**SIMULATING DRIVER DROWSINESS ALARM SYSTEM: ENHANCING ROAD SAFETY THROUGH VIRTUAL TESTING**

**ABSTRACT**

In the realm of road safety, the issue of driver drowsiness looms as a formidable challenge, replete with the potential for catastrophic outcomes. In response to this pressing concern, we unveil a pioneering driver drowsiness detection system, an exemplar of harmonious convergence between the venerable OpenCV computer vision toolkit and the cutting-edge power of Convolutional Neural Networks (CNNs).

Our methodology orchestrates OpenCV's formidable capabilities, orchestrating a virtuoso performance in face and eye detection using a meticulously tuned haar cascade classifier. This prelude, a testament to precision, provides the foundational canvas upon which our opulent CNN masterpiece is woven.Within the tapestry of our CNN model lies the essence of real-time vigilance. It discerns the subtle nuances of a driver's alertness, transforming the vehicle into a sentient sentinel. When the veils of drowsiness encroach, this sentinel awakens, invoking an integrated alarm system. It is not merely a clamor; it is the clarion call to vigilant action, a lighthouse in the fog of fatigue.

Our system, an opus of innovation, stands as a sentinel against the ever-present specter of drowsy driving. It is a testament to the intersection of technology and safety, a sentinel that thrives in the crucible of real-world scenarios. As the curtain rises on our symphony, we present empirical evidence of its virtuosity, underscoring its potential as a pivotal component in the modern mosaic of vehicle safety technology.

**KEYWORDS**

1. Transfer learning
2. YOLO
3. Cascade xml files
4. Simulation
5. Kernel
6. Softmax
7. ReLU
8. ROI
9. OpenCV
10. Threshold value

**INTRODUCTION**

In the context of contemporary road safety, the issue of driver drowsiness emerges as a paramount concern, harboring the potential for severe and even fatal consequences on our roadways. Recognizing the profound gravity of this perilous challenge and driven by an unwavering commitment to societal welfare, we embarked on a transformative journey – a mission to create an affordable and highly effective driver drowsiness detection system.

Our quest began with a steadfast pledge to tackle this issue comprehensively, leaving no stone unturned in our pursuit of holistic solutions. We ventured into the domains of object recognition, eventually guiding our path into the realm of emotion recognition, laying a robust foundation to address this multifaceted problem.

At the heart of our approach lies a synergy of cutting-edge technologies, a fusion of innovation poised to orchestrate a symphony of safety. We harnessed the formidable capabilities of OpenCV to initiate the process, skillfully navigating the intricacies of face and eye detection. Building upon this meticulous inception, we ascended into the realm of artificial intelligence, incorporating a Convolutional Neural Network (CNN) model, a beacon of real-time cognitive prowess, empowering us to discern and predict the status and condition of the faces and eyes under scrutiny.

Yet, our initiative does not culminate with mere identification; it metamorphoses into an active guardian of lives in transit. Upon the detection of telltale signs of drowsiness, our system transcends its role as a passive observer to intervene proactively, triggering an integrated alarm – a resounding clarion call, a sentinel awakening when it is needed most.

This comprehensive methodology constitutes an innovative nexus, enabling us to diligently process and analyze visual data within the intricate tapestry of real-world scenarios. Consequently, our driver drowsiness detection system emerges as a precious addition to the arsenal of modern vehicle safety technology. Our mission remains steadfast: to protect lives, preempt accidents, and unequivocally contribute to the realization of safer roads for all who traverse them.

# **LITERATURE SURVEY**

1. Fatigue detection using new CNN method

In the study conducted by Ed-Doughmi et al.[1], a novel method was introduced to analyze and anticipate fatigue employing a recursive neural network (RNN). This technique involved processing a sequence of video frames and implementing a multi-layer, model-based 3D convolutional network structure [2], derived from the RNN model. The focus was on identifying signs of fatigue in subjects, such as yawning, eye closure, and head nodding, using videos from the NTHUDDD dataset. Impressively, their approach achieved an impressive accuracy rate of 97.3% [3].

1. Fatigue detection using eye and mouth CNN

Zhao et al. introduced an automated fatigue detection algorithm [4], utilizing a multitask cascaded CNN architecture for face detection and feature point localization. Their innovative CNN, named EM-CNN, accurately identifies mouth and eye states within a designated region of interest (ROI). The algorithm's employment of PERCLOS and mouth opening degree parameters achieved an impressive 93.62% accuracy and 93.64% sensitivity. Image-based systems generally exhibit accuracies ranging from 72.25% to 99.59%, predominantly relying on eye state features for non-intrusive, cost-effective data collection via cameras. Nonetheless, challenges arise from obstacles affecting facial tracking, significantly influencing system performance. Further insights into these challenges are discussed in subsequent sections.

1. DDD with hybrid CNN and LSTM

Guo and Markoni [5] introduced an innovative real-time Driver Drowsiness Detection (DDD) method, merging Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM). This involves spatial (CNN) and temporal (LSTM) components. CNN extracts facial features, while LSTM processes these features to classify drowsiness. The process begins with face detection, landmark extraction, and spatial feature extraction through CNN. Temporal features are formed from concatenated spatial features via sliding windows, feeding an LSTM for drowsiness assessment. Using the NTHUDDD dataset from ACCV 2016 competition [6], the method achieved an 84.85% final accuracy across various experimental scenarios.

