object detection, which is very fast and robust. This algorithm was used in for face detection.

Almost in all driver face monitoring systems, because of the importance of symptoms related to eye, the eye region is always processed for extracting the symptoms. Therefore, before the processing of eye region, eye detection is required. Eye detection methods can be divided into three general categories: (1) methods based on the imaging in the infrared spectrum, (2) feature-based methods, and (3) other methods.

One of the fast and relatively accurate methods for eye detection is the method based on the imaging in the infrared (IR) spectrum. In this method, physiological and optical properties of the eye in the IR spectrum are used. The eye pupil reflects IR beams, and it seems as a bright spot when the angle of IR source and imaging device are suitable. According to this interesting property, pupil and eye are detected. The systems proposed in [4, 18–20] used such method for eye detection.

Feature-based eye detection approach includes various methods. Image binarization [5, 21, 22] and projection [23, 24] are two feature-based eye detection methods which assume that the eye is darker than the face skin. Usually, more complicated processing is needed to detect the proper location of eyes, because these methods are simple and have high error rate.

There are few methods for eye detection based on other approaches which were used in driver face monitoring systems. In [10], a geometrical face model with some feature-based methods was used to detect eyes. In addition, some systems such as used hybrid methods for eye detection. In [15], elliptical gray-level template matching and IR imaging system were used for eye detection in day and night, respectively.

Usually, the entire image is searched for detecting the face/eye. Searching the entire image increases the computational complexity of the system. Therefore, usually after early detection of the face/eyes, in the next frames, face/eye tracking is performed. In the most of driver face monitoring systems, Kalman filter [4, 19, 25] or extended versions of Kalman filter such as Unscented Kalman Filter (UKF) [23] were used. However, in some researches, search window [18] and particle filter (PF) [26] were used for tracking.

In the driver face monitoring systems, useful symptoms for fatigue and distraction detection can be divided into three general categories:
(i) symptoms related to the eye region;
(ii) symptoms related to the mouth region;
(iii) symptoms related to the head.

The eye is the most important area of the face where the symptoms of fatigue and distraction appear in it. Therefore, many of the driver face monitoring systems detect driver fatigue and distraction only based on the symptoms extracted from the eyes. The symptoms related to eye region include PERCLOS [3, 4, 10, 15], eyelid distance [25, 27], eye-blink speed [4, 10], eye blink rate [4, 19], and gaze direction [4].

Yawning is one of the hypervigilance symptoms related to the mouth region. This symptom was extracted by detecting the open mouth in [11, 16]. These systems detect the mouth based on the colour features of the lips in the image.

Some fatigue and distraction symptoms are related to the head. These symptoms include head nodding [5, 19] and head orientation [4, 10, 19]. Head nodding can be used for fatigue detection, and head orientation can be used for both fatigue and distraction detection. Driver nodding and lack of driver attention to the road can be detected by estimating the angle of head direction.

After symptom extraction, the driver state has to be determined. The determination of the driver state is considered as a classification problem. The simplest method for detecting the driver fatigue or distraction is based on applying a threshold on extracted symptom [22].

Another method for determining the driver state is knowledge-based approaches. In a knowledge-based approach, decision making about the driver fatigue and distraction is based on the knowledge of an expert which the knowledge usually appears in the form of if-then rules. In [19, 25], fuzzy expert systems were used as a knowledge-based approach for estimating the driver state.

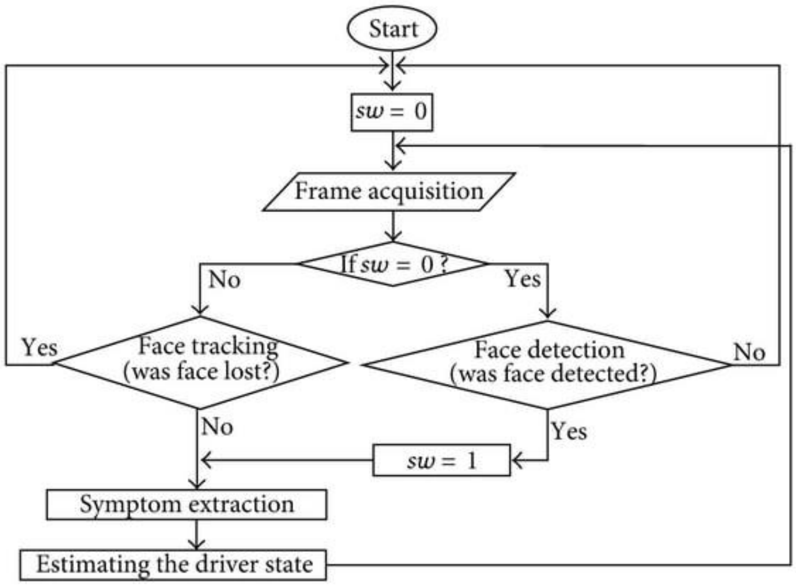
More complicated approaches such as Bayesian network [4] and nave dynamic Bayesian network [26] were used for driver state determination. These approaches are usually more accurate than threshold-based and knowledge-based approaches; however, they are more complicated.

