ADVANCED DRIVING ASSISTANCE SYSTEM USING DEEP LEARNING TECHNIQUES

P. Archana[1], S.Sowndarya[2], S Santhana Raj[3]

Abstract

When someone gets sleepy while operating a car, it's a leading contributor to accidents worldwide, posing a significant threat to public safety. Fatigue and drowsiness, often stemming from lack of sleep, irregular work schedules, or underlying medical conditions, are common among many drivers, frequently resulting in devastating road accidents that can have far-reaching consequences. Alerting the driver in a timely manner is the most effective way to prevent accidents caused by drowsiness, as it can provide the necessary intervention to maintain the driver's alertness and prevent a potential tragedy. Various techniques, including computer vision and machine learning, exist to detect and monitor driver drowsiness, offering promising solutions to this pressing issue. In this study, we provide a deep learning-based method for detecting driver sleepiness that makes use of convolutional neural networks. It is a subclass of deep learning models renowned for their effectiveness in image and video analysis. The proposed scheme utilizes The areas of the driver's face and eyes to detect drowsiness, as these physiological cues are strongly correlated with fatigue and impaired alertness. The system continuously monitors the driver through a webcam, applying advanced image processing techniques that focus on the driver's face and eyes, extracting a rich set of facial features and analyzing eye blinking patterns, yawning, and other visual indicators of drowsiness. We employ a robust algorithm to observe and evaluate the driver's eyes and expression in real-time, measuring the onset and severity of drowsiness to ensure a timely and accurate response. If the system detects an elevated blinking rate or other signs of fatigue, it promptly alerts the driver with an audible warning, aiming to prevent a potential accident and safeguard the well-being of both the driver and the general public.

Department of Computer Technology[1]
Karpagam Academy of Higher Education[1]
archana.periasamy@kahedu.edu.in[1]

Department of Computer Science[2]
Akshaya College Of Arts and Science[2]
sowndarya26@gmail.com[2]

Master of Computer Science[3]
Karpagam Academy of Higher Education[3]
santhanarajs1711@gmail[3]

* Corresponding Author

Keywords Driver Detection, Convolution Neural Network (CNN), Region of interest (ROI)

I. INTRODUCTION

Driver fatigue has been a major contributor to countless vehicular mishaps, often stemming from factors such as driver exhaustion, challenging road circumstances, as well as adverse environmental circumstances. Annually, The World Health Organization and its National Highways Traffic Safety Administration have determined that 1.35 million people have died globally due to road crashes. Generally, these accidents are largely attributed to inadequate driving practices and poor awareness of the risks posed by driver fatigue. Hazardous situations can arise when a driver is impaired by alcohol, drugs, or drowsiness, with driver fatigue recognized as a severe factor leading to lethal accidents. When drivers succumb to sleep or become distracted, they lose control of their vehicles, often with devastating consequences that can include injury, loss of life, and extensive property damage. Driver fatigue can impair judgment, slow reaction times, and diminish the ability to maintain focus on the road, all of which significantly increase the likelihood of a catastrophic collision. There is a pressing need to develop advanced, smart or intelligent vehicle systems that leverage emerging technologies to actively monitor driver alertness and intervene when necessary. Such systems could detect signs of fatigue or distraction and provide real-time warnings or even take control of the vehicle to avoid or mitigate the impact of an impending accident. Implementing these innovative safety features has the potential to save countless lives and reduce the staggering human and economic toll of road crashes worldwide. [1][2][3]

This study offers a thorough method for identifying and warning the motorist when they exhibit signs of drowsiness or daydreaming. A camera-based system monitors employing a behavior-based method that uses Euclidean distance and a face landmarks algorithm to identify the driver's head position, eye blinked, eye closure by and facial recognition. Driver weariness is measured using these attributes. in real-time and promptly alert the driver through a voice speaker, while also forwarding an email to a designated emergency contact who can provide assistance or intervene as needed. The email is transmitted to the designated recipient using an

IoT module, relying on wireless communication. The proposed system is integrated with a Raspberry Pi3, a credit card-sized a computer system, and a Pi camera that can measure the strength of impact effects and detect eye movements during an accident, subsequently alerting nearby emergency services and owners of the accident location via GPS. This comprehensive approach aims to enhance driver safety and reduce the risk of fatigue-related accidents.

