INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-2158)
Available online at: https://www.ijariit.com

# Driver Drowsiness Detection System

*Komati Jyoshna Srivalli*
*srivalli.chinni00@gmail.com*
*Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh*

*Dr. Esther Sunanda Bandaru*
*bonnie.sunanda@gmail.com*
*Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh*

*Karipireddi Swathi*
*karipireddi.swathi1@gmail.com*
*Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh*

*Kolli Sri Mahathi Gayathri*
*ksmahathig@gmail.com*
*Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh*

*Kommanaboina Lakshmi Prasanna*
*kommanaboinachimpu2000@gmail.com*
*Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh*

## ABSTRACT

***The world is becoming automated day by day in each and every aspect. The technological approaches to solve every human problem are evolving at a great interest in this rapid advancing environment. Automated houses, cars(for instance, Tesla self-driving cars), machinery, industries can be considered as the progressive outcomes which made the human life comfort, safe and easy. Although, there exist numerous situations that are damaging and the lives of people miserably . One of those circumstances is "Road Accidents". There may be many reasons for this problem where "Driver fatigue" is one of those. "Each year, around 1.5 lakh people die in road mishaps in India." –The Times of India[17]. To mitigate these accidents, we develop a model based on Convolutional Neural Networks (CNNs), Deep Learning which can detect the sleepiness or fatigue and notify the driver(or the person who is driving). In this paper, we developed a model using CNN classifier that identifies whether a driver is sleepy or not. The model here also provides an alerting alarm when the driver is sleeping (whenever he/she closes eyes). The proposed model is also evaluated using large amount of data to increase its accuracy and correctness while detecting the person's drowsiness.***

***Keywords:*** *Deep Learning, Convolutional Neural Networks(CNNs), Driver Drowsiness, Region of Interest(ROI), Haarcascade, OpenCV, Keras, Drowsiness detection, Face detection, Eye's detection, Classifier.*

## 1. INTRODUCTION

Innovation is leading the world nowadays. Majority of the critical hurdles and circumstances that affected the human lives in a terrible way earlier are now being solved with very much ease and intelligence that is increasing in a terrific way. Any work when done manually leads to any number of errors and inconsistencies. Also, it takes huge time to perform any task effectively. The ideas of development are emerging at a very fast pace to make any task simple, effective and efficient. "Artificial Intelligence" is the largest domain that is helping the mankind in solving the complexity of problems people face and thereby providing exact solutions respectively. Many subfields are also being developed as a result of lots of interest and long-time research. These fields include: Machine Learning( ML), Deep Learning(DL), Reinforcement Learning(RL), Deep Reinforcement Learning(DRL) and many more domains are under research.

With increase in technology, the human needs are still not satisfied completely in all the situations faced by the people. One of the deadliest crisis occurring today is "Road Accidents". There are many factors that are responsible for accidents occurrence. These factors include: 1.Sleepiness or fatigue, 2.Drunken driving, 3.Distractions to driver, 4.Overspeeding, 5. Not taking minimum precautions such as seatbelt or helmet and many more. However, "Driver Drowsiness" can be considered as the root cause for accidents on road. According to the National Sleep Foundation, about half of U.S. adult drivers admit to consistently getting behind the wheel while feeling drowsy[4]. About 20% admit to falling asleep behind the wheel at some point[10] in the past year – with more than 40% admitting this has happened at least once in their driving careers[4]. According to the National Highway Traffic Safety Administration, every year about 100,000 police-reported crashes involve drowsy driving[4]. These crashes result in more than 1,550 fatalities and 71,000 injuries[4]. The real number may be much higher, however, as it is difficult to determine whether a driver was drowsy at the time of a crash or accident that happens un-intensionally.

A study by the AAA Foundation for Traffic Safety estimated that 328,000 drowsy driving crashes occur annually. That's more than three times the police-reported number[4]. The same study found that 109,000 of those drowsy driving crashes resulted in an injury and about 6,400 were fatal[4]. The researchers suggest the prevalence of drowsy driving fatalities is more than 350% greater than reported[4]. As per the data received from the police department of States/UTs, 1,50,785 persons and 1,47,913 persons were killed in road accidents during the calendar years 2016 and 2017 respectively. In other words, around 400 people on an average are losing their lives every day on Indian roads[4].