1. Eye Closeness

Khunpisuth et al. [7] investigated drowsiness in ten volunteers by monitoring eye blinking and head tilting frequencies, correlating them with drowsiness. They designed a detection system using a Raspberry Pi Camera and Raspberry Pi 3 Model B to gather image data, assess drowsiness, and issue alerts. The system employed a Haar cascade classifier for face alignment and eye-blink detection. Geometric rotation rectified head misalignment, while template matching identified open or closed eyes. Drowsiness intensity was gauged based on head tilting and eye-blink frequency on a 0–100 scale. An audible alarm activated at 100. Despite a remarkable 99.59% accuracy, the system faced limitations due to skin tone and lighting conditions.

1. Detection of driver drowsiness with CNN

Hashemi et al. put forth an immediate Driver Drowsiness Detection (DDD) framework centered on eye closure area and harnessed convolutional neural network (CNN) technology [8]. Within this context, three distinct networks emerged for eye closure classification: fully designed neural network (FD-NN), transfer learning through VGG16 (TL-VGG16), and transfer learning with augmented layers in VGG19 (TL-VGG19). To validate their approach, the researchers employed the ZJU gallery dataset alongside 4157 new images. The outcome showcased network accuracies of 98.15%, 95.45%, and 95%, respectively, for the network types.

1. Fatigue detection using convolutional two-stream network

Liu et al. [9] introduced a fatigue detection algorithm that utilizes a convolutional two-stream network termed the gamma fatigue detection network to process multiple facial characteristics like eye closure duration, head nodding, and yawning. The process begins by identifying the driver's eyes and mouth through multi-task cascaded CNNs. Subsequently, static features are extracted from partial facial images, while dynamic features are obtained from partial facial optical flow. These static and dynamic features are then fused using a two-stream neural network for classification. Additionally, the paper demonstrated that the application of gamma correction [10] to enhance image contrast resulted in a 2% accuracy improvement during night-time recordings. The algorithm's performance was validated using the NTHUDDD public dataset [11], achieving an accuracy rate of 97.06%.

1. EEG features with LSTM

Budak et al. [12] developed an EEG-based drowsiness detection method with three components. The first extracts features from EEG spectrogram images and calculates energy and zero-crossing distribution features from raw EEG signals. The second directly acquires comprehensive features from EEG spectrogram images using pre-trained models. In the third phase, EEG signals are divided into sub-bands via a wavelet transform, yielding sub-band spectrogram images and statistical attributes. These features are then fed into an LSTM classifier. The method was evaluated on MIT/BIH EEG dataset, using a subset of 16 subjects. A 10-fold cross-validation yielded an average accuracy of 94.31%.

1. Smartwatch-based wearable EEG system

Li et al. [13] introduced a driver drowsiness detection system rooted in EEG signals. Unlike conventional EEG-based systems that assign discrete drowsiness labels (drowsy or alert), their approach employs an SVM-based posterior probabilistic model to classify drowsiness into three categories: alert, drowsy, and early warning. This innovative technique transforms drowsiness levels into continuous values within the range of 0 to 1, offering a nuanced measure of drowsiness. The fully wearable EEG system, comprising a commercial smartwatch and a Bluetooth-enabled EEG device, facilitates real-time data analysis. The system demonstrated varied accuracies for each detected state, achieving 91.92% accuracy for drowsy instances, 91.25% for alert states, and 83.78% for early warning conditions.

1. Hypovigilance detection using higher-order spectra

Sahayadhas et al. [14] developed a system to detect hypo vigilance due to drowsiness and inattention using ECG and EMG signals. They assessed inattention through questions and drowsiness via 2-hour simulated driving. Extracting features like bispectrum, they achieved up to 96.75% accuracy for ECG (H3) and 92.31% for EMG (H2). Combining signals using principal component analysis, they reached a maximum 97.06% accuracy with fused features using KNN classifier.

1. DDD using EEG, EOG, and ECG signals with fuzzy wavelet packet-based feature-extraction

Khushaba et al. [15] introduced a feature extraction method to identify driver drowsiness states using EEG, EOG, and ECG signals. Their fuzzy mutual information-based wavelet packet transform optimizes data extraction for drowsiness levels, classifying from alert to extremely drowsy. Using data from 31 volunteers in a simulated driving environment, they extracted features like EEG from various channels, eyeblink rate, and vital signs. After dimension reduction with spectral regression, training with classifiers achieved up to 97% accuracy using kernel spectral regression.

1. Tracking drowsiness using SWA

McDonald et al. [16] suggested using SWA data and the RF algorithm to analyze lane departure. The scientists contrasted their strategy with another PERCLOS-based image-based sleepiness measure. The comparison revealed that the SWA measure's accuracy was higher, reaching 79% and able to predict tiredness by six seconds. The PERCLOS approach only managed to reach accuracy of 55% at the same time. A dataset (72 participants) from a study at the National Advanced Driving Simulator at the University of Iowa was used to evaluate the algorithm [17]. The tiredness associated to lane departure was retrieved from the raw simulator data using the modified observer assessment of drowsiness scale. Following leaving the lane, readings were obtained every minute. The features for the PERCLOS measurement were taken from a video.