II. RELATED WORKS

In the field of Image processing in computer science is the use of sophisticated computer algorithms to modify, examine, and derive valuable information from digital photographs. Image processing, a specialized area of the processing of digital signals, has many benefits over conventional analogy methods. It makes it possible to handle complicated input data using a far wider variety of advanced algorithms and may successfully address problems like noise buildup and distortion in the signal that frequently occur within the complex processing pipeline. Moreover, image processing may be modeled and represented as a multivariate system, enabling more thorough and nuanced analysis, because digital pictures are naturally characterized over two-dimensional spatial domains. This enables the creation of potent methods for tasks like object identification, pattern recognition, and picture enhancement, which have broad implications in domains ranging from autonomous car navigation to medical imaging. [4][5]

A. DROWSINESS DETECTION THROUGH REGION OF INTEREST

The region analyzer can be a useful method for recognizing a driver's face. As seen by the blue rectangle, the ROI represents the specific area of the image that is of interest for analysis. To establish an ROI, the initial step involves obtaining the first frame's green rectangle region from the the Haar Cascade Classifier, which includes the width and height parameters necessary for defining the region. Subsequently, the rectangle is scaled up to accurately define the ROI that will be the focus of further processing and detection. Furthermore, there are multiple intricate steps required to precisely calculate the ROI area, and this meticulous process must be carried out for each and every region of interest that needs to be examined. Careful attention to detail in defining the ROI is crucial, as it serves as the foundation for the subsequent facial detection algorithms to operate effectively within the specified boundaries.

DISADVANTAGES OF REGION OF INTEREST

- The algorithm utilizes additional frames or bounding boxes to perform facial detection.
- It struggles to identify faces in low-light conditions.
- The use of region of interest may not be necessary when a Haar cascade classifier can achieve the same facial detection process.
- The algorithm exhibits difficulty identifying faces of individuals wearing glasses during driving.

B. DETECTION OF DROWSINESS THROUGH LBPH

This algorithm employs Histograms of local binary patterns are used to identify faces. The first calculation in the local binary pattern's histogram approach involves converting the original image into a binary format by generating an intermediate image. This is accomplished by first converting the image into a matrix form, where the The threshold value is the matrix's center value. The neighboring values are then defined using this threshold and can be either 0 or 1. The values in the matrix that have been 1 are taken into account, whereas the other values are discarded. These binary values correspond to individual pixels within the image. Through this process of thresholding and binarization, the facial region within the image can be identified and isolated for further processing and analysis.

DISADVANTAGES OF LBHP

- This approach results in a lower incidence of urate-related outcomes, which can have significant health implications for patients.
- The computational complexity of this approach is substantial, requiring significant processing power and memory resources to execute the underlying algorithms. This high complexity can pose challenges in terms of scalability and real-time performance, particularly while working with time-sensitive applications or big data sets.

This approach is only viable when the data sample size is relatively small and the computational resources available are limited. The high computational complexity of the underlying algorithms means that this approach may not be scalable to larger data sets or real-time applications that require fast processing. Significant processing power and memory resources are needed to execute these algorithms, which can pose significant challenges in terms of practical implementation, particularly in resource-constrained environments.

C. EYE BLINK DETECTION METHOD

Borolie and Ahmad created the ground-breaking Driver Drowsiness System, which utilizes cutting-edge non-intrusive machine-based approaches to detect and monitor driver drowsiness. The system comprises a strategically positioned web camera that captures the driver's facial expressions and head movements, both in real-time and from saved video recordings, enabling comprehensive simulation and analysis. The supplied footage is painstakingly broken down into individual pictures and processed. sequentially. Leveraging the Viola-Jones algorithm, the system accurately detects the driver's face within these frames. A sophisticated cascade classifier is then employed to extract the requisite facial features, such as the eyes, mouth, and head, providing a detailed assessment of the condition of the driver. The main sign that someone feels sleepy is the rate of eye blinking, which typically ranges from 12 to 19 per minute. The system recognizes drowsiness when the blinking frequency falls below this standard range. Rather than directly calculating the number of eye blinks, the system determines the average drowsiness level by assigning a value of zero to the detected eye, while partially or fully open eyes are represented by non-zero values. An advanced equation is utilized to compute this comprehensive average drowsiness level, providing a reliable and nuanced assessment of the driver's alertness.[6][7]