We repeatedly think that the bad weather, road conditions and vehicle failure are only the reasons for the accidents. But the unrecognized human error is "Drowsiness". Drowsiness is the main culprit that reduces the reaction time (or response time) of a person. It decreases the thinking power and ability to make decisions accurately and effectively. We have "breathalyzer" to determine the alcohol content in driver blood in case if the driver is drunken and driving. We have "speedometer" to find the speed of the vehicle in case if the driver is on over-speeding. Also, we are capable of analyzing the weather and forecasting it earlier to all the people. But we are not having any means of identifying the person "sleepiness" when he/she is driving.

The research says "Driving when drowsy is similar to that of driving under influence of alcohol"[4]. It is said that the people are more likely to be in a crash if they are fatigued. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy[10].

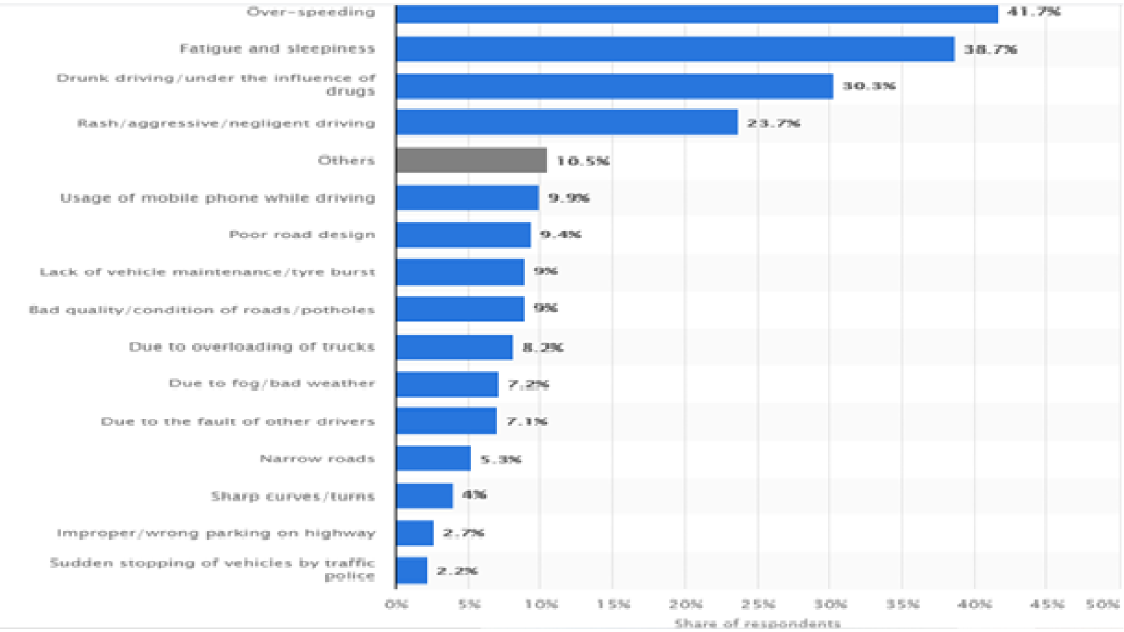

**Fig 1: Reasons for road accidents in India as of February 2020**

To reduce these incidents, we use "Deep Learning" domain to develop, train the model using CNN classifier and identify the drowsiness of the driver. In this paper, we propose a model that involves visualization techniques that include 1. Face detection 2. Eyes detection 3. Driver's state of eyes( whether they are open or close). The main objective of this paper is to design a detection system that is able to supervise the eyes of driver. The outcome is based on the symptom of driver closing the eyes (if the driver closes eyes, then action is performed, else the system monitors the driver continuously). The model is exposed to the considered dataset using data processing techniques and the result is erected ultimately.

The remaining sections of this paper are : Section 2 includes the related work and background research regarding the system design and development. Section 3 contains the system design or proposed approach. Section 4 describes the experimental setup or the implementation. Section 5 depicts the testing or the experimental results. Section 6 provides the conclusion and future work that can be performed on the experiment basis.

## 2. BACKGROUND AND RELATED WORK

In this section, we provide the theoretical work related to understand the problem and get a solution precisely. As fatigue related accidents reflect the major cause of crashes that occur across the world, researchers and organizations started working on different solutions that include identifying the patterns of driving habits, observing the surroundings of driver and analyzing the brain waves[13][16][11]. Many solutions are being generated in different dimensions, however most of the solutions are based on Machine Learning and Deep Learning on the basis of neural networks.