1. Lateral distance using wavelet transform and neural network

A model to identify driver tiredness based on lateral distance was put out by Ma et al. [18]. By combining lane curvature, location, and curvature derivative, one may determine the lateral distance. The transportable instrumentation package system [19] and a video camera mounted on the front bumper of the automobile were used to collect those three raw characteristics. Additionally, this system records live video in order to record the driver's head and face motions. The ground truth for the data from the automobile was the driver's visual data. To extract lane-related signals in the frequency and temporal domain, TRW's simulator was fed the recorded automobile data. The signals and the captured video of the driver's face were then examined. The data were then put into SVM.

1. Entropy features from SWA time series

During 14.68 hours of actual driving, data were gathered from a sensor mounted on the steering wheel. A technique that extracts the approximation entropy characteristics from a SWAs time series data by using a fixed sliding window was proposed by Li et al. [20]. The estimated entropy features series are then linearized using adaptive piecewise linear fitting with a predetermined deviation. The algorithm next computes the warping distance between the linear feature series to ascertain the level of awareness of the driver. A specifically created binary decision classifier is then used to identify the alertness state, either "drowsy" or "awake." The system's accuracy was 84.85% for "drowsy state" true detections and 78.01% for "awake state" true detections, according to experimental data.

1. ANFIS based steering wheel feature selection

Using information from the steering wheel, Arefnezhad et al. [21] demonstrated a non-invasive DDD system. The suggested technique of selection made use of wrapper feature selection algorithms, filters, and adaptive neuro-fuzzy inference systems (ANFIS). A new dataset was produced as a consequence of the study, which involved 39 bus drivers in a simulated driving environment. The steering wheel data were used to derive 36 characteristics. Four different filter indices received these features. To choose the most crucial traits, the fuzzy system was given the output of each filter. The selected characteristics are then classified using an SVM classifier, which also specifies the state of the drivers. Finally, the accuracy of the classifier is used to adjust the ANFIS's parameters using the particle swarm optimization technique. The outcome revealed accuracy of 98.12%

1. DDD based on steering wheel status

Then, based on these parameters, three models were created: a multilevel ordered logit (MOL) [22], SVM, and back propagation neural network (BPNN) model. The findings indicated that the MOL model had outperformed the others with an accuracy of 72.92% under the identical categorization settings. The MOL model has performed better than the other two models when employing these four factors while taking individual variations into account, according to the scientists' findings.

16. Driver assistance system, based on image- and vehicle-based features

In their work [24], Saito and his research team introduced a groundbreaking driver assistance system that employs a dual control strategy. This pioneering system demonstrates remarkable proficiency in identifying driver drowsiness through a comprehensive analysis of key indicators, including eyelid conditions, steering wheel interaction, and lane departure tendencies. Crucially, it possesses the capability to assume control of the vehicle when deemed essential. In scenarios where lane departure becomes imminent, the assistance system intervenes judiciously by offering partial guidance to the vehicle. This measured intervention not only aids in preventing accidents but also affords the driver a valuable window to regain control and safely reposition the vehicle within its designated lane.

17. Biomedical and motion sensors

A wearable system that employs motion and biological sensors to detect sleepiness via a mobile application was proposed by Leng et al. [25]. To provide an accurate result, the system integrates measurements from the car with biosignals from the driver. It makes use of motion sensors and a wristband it self-designed to detect biosignals and steering wheel movement. Galvanic skin response and photoplethysmogram sensors that pick up PPG signals are the wristband's two major parts. A smartwatch's built-in gyroscope and accelerometer are also utilized to measure the radian and linear speeds of the steering wheel. As a result, the system begins by gathering information from the sensors. The wristwatch will then evaluate and analyze the data after it has been acquired. Following that, five features—heart rate, stress level, respiration rate, adjustment counter, and pulse rate variability—are derived from the biological raw data that was received. These five variables are then loaded into an SVM algorithm together with steering wheel data to determine the driver's level of tiredness.

18. Yawning, blinking, and blood volume pulse-based method

. This feature led Zhang et al. [26] to develop a DDD system that makes use of the camera on a smartphone as a non-contact optical sensor. Blink and yawn signals are extracted from image sequences that the system has recorded and used as raw data. The extended-PPG signals that were collected from the picture sequence also make it possible to extract the blood volume pulse without coming into close touch with the subject. The blood volume pulse, yawning, and blinking signals are concurrently retrieved from smartphone footage using a multichannel second-order blind identification. The blinking duration and frequency, HRV, and yawning frequency are then calculated from the combined data. In the event that any of the estimated parameters exhibits a particular value, sleepiness will be proclaimed, and a phone alert will ring.