$$\%d = \frac{No.\ of\ Closed\ eyesfound}{no.\ of\ frames} \quad (1)$$

The mechanism sounds an alarm to warn the driver if the value exceeds the predetermined threshold. Furthermore, it is also thought that yawning triggers the alarm. Videos, both online and offline, are employed in experiments conducted on two distinct platforms. The outcomes demonstrate that the system can reach 90% efficiency.

D. EYE CLOSENESS DETECTION METHOD

In Khunpisuth's research, a camera mounted on a Raspberry Pi & a Raspberry Pi three Model B are used in an experiment to determine a driver's degree of tiredness. Using a classifier with a Haar cascade from the Viola-Jones method, the experiment first records video with the Pi camera before identifying facial areas in the pictures. The researchers train the system on several images captured under different lighting conditions, ultimately achieving a success rate of 83.09% in a case study with ten participants. The geographical region of Interest, and this is represented by the detected face, is shown as a blue rectangle. The last video frame is then subjected to the Haar cascade classifier, which shrinks the ROI. The vehicle's eye blink rate is used to determine the degree of tiredness following face identification. The authors use three templates to assess eye blink and area, and matched template on the face is used to determine the eye region. Since it outperforms other template matching methods, OpenCV's CV_TM_CCOEFF_NORMED approach is taken into consideration. Combining facial recognition with eye detection allows for continuous monitoring of the driver's eye blinking and closure rate. The system can determine Regardless of whether one's eyes are open or closed, the closed eye grade being higher than the open eye, and vice versa. The researchers assume that the face must be in a frontal posture for the cascade of the Haar classifier to work properly. They provide a way to fix this by turning any slanted features back toward the frontal position. First, they determine whether the head is tilted and then calculate the degree of rotation required. After accurately detecting the face and eyes, the system can determine the driver's drowsiness level. If the driver blinks their eyes too often, the mechanism signals fatigue. To warn the motorist, a loud sound happens when the sleepiness level hits 100[8]

E. INFERENCE OF THE RELATED WORKS

Existing drowsiness detection systems frequently rely on devices that monitor physiological indicators include blood pressure, heart rate, and respiration rate. These devices can make drivers feel uncomfortable, and there is no guarantee that they will wear them consistently while driving. Furthermore, these devices may malfunction or become lost, leading to inaccurate results. The current systems also struggle to perform well in low-light conditions, as they are unable to reliably detect the driver's face and eyes, resulting in reduced accuracy.

This limitation is particularly concerning, as drowsiness is a significant contributing factor to many traffic accidents, and effective detection systems are crucial for improving road safety. Developing more robust and unobtrusive drowsiness detection methods is an important challenge that researchers in this field continue to address. Potential solutions may involve integrating computer vision techniques to analyze the driver's facial features and behavior, or exploring alternative sensor modalities that can operate effectively in diverse lighting conditions. Ultimately, the design of reliable and user-friendly drowsiness detection systems remains a critical priority for enhancing driver safety and reducing the risk of fatigue-related accidents. In [16], the authors proposed a **QoS-enabled data dissemination framework for hierarchical VANETs that leverages machine learning techniques** to improve reliable and timely information delivery. It introduces a hierarchical network structure to reduce broadcast overhead and enhance scalability in highly dynamic vehicular environments

III. PROPOSED SYSTEM

In this research, the OpenCV, which is library's Haar cascading classifier was used to identify the driver's face traits in real-time movies. OpenCV is a popular open-source visual computing library that supports languages like C++, Java, and Python and offers a full suite of tools for computational image processing & machine learning applications. To categorize the input and identify the nose and eye regions, OpenCV's built-in Haar Cascade Classifiers were used. This machine learning-based approach involves a pre-trained cascade of positive and negative images, which enables the robust detection of objects in other required images. As seen in Fig. 4.1, identifying the facial and ocular areas is a crucial step in assessing driver drowsiness.