There exist many approaches have been conducted on detecting the driver drowsiness. One of those is the system that concentrates the driver's eyes[3]. It is based on a video surveillance system. It mainly deals with the observation of person's eyes and the frequency of eye blinking to evaluate the fatigueness of driver. Some studies such as Adaboost's, have used a cascade of classifiers to detect the drowsiness rapidly. This deals with performing and obtaining a solution at a great speed rapidly.

 

Another deep learning approach is the use of RNN model[15], which aims to analyze the slow and long closure of the driver's eyes. This approach uses a Haar-like descriptor and an AdaBoost classification algorithm for face and eye-tracking by using Percent Eye closure (PERCLOS)[8] to evaluate driver tiredness. PERCLOS evaluates the proportion of the total time that a drivers' eyelids are ≥80% closed and reflects a slow closing of the eyelids rather than a standard blink.

There exists an interesting approach based on the head position of the driver. To achieve better accuracy, a model based on Electro encephalography(EEG)[11][16] signals have also been proposed earlier. This solution involves Artificial Neural Networks(ANNs) used for classification which yielded 83.3% accuracy. Another model has been proposed based on combination of techniques such as Driving Quality Signals, EEG[13] and Electroculography[12]. A self-organized map network was used for classification and achieved 76.51% accuracy. Another model to find driver drowsiness was based on steering patterns[14]. In this model, three feature sets were constructed with the aid of advanced signal processing techniques to capture the steering patterns. Here, the performance is determined using Machine Learning algorithms such as SVM and K-Nearest Neighbor. The accuracy obtained in this solution is 86%.

Another interesting approach developed proposed in 2019 was a method to extract blood volume pulse and eye blink and yawn signals as multiple independent sources simultaneously by multi-channel second-order blind identification (SOBI)[1] without any other sophisticated processing, such as eye and mouth localizations[2][1]. An Internet of Things related approach proposed in 2018 used the eye closure ratio as input parameter to detect the drowsiness of the driver. If the eye closure ratio deteriorates from the standard ratio, the driver is alerted with the help of a buzzer[5].

Also, there is a model developed based on the Hidden Markov Model(HMM). Here, Viola Jones algorithm is used to detect the object i.e., face of driver and extract the face regions. Gabor wavelet decomposition is applied to extract the facial features[2]. HMM identifies the drowsiness of the person driving. An eye-tracking based system was also proposed using Viola Jones algorithm with an accuracy of 82% in simulating environment and 72.8% in real world environment. The system that evaluates the drowsiness based on the brain activity[16] and visual activity was also developed and proposed where brain activity is observed using EEG[16] and blinking and characterization for observing visual activity where further blinking activities are measured by EOG[12]. This system performed well with the accuracy of 80.6%.

Vision-oriented drowsiness detection system for bus driver surveillance was also developed[3]. This system includes identifying head and shoulder detection and driver detection using Histogram of Oriented Gradients(HOG) and Support Vector Machine(SVM) respectively[16]. Whenever the driver is identified, OpenCV was used for face detection[2][6] and eye detection[2][7]. For capturing eye shape, Spectral Regression Embedding was used. Drowiness detection for android applications was also introduced based on deep learning[9]. In this approach, first the images are extracted from Dlib library and after identifying the coordination locations of face, they are given as input to "Multi-Layer Perceptron" classifier[13]. The dataset used for this was NTHU Drowsy Driver Detection Dataset. It obtained an accuracy of 80%.

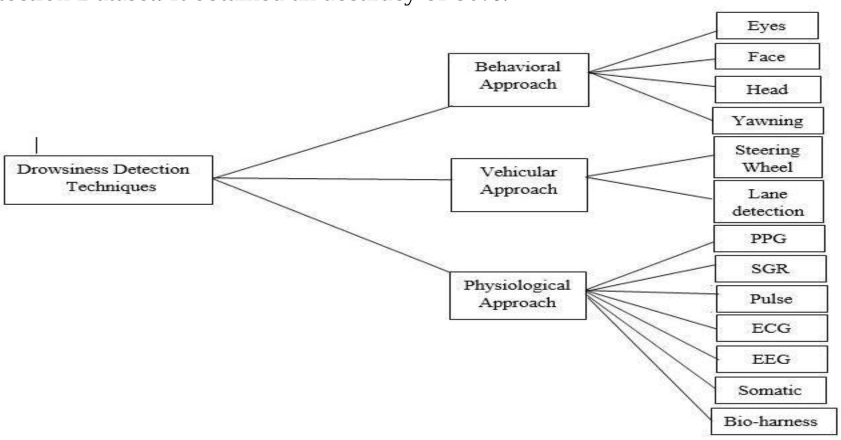


**Fig 2 : Approaches for detecting driver drowsiness**

## 3. PROPOSED MODEL

The architecture of our proposed model is depicted in Fig 3 below.