19. DDD with a smartphone

A smartphone-based DDD and warning system was suggested by Dasgupta et al. [27]. One of the first attempts to integrate speech cues with PERCLOS to detect sleepiness was this suggested system. They created their own dataset, known as the Invedrifac dataset [28], as a result. Three verification steps are used in the detection method in the work that has been presented. The PERCLOS feature is calculated in the first step using front camera photos from a smartphone. The second step is started by the system asking the driver to say his entire name if the PERCLOS number rises beyond a certain threshold. The technology prompts the motorist to tap the smartphone screen within 10 seconds after identifying them as a sleepy driver in the first two stages. Drowsiness is confirmed if the requirement is not satisfied, and

20. Combined EEG/NIRS DDD system

In their work, Nguyen et al. [29] presented a method for detecting driver sleepiness that combines EEG and near-infrared spectroscopy (NIRS). NIRS is a spectroscopic technique that makes use of the near-infrared portion of the electromagnetic spectrum. Nine volunteers participated in a driving simulation activity during which many biological signals were captured. These measures included those of heart rate using ECG signals, eye movement using EOG signals, tissue oxygenation using NIRS, and neuronal electrical activity using EEG signals. Heart rate, the strength of the alpha and beta bands, blinking frequency, and the length of the eye closure were among the factors examined to identify the sleepiness condition. The frontal beta band and oxygenation change were chosen as the most important factors from the analysis because statistical analyses revealed that they demonstrated the most significant difference between the alert and sleepy states.

**METHODOLOGY**

We have performed three the use cases of image recognition, however our main aim was to build an affordable and robust Driver drowsiness detection system.The phases are as follows

1.Object Detection

2.Emotion detection

3.Driver drowsiness detection system

We have used OpenCV to collect the webcam photos and feed them into a Deep Learning model to determine if the human's eyes are "Open" or "Closed.

Step 1.Take a real time video from a camera as the first step.

Step 2.Create a Region of Interest (ROI)

Step 3. Identify the eyes from the ROI and input the information to the classifier.

Step 4.The classifier will classify whether the eyes are open or shut.

Step 5: Determine the score to see if the subject is sleepy.

The dataset that we have used is obtained from dataflair. This contains images of 2 categories,that of eyes classified as open or closed and that of the entire face classified as yawn or no yawn.

**The Model Architecture**

Convolutional neural networks (CNN), developed with Keras, were utilized to create the model that we employed. Convolutional neural networks are a specific variety of deep neural networks that excel at classifying images. In essence, a CNN is made up of three layers: an input layer, an output layer, and a hidden layer with potential for more layers. These layers are put through a convolution operation with a filter that multiplies their 2D matrices together.

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 last layer has two nodes and is also completely connected. Except for the output layer, where we utilized Softmax, all the layers use a Relu activation function.

step-by-step algorithm at work.

Step 1.Take a real time video from a camera as the first step.

We will capture photos using a webcam as input. We therefore created an infinite loop to access the webcam and capture each frame. We employ the cv2 technique offered by OpenCV.for accessing the camera and setting the capture object (cap). We read each frame using cap.read(), then put the image in a frame variable.

Step 2.Create a Region of Interest (ROI)

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 a 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.

Step 3. Identify the eyes from the ROI and input the information to the classifier.

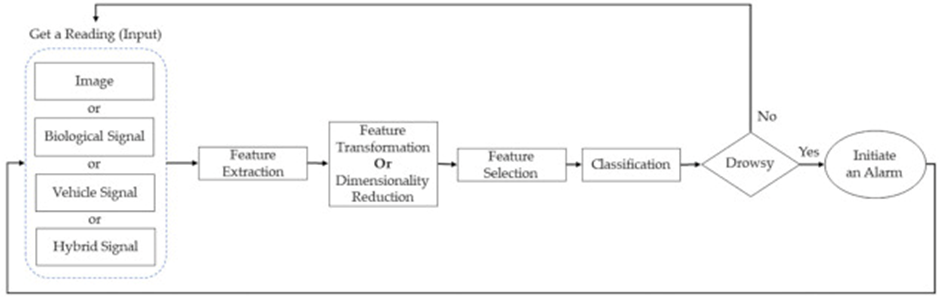
The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in left eye and right eye respectively then detect the eyes using left\_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame.

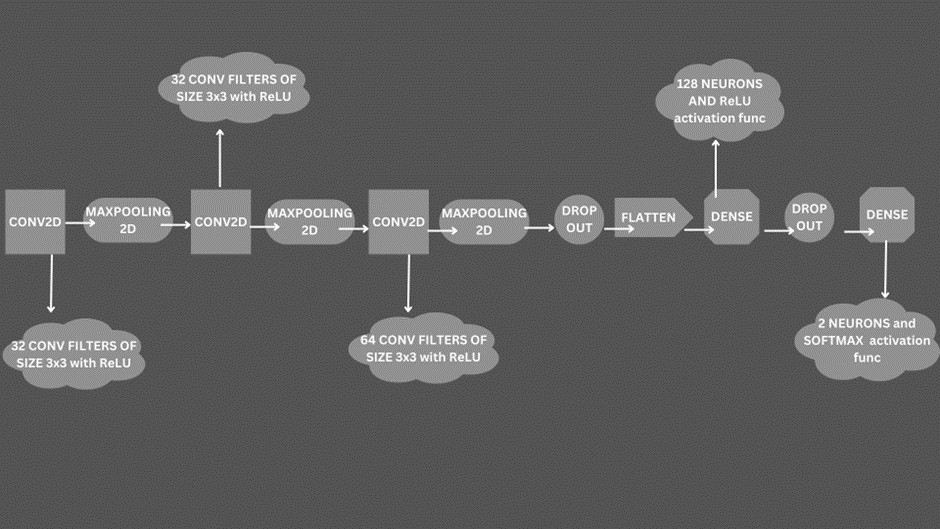
Step 4.The classifier will classify whether the eyes are open or shut.