Using the YawD dataset and the CNN model, the data was first preprocessed. The video frames were converted into individual images, resized to a standard resolution of 24 × 24 pixels, and the The OpenCV library was used to detect the face. The resulting images were then converted to grayscale, and the mouth state was labeled accordingly, with "1" representing an open mouth and "0" representing a closed mouth. This labeled data was saved in a CSV file for further analysis and model training.

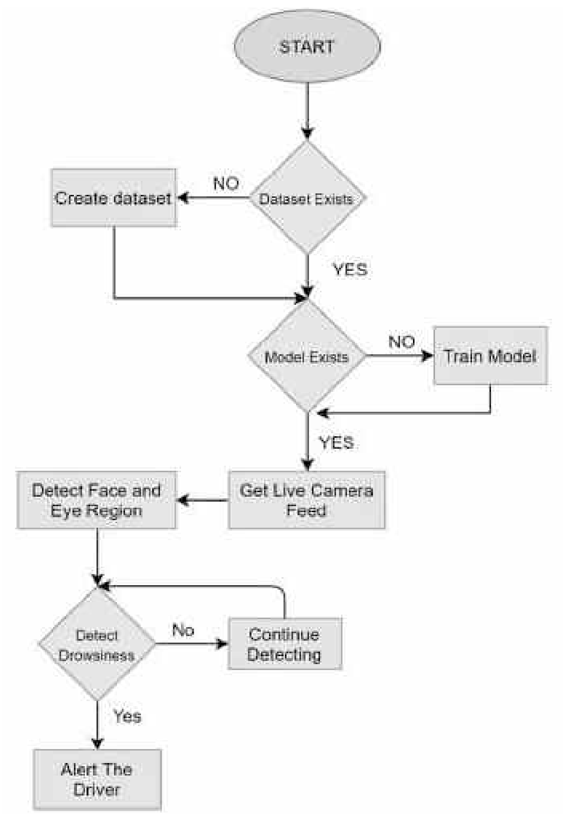

Figure.1 Overall Proposed Architecture

Similarly, The CEW database was made accessible in grayscale and as cropped eye pictures with a 24 × 24, pixels resolution. "1" and "0" were used to denote the open and closed eye states, respectively, in this dataset. The labeled information was also recorded in a CSV file. During the machine learning training procedure, the pooled dataset was then split into 80% to be utilized for training and 20 percent for validation. Lastly, every pixel value was split by 32 and recorded as a floating32 value, prepared for incorporation into the deep learning algorithm to guarantee correct scaling the the input data.

A. IMAGE SEGMENTATION

- Face Recognition
- Face tracking
- eye tracking
- eye detection
- drowsiness detection
- distraction detection

Face Detection: The module in question looks for a human face in the video play after receiving video input through the camera. The facial recognition process leverages Haar-like feature-based cascade classifiers, primarily the Frontal Face Cascade Classifier, which is designed to identify frontal views of faces. Once a face is detected, it is captured within a rectangular bounding box, stored in memory after being transformed into a grayscale visual representation. for potential future use in model training or other facial recognition tasks.

Eye Detection: The proposed model is designed to create a method for detecting sleepiness, which requires that the eyes be the main signal. A Haar cascade classifier—more precisely, the Haar Cascades Eye Classifier—is used to identify the eyes from visual data. This method makes it possible to identify the eyeballs in a rectangular shape.

Face Tracking: Considering the project's real-time needs, we must continuously monitor the participants' facial expressions to detect any potential distractions or signs of disengagement. Maintaining vigilant, real-time surveillance of the participants' faces is critical to ensuring the study yields meaningful and reliable results, as any lapses in attention could skew the data collected. Accordingly, the faces are constantly detected and analyzed throughout the duration of the study to provide a comprehensive understanding of the participants' cognitive and emotional states as they engage with the tasks at hand.