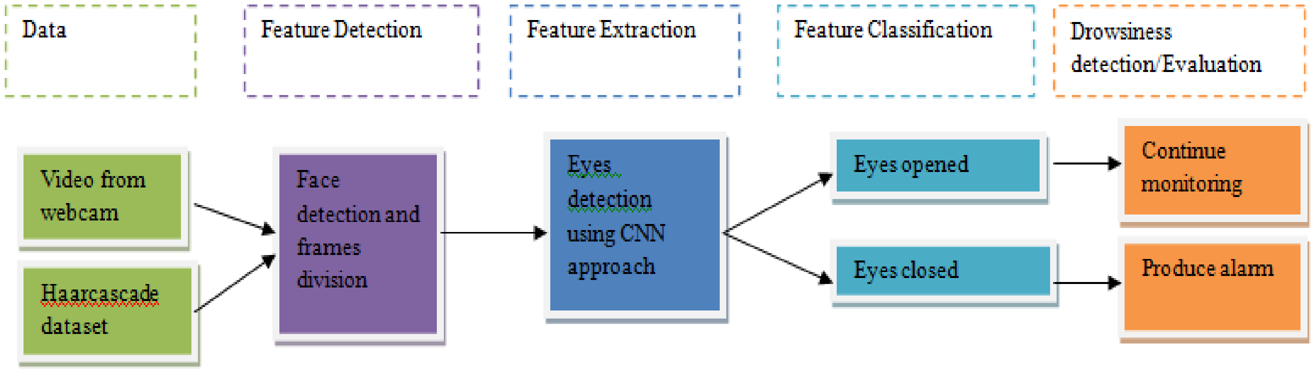


**Fig 3: Architecture of proposed model**

From the above figure, the entire procedure that is followed in this paper is displayed. This architecture includes the process of acquiring the video input from the camera/webcam using OpenCV library. The video input is then given to the model or classifier to identify the face and the Region of Interest(ROI) in the face. ROI as per this paper is "eyes". Then, the model classifies whether the eyes are in open state or closed state. If eyes are in closed condition, immediately an alarming sound is produced by the system in order to make the driver awake. If the eyes are in open condition, the system continues monitoring the driver.

### 3.1 Convolutional Neural Networks

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

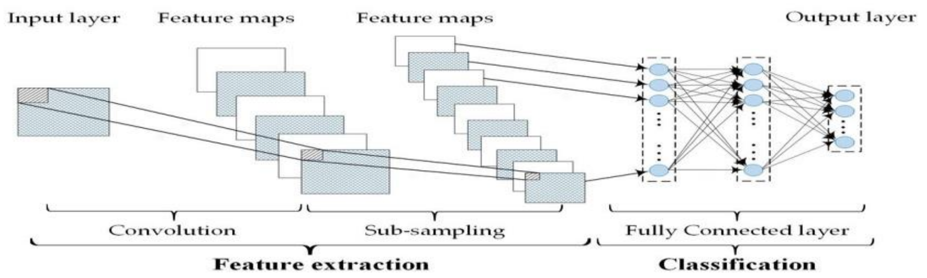


**Fig 4: Convolutional Neural Network(CNN) architecture**

A CNN basically consists of an input layer, an output layer and a hidden layer which can have multiple numbers of layers. A convolution operation is performed on these layers using a filter that performs 2D matrix multiplication on the layer and filter. The CNN model architecture consists of the following layers:

- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 64 nodes, kernel size 3
- Fully connected layer; 128 nodes

The final layer is also a fully connected layer with 2 nodes. In all the layers, a Relu activation function is used except the output layer in which Softmax is used(identifies whether the driver is sleepy or not sleepy).