We are using CNN classifier for predicting eye status. 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 loaded our model. Now we predict each eye with our model.If the value of 1, it states that eyes are open, if the value of = 0 then, it states that eyes are closed.

Step 5: Determine the score to see if the subject is sleepy

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing the score and when our 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 the person.A threshold is defined for example if the score becomes greater than 15 that means the person’s eyes are closed for a long period of time. This is when we beep the alarm.



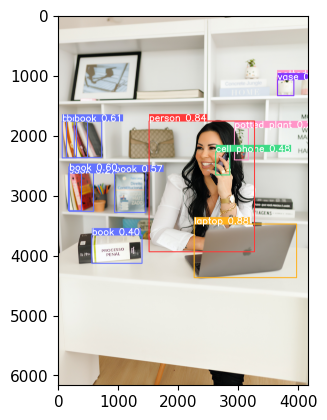


**RESULTS**

As mentioned in the methodology section, we have performed the execution of this simulation by thorough understanding of computer vision in 3 phases:

Phase 1 results:

Object detection helps us in identifying a person so we can segment and localize our focus onto the person, here, the driver. Some screenshots of the real time object detection can be visualized as follows wherein accuracy of object detection can be identified through the coloured boxes with the accuracy scores embedded onto them.

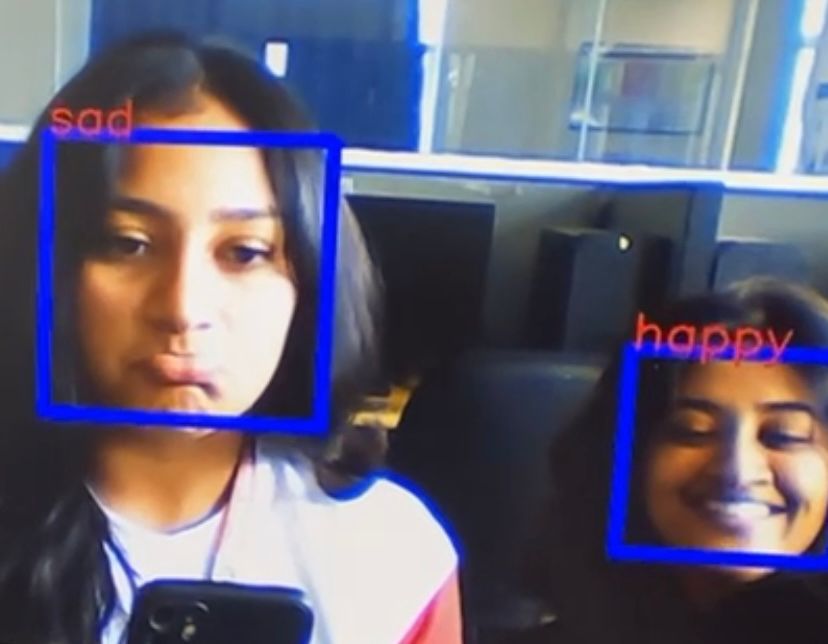


We have used the YOLO V5 model, as a part of transfer learning during object detection using CNN.

The image of the left signifies the object detection on a still image whereas the object on the right is a still from a real time prediction. It is very clear that this model predicts a large number of objects with excellent accuracy. On the left it classifies objects as a book, textbook ,vase ,cell phone, laptop with an average accuracy of 75% whereas in the real time prediction, it classifies the two people in the screen with an accuracy of 60%, 65% thus proving that this algorithm is accurate for our driver detection part.

Phase 2 results:

In phase 1 we built a model that identifies the driver and saved that model. In phase 2 we build a CNN model using transfer learning via Mobilenet architecture to identify emotions in real time. Similar to the previous pattern, we implement the OpenCV model that identifies a range of emotions in real time. Following are the set of emotions recognised by the algorithm and mentioned alongside in the box.

****

Using this algorithm a wide range of emotion recognition can be done. In the image on the left, it is identifying girl 1 is sad based on the frown and girl 2 is happy based on the smile. In the image on the right, it is identifying girl 1 to be in a state of fear from the eyes and forehead and girl 2 to be surprised based on the pout. Using such precision, in real time, the yawns or droopiness in the eyes of the driver can easily be analyzed using this algorithm. Again this model is stored thus completing the drowsiness detection part.

Phase 3 : It is basically an integration of the models created during phases 1 & 2 to create the actual simulation of the driver drowsiness alarm system by building a customized CNN model using the previous 2 stored models and adding an alarm sound that triggers when the algorithm detects that the driver’s eyes are shut for 20 seconds or more which can be visualized as follows:



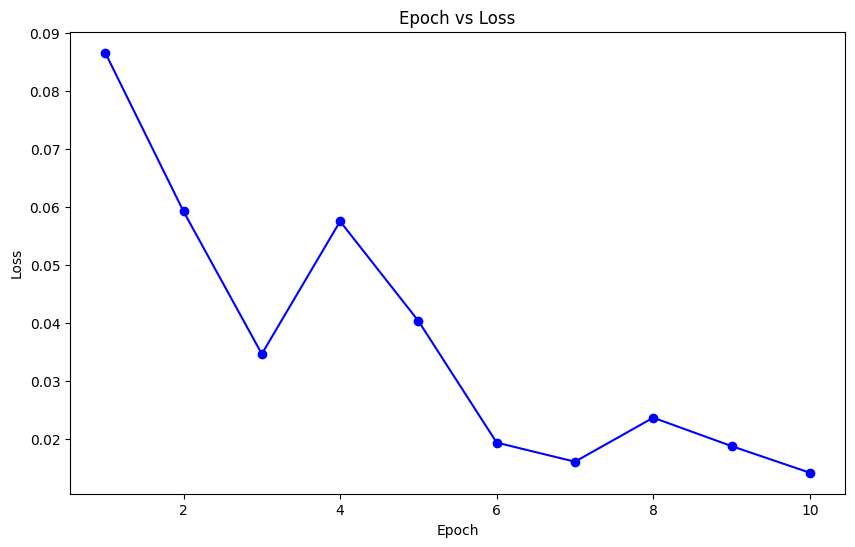
After compilation, the OpenCV model opens a window as follows, where the model analyzes the driver and his/her emotions constantly, in real time. It shows 2 attributes namely, Open/Closed and a Score.

Image on the LHS: While the driver’s eyes are Open, the score stays at 0.

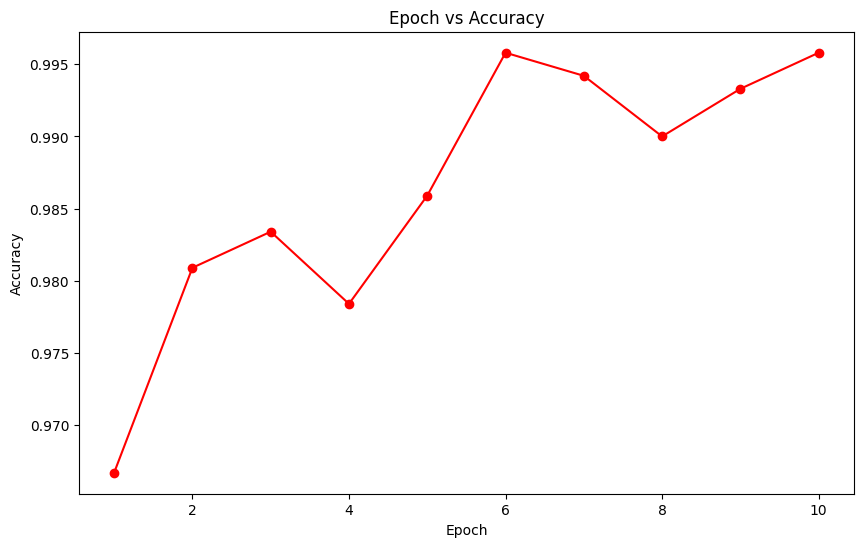
Image on the RHS: While the driver’s eyes are Closed, the score shoots up based on how long the driver has shut his/her eyes and is explained further along.



The average accuracy provided by the algorithm during real time analysis is a whopping 97.32% which shows that this simulation works perfectly well to analyze the driver drowsiness.



The Epoch vs Loss line graph indicates that loss of model is reducing over time during real time analysis.



The Epoch vs Accuracy line graph indicates that accuracy of predicting the driver drowsiness is increasing over time during real time analysis.

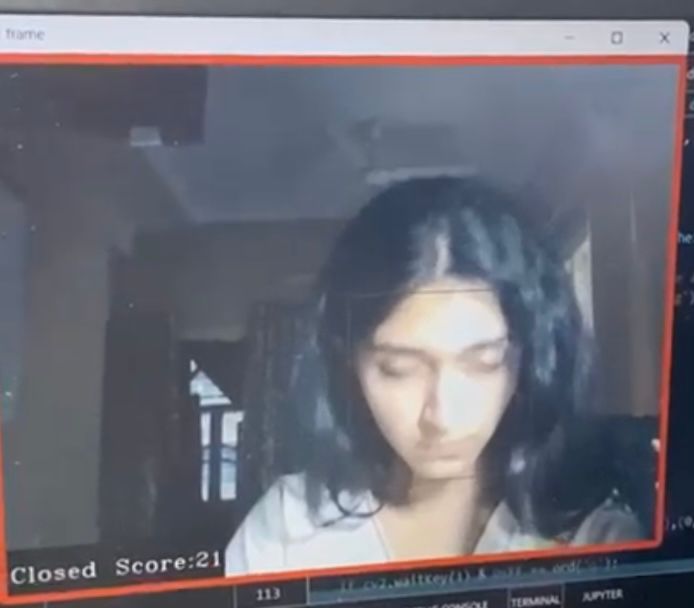
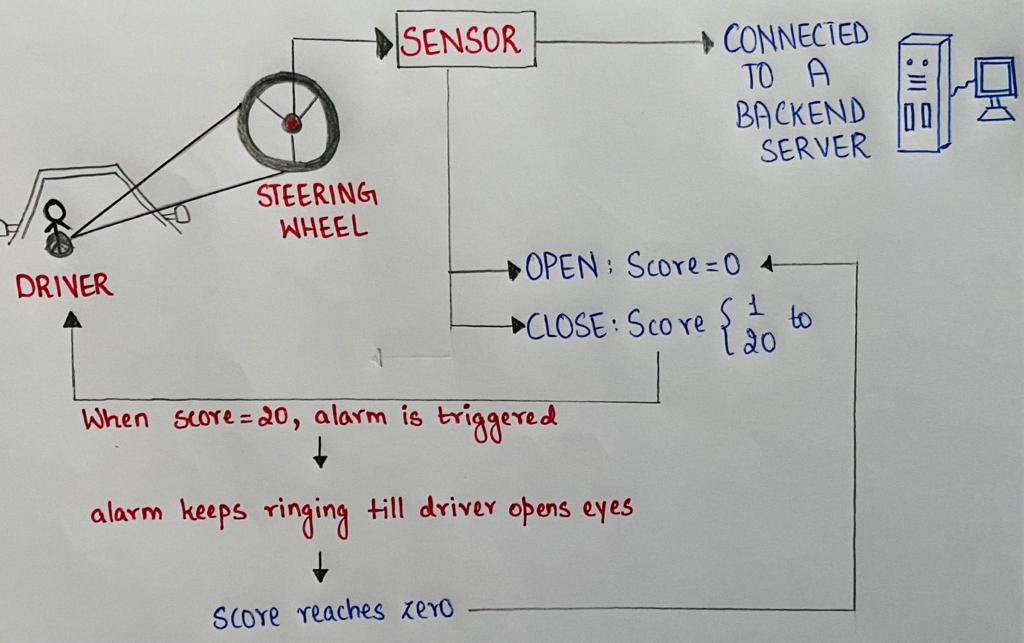
 