Eye Tracking: This component receives its input data from the results generated by the preceding processing module. The ocular state of the user, which reflects their level of alertness and attentiveness, is determined through the application of the Perclos algorithm. This algorithm analyzes the user's eye blink patterns and duration of eyelid closure to provide a quantitative measure of their vigilance and drowsiness levels.

Drowsiness detection: The frequency is determined by the previous module, and if it stays at zero for an extended duration, the system generates an alert to notify the driver of drowsiness.

To identify the user's face & eye regions frame-by-frame, the system starts the camera that the user is using and uses OpenCV's HAAR cascade. The trained convolution neural system model then receives these frames. The CNN model provides the user's ocular condition, including whether the eyes are wide or narrow, as well as if the user is grunting or displaying other actions. The device sends out a sleepiness signal if the user's eyelids stay closed for a predetermined amount of time. Similarly, the system provides a sleepiness signal if an individual is yawning frequently for a certain amount of time. One dynamic parameter that may be changed as necessary is the time threshold. The proposed deep learning model is a convolutional neural network. After extensive experimentation, the researchers optimized their suggested model with two dense layers, four layers that maximum-pool, and four convolutional (conv2d) layers. Four layers for dropouts were used in order to reduce overfitting incorporated.

Frame per frame, the system detects the user's facial and eye regions using OpenCV's HAAR cascade after initializing the user's camera. The trained convolution neural network model then receives these frames. The CNN model provides outputs indicating whether the eyes are open or closed, as well as if the driver is gasping or acting in any other way. The technology generates a sleepiness alarm if the eyelids stay closed for the designated amount of time. Similarly, the system will issue a sleepiness alarm if the individual in question is yawning frequently for the specified time threshold. One dynamic parameter that may be changed is the time threshold adjusted as needed. The proposed deep learning model is a convolutional neural network. After extensive experimentation, the researchers perfected their model with two dense layers, four max-pooling layers that were flattened, and four convolutional (conv2d) layers. Four-layer dropouts were used in order to reduce overfitting incorporated.

B. TRAIN THE MODEL

Processing data like temperature readings, market prices, or color intensities—in this instance, face and eye detection—is how machine learning works. Usually, the data is preprocessed to create features. For instance, we may use an edge detector to extract parameters such as edge direction, edge the force and offset form the facial center for every face in a library of 1,000 face photos. A vector with faces of no less than 500 data each face may be created by combining these characteristics. The collected data may then be used to build models using machine learning. A clustering technique could be the better option if determining how faces clump together is the aim. Alternatively, a classifier algorithm may be used to learn how to determine a woman's age based on the edge structures on her face that are observed. Machine learning algorithms may evaluate the characteristics that have been retrieved and modify the weights, boundaries, and other criteria to optimize the performance in order to accomplish these goals. The core of machine learning is this process of adjusting parameters to achieve a certain objective.

Understanding the efficiency and limitations of machine learning techniques is crucial. This may be a nuanced and multifaceted undertaking. Typically, we partition the input dataset into a comparatively small test set and a large training set. With the data feature vectors, we can next use our classifier technique on the training set to construct an age prediction model. The performance and generalisation potential of our age forecasting classifier may next be assessed using the remaining test set photos. A thorough analysis of the held-out testing equipment is key to assessing the true efficacy of the learned model, as this provides an unbiased estimate of its real-world predictive power.