### 3.2 System Modules

From fig 3, it can be observed that the system comprises of 5 main modules : 1.Video acquisition 2.Face detection and frame division 3.Eyes detection 4.Eyes classification 5. Drowsiness detection. Along with these 5 modules, we use webcam or camera to capture the video which is a typical hardware component and audio alarm for providing output. System flow can be represented as below in Fig 5.

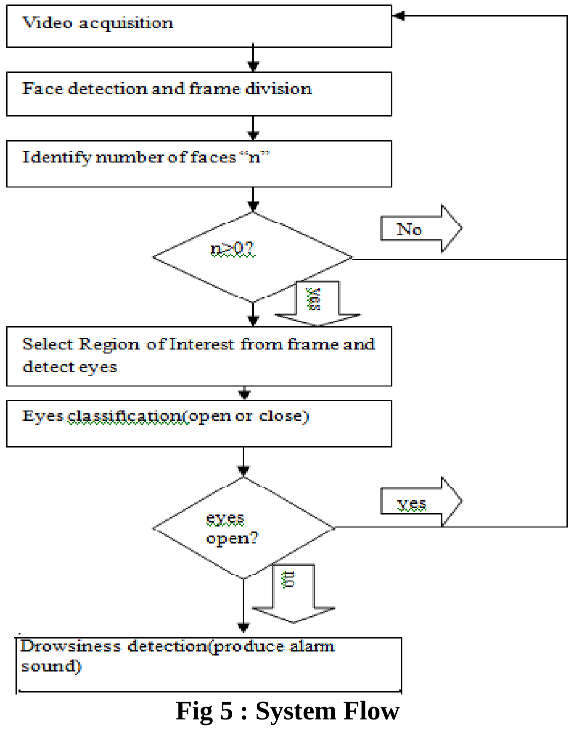


**Fig 5 : System Flow**

## 4. EXPERIMENTAL SETUP

In this section, we describe all the elements used in the experiment to explore our proposed model approach in drowsiness detection. This includes the dataset description and implementation of the proposed model.

### 4.1 Dataset

The dataset we use here is created using script that captures eyes from the camera and stores in our local disk. This dataset comprises of 7000 images of people's eyes under different lighting conditions. As it is known, nearly 70% of dataset is for training and 30% is for testing. These images are stored as a dataset in an architecture file "models/cnnCat2.h5". The entire dataset is split into training data and testing data. The training data is used primarily to get the results in different conditions. This process of using training data is often called as learning step. Later, we use the remaining data i.e., the testing data to explore the output. Here, this process of using testing data is referred to as construction step.

### 4.2 Experimental Methodology

This subsection provides the implementation process of each module of the system.

**4.2.1 Video Acquisition** : We take video as input from a camera/ webcam. OpenCV library plays a very important role in obtaining and processing visuals that are captured simultaneously. There is no requirement to mention the resolution for the camera when OpenCV is used because the library itself opens the camera with a default resolution. To access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, "cv2.VideoCapture(0)" to access the camera and set the capture object (cap). "cap.read()" will read each frame and we store the image in a frame variable.

**4.2.2 Face detection and Frame division** : Once the input video is obtained, our next step is to identify the face in the capturing video. The video is internally divided into number of frames that include image captured at a particular point of time in each frame. To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier "face = cv2.CascadeClassifier(' path to our haar cascade xml file')". Then we perform the detection using "faces = face.detectMultiScale(gray)"**.** It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face. However, we deal with the video and OpenCV that is processed using RGB video for processing. From this, we determine the Region of Interest(top part of face) and is shown using rectangle.

```
for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1 )
```

**4.2.3 Eyes detection :** Once we generate the frames from the video as mentioned in the second step, we need to detect the eyes of a person from the Region of Interest that is obtained from above step. This can be done by using the same step for detecting face. Instead of considering the entire frame for detecting eyes, we use convolutional neural network technique to extract Region of Interest(top part of the face) where eyes are present. Once the Region of Interest is determined, we apply edge-detection technique to identify the eyes in the frame. First, we set the cascade classifier for eyes in leye and reye respectively then detect the eyes using left_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from each full frame. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye part from the frame with this code. l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye.