4.2 PROPOSED MODEL:

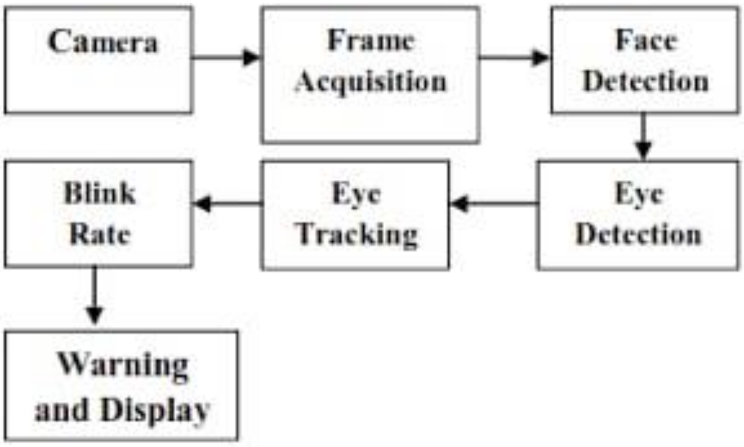
Compared with the traditional Adaboost method that trains eye classifiers to detect eye state directly, we proposed an improved strategy to detect eye state. The proposed strategy mainly includes 3 steps and 2 categories of classifiers corresponding to face and eye, respectively. Specifically, we first detect face efficiently by classifiers of both front face and deflected face. Then, the candidate region of the eye is determined according to the geometric distribution of facial organs.

Finally, trained classifiers of open eyes and closed eyes are used to detect eyes in the candidate region quickly and accurately. Table 2 shows the comparison of elapsed time for 2 methods in PC system. The shortest and longest time for processing a single frame are also presented. Concerning our method, although it needs to detect both face and eye, the average processing speed of the system is up to 20 fps. Hence, the system has a good performance of real-time.

The longest processing time of single frame is more than 2 times of the shortest one, because it takes at least 2 times to detect face and eye state due to the situation of deflected face. Concerning the traditional method, in spite of detecting eye with only one step directly, it costs more time to detect eyes. The reason is that it needs to use eye classifier searching in the whole frame instead of a small candidate region as our method.