Image on the LHS: While the driver’s eyes are Open, the score stays at 0 and there is no change in the score as long as her eyes are open.

Image on the RHS: While the driver’s eyes are Closed, the score starts increasing at a rate of 1 score/ second. Usually, drivers may shut their eyes for blinking, or to prevent glares from a vehicle’s headlamp in the opposite direction, but that lasts only up to 5 seconds max so the alarm does not get triggered although score increases to 5. But once the score reaches 20, i.e., if the driver has closed her eyes for more than 20 seconds, the alarm gets triggered (indicated by the red box in the above screenshot) and continues beeping till the score gets down to 0 ( like the image on the left).



In this way, we have created a simulation for the alert system that is accurate, cost effective and easy to implement onto sensors and can be attached anywhere onto cars/ trucks using a simple device that has a tiny camera and sensors connected to a backend network.

**CONCLUSION**

In summary, the significant real-world problem of driver drowsiness poses severe risks, including the potential for fatal accidents. Driver drowsiness detection systems play a vital role in mitigating these risks and enhancing road safety. These systems serve as invaluable tools by effectively notifying drivers of their fatigue, prompting them to take necessary actions, and thereby contributing significantly to the advancement of modern vehicle safety technology.

In the lines of contributing to the well-being of the society,we have made an effort to build a driver drowsiness system that is cheap and effectively navigates the issue.

Our approach towards the system started off by experimenting with object recognition and we then moved on to emotion recognition which gave us a stronghold to work on a complex statement.Our approach involved utilizing OpenCV for the initial step of face and eye detection, employing a haar cascade classifier. Subsequently, we harnessed the power of a Convolutional Neural Network (CNN) model to make predictions regarding the status or condition of the detected faces and eyes. At the occurrence of drowsiness ,it alerts the driver using an alarm.This combined methodology allows us to effectively process and analyze the visual data for a specific task.