**4.2.4 Eyes Classification:** After detecting the eyes image from the Region of Interest(ROI), the next step we perform is to determine whether eyes are open or closed. To find this, we feed the eyes image into the classifier(4.2.3), We are using CNN classifier for predicting the state of eyes. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images cv2.resize(r_eye, (24,24))**.** We normalize our data for better convergence r_eye = r_eye/255 (All values will be between 0-1). Expand the dimensions to feed into our classifier. We load our model using model = load_model('models/cnnCat2.h5'). lpred=model.predict_classes(l_eye) predicts whether the eyes are open or closed:

```
If lpred[0]=1, then eyes are open. If  lpred[0]=0, then eyes are closed.
```

**4.2.5 Drowsiness Detection :** After classifying whether the eyes are open or closed by the model or classifier, we calculate "score" that is crucial to detect the drowsiness of a person easily from the above data generated. "Score" is a value generally we use to identify how long a person has closed his/her eyes. if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of person.

```
cv2.putText(frame, "Open", (10, height-20), font, 1, (255,255,255), 1, cv2.LINE_AA )
```

 

Whenever the person/driver keeps his eyes open, the score is reduced continuously. On the other hand, if the person closes his/her eyes for two seconds or a certain number of continuous frames on the basis of frame rate, then the score value is incremented gradually and a beep alarm sound is played as warning to the driver(or person). This is done by "sound.play()". If the person closes his/her eyes in intervals and in a non-continuous manner, then it is considered to be "eye blink". It is known that converting the video from RGB to grayscale occupies very large amount of memory. To overcome this problem, instead of converting the video to grayscale, we can use RGB video directly for processing. This makes the processing easy and helps to consume very less space.

## 5. EXPERIMENTAL RESULTS

This section specifies the results obtained by conducting the experiment. We depict the testing results on many conditions. These conditions include: 1. Lighting conditions, 2. Position of driver face, 3. Driver's head tilted.

### 5.1 Test case 1 : Lighting conditions

**Fig 6 : Result when there is more light and when there is mild darkness**

**Result:** From Fig6 above, the result is clear that when there is plenty of light or when the light is dark mildly, the face, eyes and drowsiness of the driver are successfully detected producing the alarm audio when the drowsiness is detected.

### 5.2 Test case 2 : Position of driver face

#### 5.2.1 Face is at center

**Fig 7 : Face at center position in frame**

**Result:** From Fig 7 above, the result is clearly obtained that when the driver's face is at the center while capturing using webcam, the face, eyes and drowsiness are detected successfully producing the alarming audio when the drowsiness is detected.

#### 5.2.2 Face is at left

**Fig 8 : Face at left position in frame**

**Result:** From Fig 8 above, it can be observed that when the face of driver is kept at the left side of cameramwhile capturing the input, the face, eyes, eye blinks and drowsiness are successfully detected producing the audio alarm while the driver is closing his/her eyes.

#### 5.2.3 Face is at right

**Fig 9 : Face at right position in frame**

**Result:** From Fig 9 above, it can be observed that when the face of driver is kept at the right side of camera while capturing the input, the face, eyes, eye blinks and drowsiness are successfully detected producing the audio alarm while the driver is closing his/her eyes when the drowsiness is detected.

 

### 5.3 Test case 3 : Driver's head tilted


**Fig 10 : Tilted head position in frame**

**Result:** From Fig 10 above,it can be clearly observed that when the driver's head is tilted to any angle to any side of the camera while capturing the video input, the face, eyes, eye blinks and drowsiness are detected clearly thereby producing the beep alarm sound when the drowsiness is detected.

## 6. CONCLUSION

The main objective of this paper is to develop a "Drowsiness surveillance system" that can be applied in real world environment. This can be embedded in many automobiles. This system helps the people in becoming alert while there are in driving which make their drives safe. In this project, we designed and developed a basic system that can capture the video of the driver(or)person (Video Acquisition), identify the face and Region of Interest(ROI) from the input (Face detection and frame division), detect the eyes from the Region of Interest(ROI) from the frames (Eyes Detection), check whether eyes are open or closed (Eyes Classification) and detect the driver drowsiness. All these modules can be implemented according to the requirements. This system also provides an alarming alert sound as output when the model predicts drowsiness continuously. This system not only worked in the abundance of light but also in mild darkness where preference is on eyes.

### 6.1 Limitations

This system has its own limitations that include:

- The system does not work when the driver wear sunglasses or shades(since, eyes are the important factor of this system functionality and they cannot be identified when sunglasses are used).
- The system does not work when the light is incident on the camera/webcam directly.