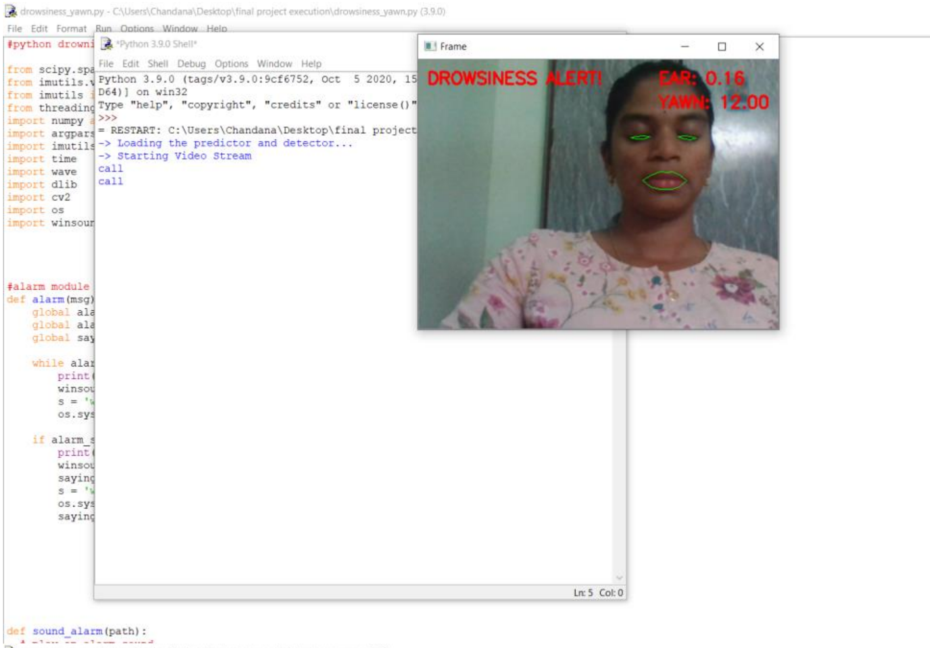


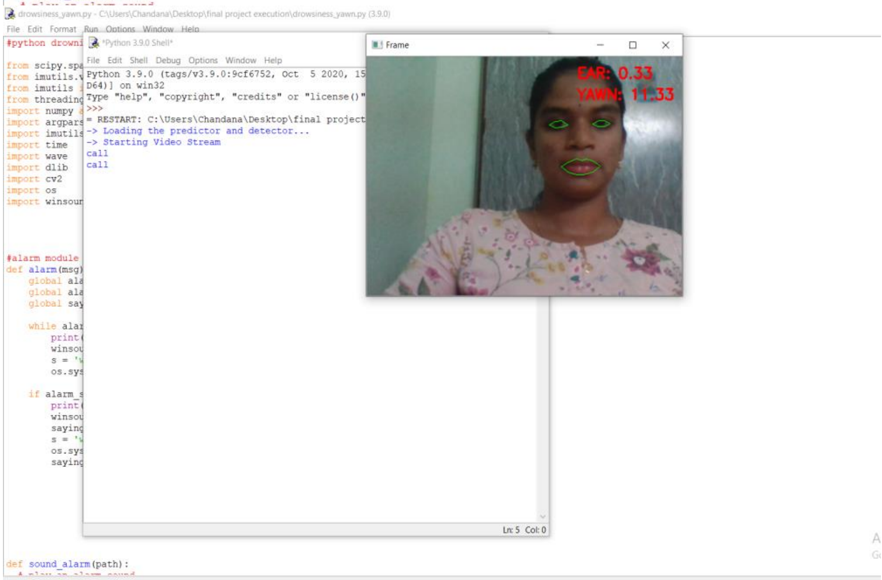
4.3 BLOCK DIAGRAM:



The block diagram consists of all the connections which are made in the execution of the project. The proposed system takes input from camera. An algorithm is developed to detect the face and eyes and subsequently track the eyes and perform calculations to determine if the driver is drowsy. The warning system issues an alarm to the driver if the above condition is true. Drowsiness during driving is a very serious problem without a doubt. The increasing numbers of deaths due to road accidents have shown drowsiness as one of the topmost reasons for the same. It thus becomes a substantial motivation to combat drowsiness among drivers, reduce accidents and save lives. There can be numerous causes to drowsiness. In this paper, we consider four parameters namely eye detection, blinking, eyelid closure and yawning

EXPERIMENTAL RESULTS

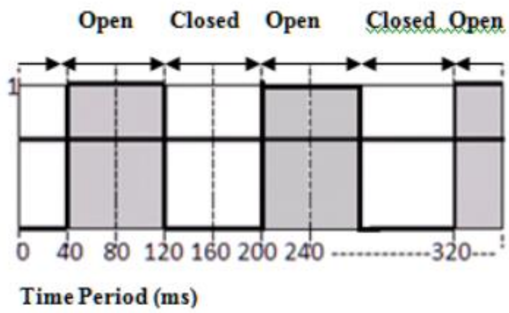




DISCUSSION OF RESULTS

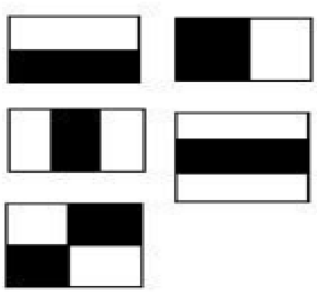
BLINK DETECTION:

When continuous frames of either a steady intensity or steady dip is not detected and rather, an alternate change of pixel intensity is observed over a period of time, it is detected as blink.



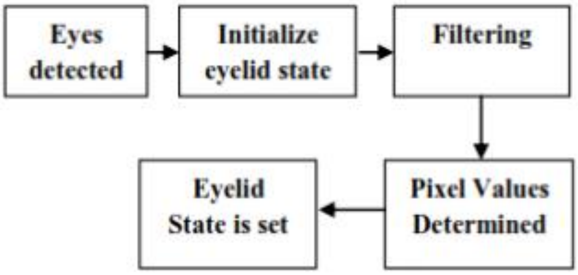
EYE DETECTION:

Once the cascade classifiers detect and pass an image detected as face, the eyes are searched for. Using the Haar Classifiers, feature points are analyzed and pixel values are determined from the passed image. A sample of Haar feature types are passed and the filter gradient calculates the pixel values of the region of interest which are the eyes here. Some of the Haar types used in searching and locating the eyes



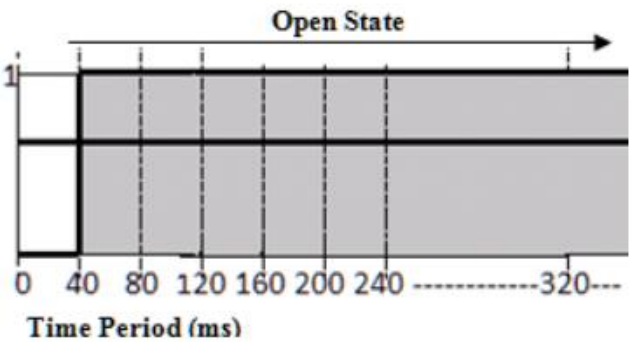
EYE CLOSURE EVALUATION:

The eye related parameters used to detect drowsiness in the proposed system are namely eyelid closure, gaze detection and blink rate monitoring. A square wave has been used to recognize the different states of the eye. A logical 1 and 0 are respectively used to signify input of frames and otherwise. The following block diagram briefly depicts the algorithm involved



GAZE:

Gaze is the phenomenon which synonymously indicates complete open state of eyes. For a period of time, if the input frames acquired records a continuous and close to constant pixel intensity for a few integral multiple of time period, a gaze is detected.



CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

We have successfully designed a prototype drowsiness detection system using OpenCV software and Haar Classifiers.
The real-time system has been successfully created to detect the face and hence the eyes and mouth of the driver to check whether he is blinking or yawning to acquire information about his level of alertness. The camera fixed near the headboard of the driver continuously captures images of the driver and these images are one by one processed. If the eyes are detected to be closed in more than 5 consecutive frames, the necessary buzzer sets off. However, the only alarm that has been implemented is a buzzer to alert the driver and we have not interfered with the actual working and performance of the vehicle.
The system has been tried and tested in different lighting conditions and with different people with varied facial characteristics. It has been experimentally found that absolute accuracy is achieved when the lighting conditions are bright and favourable. The biggest drawback experienced till now is the presence of beard or sunglasses or spectacles on the driver's face. This interferes with the detection of eyes and mouth and may lead to false triggering.

This system is real time and checks the state of the driver all through the journey. Captured video was divided into frames and each frames were analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing
image and analyzing the state of driver is carried out again and again. In this implementation during the drowsy state an alarm is given and Corresponding message is shown. If the driver is not drowsy, then it remains constant.This is a computer vision system that can automatically detect driver drowsiness in a real-time video stream and then play an alarm if the driver appears to be drowsy.
by this method we can give alert to driver about his drowsiness so that we can reduce the count of accidents and save many lives

7.2 Future Scope

In Future, we can update this system in many ways such as,

1. We can build a system that is linked with an application on a smartphone and check the real-time reaction rate through actual testing.
2. We can build a system where it can know when the driver is driving rash, it can subtly alert him.
3. We can build a system that also helps to know whether the driver has beyond the recommended levels of alcohol it can help them to decide to drive or not.
4. We can build that is connected to the cloud.

REFERENCES

1. Driver drowsiness detection using ANN image processing
2017
T. Vesselenyi at University of Oradea
https://iopscience.iop.org/article/10.1088/1757-899X/252/1/012097/pdf

2. A Deep Neural Network for Real-Time Driver Drowsiness Detection
2019
Toan H. VU
https://www.jstage.jst.go.jp/article/transinf/E102.D/12/E102.D_2019EDL8079/_pdf

3. Real-Time Driver Drowsiness Detection System Using Eye Aspect
Ratio and Eye Closure Ratio
2019
https://poseidon01.ssrn.com/delivery.php?ID=062021004021120115107006122123027091102008050058026054101090096024087014024115006095018061040055106006002003067108120115000115106016000058010116022120126090083023118067034057118124084009003125124000117009015084024080093107085084023025087024075066110090&EXT=pdf

4. Efficient Driver Fatigue Detection and Aler
ting System
2015
Miss. Kanchan Manohar Sontakke
PRECLOS(most helpful vision-based fatigue evaluation method)
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.2168&rep=rep1&type=pdf

5. Human Drowsiness Detection System
2019
Rukhsar Khan at IJEAT(International Journal of Engineering and Advanced Technology)
https://www.ijeat.org/wp-content/uploads/papers/v8i4/D6104048419.pdf