**REFERENCES**

1. Weng, CH., Lai, YH., Lai, SH. (2017). Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network. In: Chen, CS., Lu, J., Ma, KK. (eds) Computer Vision – ACCV 2016 Workshops. ACCV 2016. Lecture Notes in Computer Science(), vol 10118. Springer, Cham. https://doi.org/10.1007/978-3-319-54526-4\_9
2. Zhao, Zuopeng, et al. "Driver fatigue detection based on convolutional neural networks using EM-CNN." Computational intelligence and neuroscience 2020 (2020).
3. Guo, Jing-Ming, and Herleeyandi Markoni. "Driver drowsiness detection using hybrid convolutional neural network and long short-term memory." Multimedia tools and applications 78 (2019): 29059-29087.
4. Weng, Ching-Hua, Ying-Hsiu Lai, and Shang-Hong Lai. "Driver drowsiness detection via a hierarchical temporal deep belief network." Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13. Springer International Publishing, 2017.
5. Khun Pisuth, Oraan, et al. "Driver drowsiness detection using eye-closeness detection." 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE, 2016.
6. Hashemi, Maryam, Alireza Mirrashid, and Aliasghar Beheshti Shirazi. "Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network." SN Computer Science 1 (2020): 1-10.
7. Liu, Weihuang, et al. "Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection." Future Internet 11.5 (2019): 115.
8. Weng, Ching-Hua, Ying-Hsiu Lai, and Shang-Hong Lai. "Driver drowsiness detection via a hierarchical temporal deep belief network." Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13. Springer International Publishing, 2017.
9. Hoang, Thang, et al. "Generic gamma correction for accuracy enhancement in fringe-projection profilometry." Optics letters 35.12 (2010): 1992-1994.
10. Budak, Umit, et al. "An effective hybrid model for EEG-based drowsiness detection." IEEE sensors journal 19.17 (2019): 7624-7631.
11. Li, Gang, Boon-Leng Lee, and Wan-Young Chung. "Smartwatch-based wearable EEG system for driver drowsiness detection." IEEE Sensors Journal 15.12 (2015): 7169-7180.
12. Ed-Doughmi Y., Idrissi N., Hbali Y. Real-time system for driver fatigue detection based on a recurrent neuronal network. *J. Imaging.* 2020;6:8. doi: 10.3390/jimaging6030008.
13. Tran D., Bourdev L., Fergus R., Torresani L., Paluri M. Learning spatiotemporal features with 3d convolutional networks; Proceedings of the IEEE International Conference on Computer Vision; Santiago, Chile. 7–13 December 2015; pp. 4489–4497.
14. Sahayadhas, Arun, et al. "Physiological signal based detection of driver hypovigilance using higher order spectra." Expert Systems with Applications 42.22 (2015): 8669-8677.
15. Khushaba, Rami N., et al. "Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm." IEEE transactions on biomedical engineering 58.1 (2010): 121-131.
16. McDonald, Anthony D., et al. "Real-time detection of drowsiness related lane departures using steering wheel angle." Proceedings of the human factors and ergonomics society annual meeting. Vol. 56. No. 1. Sage CA: Los Angeles, CA: Sage Publications, 2012.
17. Ma, Jiaqi, Yi Lu Murphey, and Hong Zhao. "Real time drowsiness detection based on lateral distance using wavelet transform and neural network." 2015 IEEE symposium series on computational intelligence. IEEE, 2015.
18. Brown, T., et al. "Final report: Advanced countermeasures for multiple impairments." National Highway Traffic Safety Administration: Washington, DC, USA (2011).
19. Murphey, Yi L., et al. A Transportable Instrumentation Package for In-Vehicle On-Road Data Collection for Driver Research. No. 2013-01-0202. SAE Technical Paper, 2013.
20. Li, Zuojin, et al. "Online detection of driver fatigue using steering wheel angles for real driving conditions." Sensors 17.3 (2017): 495.
21. Arefnezhad, Sadegh, et al. "Driver drowsiness detection based on steering wheel data applying adaptive neuro-fuzzy feature selection." Sensors 19.4 (2019): 943.
22. Wang, Xuesong, and Chuan Xu. "Driver drowsiness detection based on non-intrusive metrics considering individual specifics." Accident Analysis & Prevention 95 (2016): 350-357.
23. Zhang, Chao, et al. "Driver drowsiness detection using multi-channel second order blind identifications." IEEE Access 7 (2019): 11829-11843.
24. Nguyen, Thien, et al. "Utilization of a combined EEG/NIRS system to predict driver drowsiness." Scientific reports 7.1 (2017): 43933.
25. Dasgupta, Anirban, Daleef Rahman, and Aurobinda Routray. "A smartphone-based drowsiness detection and warning system for automotive drivers." IEEE transactions on intelligent transportation systems 20.11 (2018): 4045-4054.
26. Saito, Yuichi, Makoto Itoh, and Toshiyuki Inagaki. "Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents." IEEE Transactions on Human-Machine Systems 46.5 (2016): 660-671.
27. INVEDRIFAC—A Video and Image Database of Faces of In-vehicle Automotive Drivers, India. 2019. [(accessed on 20 September 2021)]. Available online: https://sites.google.com/site/invedrifac/
28. Leng, Lee Boon, Lee Boon Giin, and Wan-Young Chung. "Wearable driver drowsiness detection system based on biomedical and motion sensors." 2015 IEEE SENSORS. IEEE, 2015.
29. Chand, H. V., & Karthikeyan, J. (2022). CNN Based Driver Drowsiness Detection System Using Emotion Analysis. Intelligent Automation & Soft Computing, 31(2).
30. Zilberg, E., Xu, Z. M., Burton, D., Karrar, M., & Lal, S. (2007, August). Methodology and initial analysis results for development of non-invasive and hybrid driver drowsiness detection systems. In The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007) (pp. 16-16). IEEE.
31. Kiashari, S. E. H., Nahvi, A., Bakhoda, H., Homayounfard, A., & Tashakori, M. (2020). Evaluation of driver drowsiness using respiration analysis by thermal imaging on a driving simulator. Multimedia Tools and Applications, 79, 17793-17815
32. Awais, M., Badruddin, N., & Drieberg, M. (2014, April). Driver drowsiness detection using EEG power spectrum analysis. In 2014 IEEE Region 10 symposium (pp. 244-247). IEEE.
33. Babaeian, M., & Mozumdar, M. (2019, January). Driver drowsiness detection algorithms using electrocardiogram data analysis. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0001-0006). IEEE.
34. Zhenhai, G., DinhDat, L., Hongyu, H., Ziwen, Y., & Xinyu, W. (2017, January). Driver drowsiness detection based on time series analysis of steering wheel angular velocity. In 2017 9th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 99-101). IEEE.
35. Guede-Fernandez, F., Fernandez-Chimeno, M., Ramos-Castro, J., & Garcia-Gonzalez, M. A. (2019). Driver drowsiness detection based on respiratory signal analysis. IEEE access, 7, 81826-81838.