C. TEST THE MODEL

The testing set is not used in the training phase, and a classifier is is not given access to the test set's age labels. The classifier is then run on each of the 100 pictures in the test set, and its precision of the feature vector's age predictions compared to the actual ages is noted. We may look at adding more features to the data or taking into account different classifier types if the classifier performs poorly. There are many different types of classifiers and training algorithms. On the other hand, when the classifier works well, we have a potentially profitable model that can be used with actual data. This technology might be used to modify a video game's performance according on the user's age. Once established, the classifier sees faces it has never seen before and uses the knowledge it gained from the training process to make choices. Lastly, we usually use the validation data set when creating a categorization system. Testing the entire system completely in the end might be a big step. Usually, we want to adjust the parameters before deciding on our classifier and the final evaluation. To do this, we may divide the original 1,000-face assortment across three parts: 800 faces for training, 100 faces for validation, and 100 faces for testing. We can track our progress by doing initial assessments on the validation data as we go through the training data. The classifier may then be run on the test set to determine the final outcome if we are happy with the output from the validation set.

IV. RESULT AND DISCUSSION

One of the most important performance metrics is accuracy, which shows how many of a model's predictions were accurate overall. It can provide immediate feedback on whether a model has been trained effectively and how it may perform generally. However, accuracy alone does not offer detailed information on the model's use in the specific problem at hand. Precision, also known as Positive Predictive

Value, is a suitable metric for situations where False positives come at a great cost. It focuses on the percentage of genuine positives among all of the model's positive predictions. When there are serious repercussions for misidentifying a good occurrence, this statistic becomes even more crucial. Conversely, when the cost of incorrect findings is high, the model measure that is used to choose the optimal model is called recall, or sensitivity. Out of all the observed positive events, recall quantifies the percentage of genuine positives that the the model properly detected. When the penalty of missing a positive incidence is substantial, this statistic becomes extremely important. Combining precision and recall, the F1-score provides a thorough assessment of the model's correctness. By balancing these two crucial indicators, it offers a fair evaluation of the model's performance. A high F1-score is a useful indicator for assessing model performance since it denotes a model with minimal false positives and false negatives. The selection and optimisation of models for particular applications and issue settings are guided by these model assessment metrics, which provide a better knowledge of a model's strengths and limits.

True Positives (TP):- These are the positively predicted values that were accurately predicted, indicating that both the actual and anticipated class values are yes.

True Negatives (TN):- These are the accurately predicted negative values, indicating that both the actual class value and the projected class value are zero.

When a actual class differs from the anticipated class, we get misleading results and false negatives..

False Positives (FP):- When the anticipated class is yes and the actual school is no.

False Negatives (FN):- When the anticipated class is no and the actual course is yes.

$$Accuracy = \frac{(TP+TN)}{(TN+FN)+(FP+TP)} \quad (1)$$

$$Recall = \frac{TP}{(FN+TP)} \quad (2)$$

$$Precision = \frac{TP}{(FP+TP)} \quad (3)$$

$$F1-measure = 2 \times \frac{(Precision \times Recall)}{(Precision+Recall)} \quad (4)$$

Whereas

- False Positives (FP) = wrongly detected
- True Negative (TN) = correctly rejected
- False Negative (FN) = erroneously rejected
- True Positive (TP) = appropriately identified

Precision: Finding the number of predicted positive classes that truly belong to the beneficial class is known as precision.

$$Recall = \frac{TP}{(FN+TP)} \quad (5)$$

Recall: The number of favorable class projections made proportion of all positive specimens in the collection of data is known as recall.

$$Recall = \frac{TP}{(FN+TP)} \quad (6)$$

Figure 2 Drowsiness Detected.

Figure 3 Drowsiness is not detected.

V. CONCLUSION

This study employs Using deep learning to identify driver fatigue in real-time. Convolutional neural network models are utilized to identify the onset of drowsiness based on visual cues. The researchers leveraged the driver's cheek and eye regions were precisely detected using OpenCV's Haar cascade technique, which acted as the input to the proposed CNN model. Through extensive training, the CNN was able to learn the distinctive patterns and features associated with drowsiness, achieving an excellent real-time performance with an average accuracy of 96%. Looking ahead, the researchers plan to further improve the system's performance by acquiring a larger and more diverse dataset for training. This would allow the CNN to better generalize to a wider range of driving conditions and driver characteristics. Additionally, the team is considering integrating a face recognition feature to enhance security and prevent vehicle theft. Finally, the researchers aim to convert the system into a mobile application, making the drowsiness detection technology more accessible to a broader audience of drivers. [2][15]

REFERENCES

[1] J. Gwak, A. Hirao, and M. Shino, “An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing,” Applied Sciences, vol. 10, no. 8, p. 2890, 2020, doi: 10.3390/app10082890.