6. https://www.irjet.net/archives/V3/i12/IRJET-V3I12315.pdf

7. M. Eriksson and N. P. Papanikolopoulos, “Eye-tracking for detection of driver fatigue,” in Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC '97), pp. 314–319, 1997.
https://ieeexplore.ieee.org/abstract/document/660494

8. Y. Wu, J. Li, D. Yu, and H. Cheng, “Research on quantitative method about driver reliability,” Journal of Software, vol. 6, no. 6, pp. 1110–1116, 2011.
http://www.jsoftware.us/vol6/jsw0606-20.pdf

9. B. T. Jap, S. Lal, P. Fischer, and E. Bekiaris, “Using EEG spectral components to assess algorithms for detecting fatigue,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2352–2359, 2009.

https://www.sciencedirect.com/science/article/abs/pii/S0957417407006914

10. Camera-based drowsiness reference for driver state classification under real driving conditions

https://ieeexplore.ieee.org/document/5548039

11. Driver Sleepiness Detection System Based on Eye Movements Variables

https://journals.sagepub.com/doi/full/10.1155/2013/648431

12. Real Time Driver Drowsiness Detection Based on Driver's Face Image Behavior Using a System of Human Computer Interaction

https://www.researchgate.net/publication/322250457_Real_Time_Driver_Drowsiness_Detection_Based_on_Driver's_Face_Image_Behavior_Using_a_System_of_Human_Computer_Interaction_Implemented_in_a_Smartphone

13. Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3356401

14. Driver's Fatigue and Drowsiness Detection to Reduce Traffic Accidents on Road

https://link.springer.com/chapter/10.1007/978-3-642-23678-5_47

15. Drowsiness Detection Based on Eye Closure and Yawning Detection

https://www.ijrte.org/wp-content/uploads/papers/v8i4/D9716118419.pdf

APPENDICES

CODE:

```
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import wave
import dlib
import cv2
import os
import winsound




#alarm module
def alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

    while alarm_status:
        print('call')
        winsound.Beep(1000,2000)
        s = 'wave "'+msg+'"'
        os.system(s)

    if alarm_status2:
        print('call')
        winsound.Beep(1000,2000)
        saying = True
        s = 'wave "' + msg + '"'
        os.system(s)
        saying = False


def sound_alarm(path):
  # play an alarm sound
  winsound.Beep(1000,2000)
  playsound.playsound(path)


#detecting eye ratio
def eye_aspect_ratio(eye):
```

```
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear
#detecting ear ratio
def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)

#detecting lip ratio
def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))

    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))

    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance


ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
                help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0

#loading predictor and detector
```

```
print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")    #Faster but less
accurate
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')

#Starting camera
print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)



#infinity loop
while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #rects = detector(gray, 0)
    rects = detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30,
30),flags=cv2.CASCADE_SCALE_IMAGE)

    #for rect in rects:
    for (x, y, w, h) in rects:
        rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)

        eye = final_ear(shape)
        ear=eye[0]
        leftEye = eye [1]
        rightEye = eye[2]

        distance = lip_distance(shape)

        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

        lip = shape[48:60]
        cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)


    if ear<EYE_AR_THRESH:
      COUNTER += 1
      # if the eyes were closed for a sufficient number of
```

```
      # then sound the alarm
      if COUNTER >= EYE_AR_CONSEC_FRAMES:
        # if the alarm is not on, turn it on
        if not ALARM_ON:
          ALARM_ON = True
          # check to see if an alarm file was supplied,
          # and if so, start a thread to have the alarm
          # sound played in the background
          """if args["alarm"] != "":
            t = Thread(target=sound_alarm,
              args=(args["alarm"],))
            t.deamon = True
            t.start()"""
        # draw an alarm on the frame
        cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
          cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        winsound.Beep(1000,2000)
    # otherwise, the eye aspect ratio is not below the blink
    # threshold, so reset the counter and alarm
    else:
      COUNTER = 0
      ALARM_ON = False
    if (distance > YAWN_THRESH):
            cv2.putText(frame, "Yawn Alert", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            winsound.Beep(1000,2000)
            if alarm_status2 == False and saying == False:
                alarm_status2 = True
                t = Thread(target=alarm, args=('take some fresh air sir',))
                t.deamon = True
                t.start()
    else:
        alarm_status2 = False

        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300,
30),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300,
60),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    cv2.imshow("Frame", frame)
    key = cv2.waitKey(10) & 0xFF

    if key == ord("q"):
        break

cv2.destroyAllWindows()
vs.stop()
```