### 6.2 Future Work

To expand this as a future work, it would be impactful to explore the system with more complexity and finding the solution. This may include working on the application of new techniques and methodologies that are emerging at a great pace. Future enhancements can be done such that this system may be developed without any limitations as mentioned. Gathering more data with many features is another way of improving the performance of the system. Considering automation approaches in Deep Reinforcement Learning(DRL) domain where the agent learns to make decisions using trial and error method can be done. This system may be enhanced by including the external factors like weather, steering, automobile condition. Planning, Designing, Developing, Testing, Deploying and Maintaining are the main steps to either design a system or enhance a system. There exist many fields in which this system can be enhanced:

- **Mobile application:** This system can be deployed as a mobile application where it can be accessed by anyone, from anywhere and at anytime. By doing this, people become safer.
- **Internet of Things:** This system can be improved by including sensors and activators that help in receiving input and providing output respectively. Usage of RaspBerry pi, Arduino may increase the performance and accuracy of this system.
- This system can be improved for mass transportation that include bus transportation, trains and aeroplanes where hundreds and thousands of people's lives will be saved.

The performance and accuracy of the proposed system can be improved by applying many algorithms like YOLO(You Only Look Once), SSD(Single Shot Detector) to the system for detecting moving objects. Automation can be included more which may detect the sleepiness or fatigueness of the driver and may reduce the vehicle speed automatically.

## 7. REFERENCES

[1]. Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications, published by IEEE, January 2019.
[2]. Real-Time Driver-Drowsiness Detection System Using Facial Features, published by IEEE, 21 August 2019.
[3]. Driver Drowsiness Detection System using Image Processing Technique by the Human Visual System, International Journal of Emerging Technologies in Engineering Research (IJETER) Volume 6, Issue 6,June (2018).
[4]. National Safety Council (NSC) , https://www.nsc.org/company.
[5]. IOT Based Real-Time Drowsy Driving Detection System for the Prevention of Road Accidents, **DOI:10.1109/ICIIBMS.2018.8550026, Conference: 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), October 2018.**
[6]. Seemar Puja*et al., "DROWSY DRIVER DETECTION USING IMAGE PROCESSING", INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY(IJESRT), July, 2017.
[7]. Xia Liu; Fengliang Xu; Fujimura, K., "Real-time eye detection and tracking for driver observation under various light condtions," Intelligent Vehicle Symposium, 2002, IEEE, vol.2, no., pp.344,351 vol.2, 17-21 June 2002.
[8]. A PERCLOS-Based Driver Fatigue Recognition Application for Smart Vehicle Space, IEEE, December 2010.
[9]. Detection of driver drowsiness in driving environment using deep learning methods, Vedat Tümen; Özal Yıldırım; Burhan Ergen, published by IEEE, April 2018.

[10].Transport Accident Commission (TAC), “Fatigue Statistics”: https://www.tac.vic.gov.au/road-safety/statistics/summaries/fatigue-statistics.
[11]. A C. S. Wei, Y. T. Wang, C. T. Lin, and T. P. Jung, “Toward Drowsiness Detection Using Non-hair-Bearing EEG-Based Brain-Computer Interfaces,”IEEE Transactions on Neural Systems and Rehabilitation Engineering,2018.
[12]. Chieh, T.C., Mustafa, M.M., Hussain, A., Hendi, S.F., Majlis, B.Y.: Development of vehicle driver drowsiness detection system using electrooculogram (EOG). In: 1st International Conference on Computers, Communications, Signal Processing with Special Track on Biomedical Engineering, pp. 165–168 (2005).
[13]. Chaabene, S.; Bouaziz, B.; Boudaya, A.; Hökelmann, A.; Ammar, A.; Chaari, L. Convolutional Neural Network for Drowsiness Detection Using EEG Signals. Sensors 2021, 21.
[14]. Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection Sadegh Arefnezhad, Sajjad Samiee, Arno Eichberger, Ali Nahvi Sensors (Basel) 2019 Feb; 19(4): 943. Published online 2019 Feb 22.
[15]. Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network Younes Ed-Doughmi 1,* , Najlae Idrissi 1 and Youssef Hbali 2, MDPI, Journal of Imaging, 4 March 2020.
[16]. Eeg (Brain Control Interface) Based Drowsiness Detection And Impact Identification, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-5 March, 2019.
[17]. The Times of India, https://timesofindia.indiatimes.com/blogs/the-interviews-blog/.