[2] Z. Kepesiova, J. Ciganek, and S. Kozak, “Driver drowsiness detection using convolutional neural networks,” in Proc. 2020 Cybernetics & Informatics (K&I), 2020, pp. 1–5, doi: 10.1109/KI48306.2020.9039851.

[3] F. You, X. Li, Y. Gong, H. Wang, and H. Li, “A real-time driving drowsiness detection algorithm with individual differences consideration,” IEEE Access, vol. 7, pp. 179396–179408, 2019, doi: 10.1109/ACCESS.2019.2958667.

[4] S. Mehta, S. Dadhich, S. Gumber, and A. J. Bhatt, “Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio,” SSRN Electronic Journal, 2019, doi: 10.2139/ssrn.3356401.

[5] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som, and H. A. Ameen, “Drowsiness detection system using eye aspect ratio technique,” in Proc. IEEE Student Conf. Research and Development (SCOReD), 2020, pp. 1–6, doi: 10.1109/SCOReD 50371.2020.9251035.

[6] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, “YawDD: Yawning detection dataset,” IEEE Dataport, 2020, doi: 10.21227/e1qm-hb90.

[7] F. Song, X. Tan, X. Liu, and S. Chen, “Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients,” Pattern Recognition, vol. 47, no. 9, pp. 2825–2835, 2014.

[8] B. K. Savas and Y. Becerikli, “Real-time driver fatigue detection system based on multi-task CNN,” IEEE Access, vol. 8, pp. 12491–12498, 2020, doi: 10.1109/ACCESS.2020.2963960.

[9] S. Bavkar, B. Iyer, and S. Deosarkar, “Rapid screening of alcoholism: An EEG-based optimal channel selection approach,” IEEE Access, vol. 7, pp. 99670–99682, 2019, doi: 10.1109/ACCESS.2019.2927267.

[10] S. Bavkar, B. Iyer, and S. Deosarkar, “BPSO-based method for screening of alcoholism,” in Proc. ICCCE 2019, Lecture Notes in Electrical Engineering, vol. 570, Singapore: Springer, 2020, pp. 47–53, doi: 10.1007/978-981-13-8715-9_6.

[11] S. Bavkar, B. Iyer, and S. Deosarkar, “Optimal EEG channels selection for alcoholism screening using EMD domain statistical features and harmony search algorithm,” Biocybernetics and Biomedical Engineering, vol. 41, no. 1, pp. 83–96, 2021.

[12] P. Deshpande and B. Iyer, “Research directions in the Internet of Everything (IoET),” in Proc. 2017 Int. Conf. Computing, Communication and Automation (ICCCA), Greater Noida, India, pp. 1353–1357, 2017, doi: 10.1109/CCAA.2017.8230008.

[13] D. Deshmukh and B. Iyer, “Design of IPSec virtual private network for remote access,” in Proc. 2017 Int. Conf. Computing, Communication and Automation (ICCCA), Greater Noida, India, pp. 716–719, 2017, doi: 10.1109/CCAA.2017.8229894.

[14] B. Iyer and N. Patil, “IoT enabled tracking and monitoring sensor for military applications,” Int. J. System Assurance Engineering and Management, vol. 9, pp. 1294–1301, 2018, doi: 10.1007/s13198-018-0727-8.

[15] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, “Driver drowsiness detection model using convolutional neural networks techniques for Android application,” in Proc. 2020 IEEE Int. Conf. Informatics, IoT, and Enabling Technologies (ICIoT), 2020, pp. 1–6.

[16] K. G. Krishnakumar and E. J. Thomson . Fredrik, “QoS enabled data dissemination in hierarchical VANET using machine learning approach,” International Journal of Signal and Imaging Systems Engineering, vol. 10, no. 5, pp. 231–236, 2017.